



## Indian Financial Market Data **for R/Rmetrics**

Diethelm Würtz  
Mahendra Mehta  
Andrew Ellis  
Yohan Chalabi



## R/Rmetrics eBook Series

R/Rmetrics eBooks is a series of electronic books and user guides aimed at students and practitioner who use R/Rmetrics to analyze financial markets.

*A Discussion of Time Series Objects for R in Finance* (2009)  
Diethelm Würtz, Yohan Chalabi, Andrew Ellis

*Portfolio Optimization with R/Rmetrics* (2010),  
Diethelm Würtz, William Chen, Yohan Chalabi, Andrew Ellis

*Basic R for Finance* (2010),  
Diethelm Würtz, Yohan Chalabi, Longhow Lam, Andrew Ellis  
Early Bird Edition

*Financial Market Data for R/Rmetrics* (2010)  
Diethelm Würtz, Andrew Ellis, Yohan Chalabi  
Early Bird Edition

*Indian Financial Market Data for R/Rmetrics* (2010)  
Diethelm Würtz, Mahendra Mehta, Andrew Ellis, Yohan Chalabi

# INDIAN FINANCIAL MARKET DATA FOR R/RMETRICS

DIETHELM WÜRTZ  
MAHENDRA MEHTA  
ANDREW ELLIS  
YOHAN CHALABI

*Series Editors:*

Prof. Dr. Diethelm Würtz  
Institute of Theoretical Physics and  
Curriculum for Computational Science  
Swiss Federal Institute of Technology  
Hönggerberg, HIT G 32.3  
8093 Zurich

Dr. Martin Hanf  
Finance Online GmbH  
Zeltweg 7  
8032 Zurich

*Contact Address:*

Rmetrics Association  
Zeltweg 7  
8032 Zurich  
info@rmetrics.org

*Publisher:*

Finance Online GmbH  
Swiss Information Technologies  
Zeltweg 7  
8032 Zurich

*Authors:*

Yohan Chalabi, ETH Zurich  
Mahendra Mehta, NeuralTechSoft Mumbai  
Andrew Ellis, Finance Online GmbH Zurich  
Diethelm Würtz, ETH Zurich

ISBN: 978-3-906041-05-6

© 2009-2010, Finance Online GmbH, Zurich

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Rmetrics Association, Zurich.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and authors have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

**Trademark notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

## DEDICATION

*This book is dedicated to all students  
and academic researchers who work in empirical finance  
and cannot afford to buy data from commercial providers.*



## PREFACE

Are you working with R and Rmetrics in the field of teaching finance? Then you will often use financial market data from the Internet, most likely downloaded from an Internet portal like Yahoo Finance or the Federal Reserve.

You may have asked yourself whether there are also other data sources on the Internet that can be used for courses and lectures and how can the download functions be customized to your personal needs. This Rmetrics eBook tries to answer questions you might have about these issues. We will show you how to access daily Indian stock market data and indices from the *National Stock Exchange of India*, NSE, and related financial market data, such as ETFs from the NSE or other sources. We will also show you how to write convenient download functions to make your life more comfortable.

In this eBook you will find solutions for how to compose download URLs, how to download financial time series and how to generate listings for the financial instruments to make the search for individual time series easier. In all cases examples and exercises are given.

This eBook is a Sweave document and we will keep it up-to-date with changes on the servers of the data Internet portals and with the most recent R and Rmetrics packages. This eBook is copyrighted by the *Rmetrics Association* and *Finance Online* in Zurich. It can be ordered from the Rmetrics website [www.rmetrics.org](http://www.rmetrics.org).

Enjoy it!

Diethelm Würtz  
Mahendra Mehta  
Zurich and Mumbai, May 2010





# CONTENTS

DEDICATION	V
PREFACE	VII
CONTENTS	IX
LIST OF FIGURES	XIII
LIST OF TABLES	XV
<b>I NSE Download Functions</b>	<b>1</b>
1 NSE STOCK MARKET DATA	2
1.1 <i>The National Stock Exchange of India</i> . . . . .	2
1.2 <i>NSE Index Time Line</i> . . . . .	3
1.3 <i>NSE Major Indices</i> . . . . .	3
1.4 <i>NSE Other Indices</i> . . . . .	6
1.5 <i>References and Index Fact Sheets</i> . . . . .	9
1.6 <i>NSE Legal Issues</i> . . . . .	9
2 DOWNLOADING NSE INDICES	10
2.1 <i>The Download URL</i> . . . . .	10
2.2 <i>Downloading an Index Time Series</i> . . . . .	11
2.3 <i>The Function nseIndexDownload()</i> . . . . .	13
3 DOWNLOADING NSE EQUITIES	14
3.1 <i>The Download URL</i> . . . . .	14
3.2 <i>Downloading Equity Time Series</i> . . . . .	15
3.3 <i>The Function nseEquityDownload()</i> . . . . .	16
4 DOWNLOADING NSE DEBT SERIES	19
4.1 <i>The Download URL</i> . . . . .	19
4.2 <i>Download Debt Time Series</i> . . . . .	20

4.3	<i>The Function nseDebtDownload()</i> . . . . .	21
4.4	<i>NSE Closing Dates</i> . . . . .	21
<b>II NSE Listing Functions</b>		<b>24</b>
5	INDEX CATEGORIZATION	25
5.1	<i>Categorization Summary</i> . . . . .	25
5.2	<i>Major Index Categories</i> . . . . .	25
5.3	<i>Other Index Categories</i> . . . . .	26
6	NSE MARKET INDEX LISTINGS	27
6.1	<i>The NIFTY Index Listing</i> . . . . .	27
6.2	<i>The Function nseListing()</i> . . . . .	30
6.3	<i>The Major Index Listing Files</i> . . . . .	31
6.4	<i>The Other Index Listing Files</i> . . . . .	32
<b>III Related Data from Other Sources</b>		<b>34</b>
7	DATA FROM BOMBAY STOCK EXCHANGE	35
7.1	<i>BSE Internet Portal</i> . . . . .	35
7.2	<i>Downloading Equity Prices</i> . . . . .	35
7.3	<i>The Function bseDownload()</i> . . . . .	38
7.4	<i>Downloading BSE Indices</i> . . . . .	40
7.5	<i>BSE Equity Listings</i> . . . . .	41
7.6	<i>BSE Scrip Code and Scrip Code Search</i> . . . . .	42
7.7	<i>BSE Holiday Calendar</i> . . . . .	45
8	DATA FROM FINANCE YAHOO	46
8.1	<i>Finance Yahoo Internet Portal</i> . . . . .	46
8.2	<i>Downloading CNX Indices</i> . . . . .	46
8.3	<i>The Function yahooSeries()</i> . . . . .	48
8.4	<i>Downloading BSE Indices</i> . . . . .	50
9	DATA FROM ONVISTA	54
9.1	<i>OnVista Internet Portal</i> . . . . .	54
9.2	<i>MSCI India Index</i> . . . . .	55
9.3	<i>The Function msciOnvistaDownload()</i> . . . . .	56
10	DATA FROM OANDA	60
10.1	<i>Oanda Internet Portal</i> . . . . .	60
10.2	<i>The Download URL</i> . . . . .	60
10.3	<i>The Function oandaDownload()</i> . . . . .	62

<b>IV Appendix</b>	<b>65</b>
A PACKAGES REQUIRED FOR THIS EBOOK	66
B NSE FUNCTION SUMMARY	70
C LYNX TEXT READER	71
D RMETRICS TERMS OF LEGAL USE	73
E NSE TERMS OF LEGAL USE	75
BIBLIOGRAPHY	77
INDEX	78
ABOUT THE AUTHORS	79



## LIST OF FIGURES

2.1	CNX IT - Log Index and log Returns . . . . .	12
3.1	NTPC Equity Series . . . . .	18
4.1	Overnight Interest Rates . . . . .	22
6.1	TATASTEEL - Log Returns and Turnover . . . . .	30
7.1	Volatility and High-Low Spreads for HINDALCO . . . . .	38
7.2	Volatility vs High-Low Spreads for HINDALCO . . . . .	39
8.1	Adjusted Closing Price and Volume of CNX Nifty . . . . .	48
8.2	Adjusted Close and Volume for the Sensex Index . . . . .	51
8.3	NTPC Data from Finance Yahoo . . . . .	53
9.1	Log-Index and Log>Returns of MSCI India Index . . . . .	57
9.2	Log MSCI Index . . . . .	58
9.3	Correlation Plot of MSCI Returns . . . . .	59
10.1	Log of INR FX Rates . . . . .	64



# LIST OF TABLES

1.1	NSE Time Line of Milestones	3
1.2	NSE List of Major Indices	4
1.3	NSE List of Major Indices	6
1.4	Constituents List of Industry Indices	7
2.1	NSE Symbols for Index Time Series	10
3.1	Fixed Ranges for Downloads from NSE	14
4.1	NSE Debt Segments	19
5.1	NSE Major Index Categorization Scheme	25
5.2	NSE Other Index Categorization Scheme	26
6.1	NSE Major Index Listing Files	31
6.2	NSE Other Index Listing Files	32
7.1	BSE Major Indices	41
7.2	BSE Sector Indices	41
7.3	BSE Dollex Indices	41
7.4	BSE Equity Listing	41
7.5	BSE SENSEX Index Listing	42
7.6	BSE Holiday Calendar 2010	45
8.1	MSCI Indices from OnVista's Internet Portal	47
8.2	MSCI Indices from OnVista's Internet Portal	50
9.1	Category Links on the OnVista Server	54
B.1	Functions to download data from NSE	70
B.2	Arguments for Download and Listing Functions	70
E.1	DotEx Data Products	75





PART I

# NSE DOWNLOAD FUNCTIONS

# CHAPTER 1

## NSE STOCK MARKET DATA

```
> library(fBasics)
> library(fImport)
```

### 1.1 THE NATIONAL STOCK EXCHANGE OF INDIA

NSE is the *National Stock Exchange* of India. Its Internet portal can be accessed on the following web page

<http://www.nseindia.com>

The equities or capital market segment of the NSE started operations in November 1994 and operations in derivatives in June 2000. The following years showed a rapid development of the Indian capital market with the introduction of Internet trading, of Exchange traded funds (ETF), and of stock derivatives. Another milestone was the creation of Indias first volatility index, IndiaVIX in April 2008. Currency derivatives started in August 2008 for currency futures in the US Dollar and and in Indian Rupees, INR. Interest rate futures were added to the financial products in September 2009.

A more detailed overview on the history of NSE can be found on the Web page

[http://www.nseindia.com/content/us/us\\_organisation.htm](http://www.nseindia.com/content/us/us_organisation.htm)

from where this information has been taken. We also refer for further information to the Wikipedia page

[http://en.wikipedia.org/wiki/National\\_Stock\\_Exchange\\_of\\_India](http://en.wikipedia.org/wiki/National_Stock_Exchange_of_India)

## 1.2 NSE INDEX TIME LINE

A time line of the NSE with the major historical milestone can be found on

[http://www.nseindia.com/content/us/us\\_milestones.htm](http://www.nseindia.com/content/us/us_milestones.htm)

A brief summary is given in the following table.

LISTING 1.1: NSE TIME LINE OF MILESTONES

---

November 1992	Incorporation
June 1994	Wholesale debt market segment goes live
November 1994	Capital market segment goes live
October 1995	Became largest stock exchange in the country
April 1996	Launch of S&P CNX Nifty
December 1996	Launch of CNX Nifty Junior
May 1998	Launch of NSE's Web-site
February 2000	Commencement of Internet Trading
June 2000	Derivatives Trading (Index Futures)
September 2000	Launch of 'Zero Coupon Yield Curve'
June 2001	Trading in Index Options
July 2001	Trading in Options on Individual Securities
November 2001	Trading in Futures on Individual Securities
January 2002	Exchange Traded Funds (ETFs)
October 2002	NSE Government Securities Index
January 2003	Trading in Retail Debt Market
June 2003	Interest Rate Futures
August 2003	Futures and Options in CNXIT Index
June 2005	Futures and Options in BANK Nifty Index
June 2007	Derivatives on Nifty Junior and CNX 100
October 2007	Derivatives on Nifty Midcap 50
January 2008	Mini Nifty derivative contracts
March 2008	Long term option contracts on S&P CNX Nifty Index
April 2008	India Volatility Index VIX
August 2008	Currency Derivatives
August 2009	Interest Rate Futures

---

## 1.3 NSE MAJOR INDICES

The Internet portal of NSE offers a large number of daily historical time series for indices and equities. The indices are categorized by the NSE in two groups named *major indices* and *other indices*.

The major indices can be found on the following web page

[http://www.nseindia.com/content/indices/ind\\_majorindices.htm](http://www.nseindia.com/content/indices/ind_majorindices.htm)

Here is a list of the category of the major equity indices.

LISTING 1.2: NSE LIST OF MAJOR INDICES

---

S&P CNX Nifty Index  
CNX Nifty Junior Index  
CNX 100 Index  
S&P CNX 500 Index  
CNX Midcap Index  
Nifty Midcap 50 Index  
S&P CNX Defty Index

---

### *S&P CNX Nifty Index*

The *S&P CNX Nifty* is a well diversified 50 stock index accounting for 22 sectors of the Indian economy. The index is used for a variety of purposes such as benchmarking portfolios, index based derivatives and index funds. The cited and more information can be found on the following web page

[http://www.nseindia.com/content/indices/ind\\_nifty.htm](http://www.nseindia.com/content/indices/ind_nifty.htm)

For additional information about the index methodology, and the index components follow the links

[http://www.nseindia.com/content/indices/nifty\\_freefloat\\_method.pdf](http://www.nseindia.com/content/indices/nifty_freefloat_method.pdf)  
[http://www.nseindia.com/content/indices/ind\\_niftylist.csv](http://www.nseindia.com/content/indices/ind_niftylist.csv)

### *CNX Nifty Junior Index*

The next rung of liquid securities after *S&P CNX Nifty* is the *CNX Nifty Junior* index. This index may be useful to think of the *S&P CNX Nifty* and the *CNX Nifty Junior* as making up the 100 most liquid stocks in India. The cited and more information can be found on following web page

[http://www.nseindia.com/content/indices/ind\\_jrnifty.htm](http://www.nseindia.com/content/indices/ind_jrnifty.htm)

For additional information about the index components follow the link

[http://www.nseindia.com/content/indices/ind\\_jrniftylist.csv](http://www.nseindia.com/content/indices/ind_jrniftylist.csv)

### *CNX 100 Index*

The *CNX 100* index is a diversified 100 stock index accounting for 35 sectors of the Indian economy. See

[http://www.nseindia.com/content/indices/ind\\_cnx100.htm](http://www.nseindia.com/content/indices/ind_cnx100.htm)

For additional information about the index components follow the link

[http://www.nseindia.com/content/indices/ind\\_cnx100list.csv](http://www.nseindia.com/content/indices/ind_cnx100list.csv)

### *S&P CNX 500 Index*

The *S&P CNX 500* is India's first broad based benchmark of the Indian capital market. The *S&P CNX 500* represents about 92.57% of total market capitalization and about 91.17% of the total turnover on the NSE (September 2009). More information can be found on the following web page

[http://www.nseindia.com/content/indices/ind\\_cnx500.htm](http://www.nseindia.com/content/indices/ind_cnx500.htm)

For additional information about the index components follow the link

[http://www.nseindia.com/content/indices/ind\\_cnx500list.csv](http://www.nseindia.com/content/indices/ind_cnx500list.csv)

### *CNX Midcap Index*

The primary objective of the *CNX Midcap* index is to capture the movement and be a benchmark of the midcap segment of the Indian market. More information about the method of computation, base date and value, criteria for selection of constituent stocks can be found on the following web page

[http://www.nseindia.com/content/indices/ind\\_cnxmidcap.htm](http://www.nseindia.com/content/indices/ind_cnxmidcap.htm)

For additional information about the index components follow the link

[http://www.nseindia.com/content/indices/ind\\_cnxmidcaplist.csv](http://www.nseindia.com/content/indices/ind_cnxmidcaplist.csv)

### *Nifty Midcap 50 Index*

The primary objective of the *Nifty Midcap 50* Index is to capture the movement of the midcap segment of the Indian market. It can also be used for index-based derivatives trading. More information about the method of computation, base date and value, criteria for selection of constituent stocks, and statistics can be found on the following web page

[http://www.nseindia.com/content/indices/ind\\_niftymidcap50.htm](http://www.nseindia.com/content/indices/ind_niftymidcap50.htm)

For additional information about the index components follow the link

[http://www.nseindia.com/content/indices/ind\\_niftymidcap50list.csv](http://www.nseindia.com/content/indices/ind_niftymidcap50list.csv)

### *S&P CNX Defty Index*

The *S&P CNX Defty* is the US Dollar denominated *S&P CNX Nifty Fore* more information about the method of computation, calculation of closing values, and specifications we refer to the following link

[http://www.nseindia.com/content/indices/ind\\_defty.htm](http://www.nseindia.com/content/indices/ind_defty.htm)

### *NSE Other Indices*

The other indices can be found on the following web page

[http://www.nseindia.com/content/indices/ind\\_otherindices.htm](http://www.nseindia.com/content/indices/ind_otherindices.htm)

## 1.4 NSE OTHER INDICES

Here is a list of the category of the *Other Indices*.

LISTING 1.3: NSE LIST OF MAJOR INDICES

---

CNX IT Sector Index  
CNX FMCG Index  
CNX Millenium Index  
CNX PSE Index  
CNX MNC Index  
CNX IBG Index  
S&P CNX Industry Indices  
Customised Indices

---

### *CNX IT Sector Index*

Information technology, IT, is an important factor in the economy of India. Many large and profitable Indian companies today belong to the IT sector. As a benchmark of the Indian IT sector, the CNX IT sector index was developed. See

[http://www.nseindia.com/content/indices/ind\\_cnxit.htm](http://www.nseindia.com/content/indices/ind_cnxit.htm)

### *CNX FMCG Index*

FMCGs, Fast Moving Consumer Goods, are those goods and products, which are non-durable, mass consumption products, available off the shelf. The CNX FMCG Index is a 15 stock Index from the FMCG sector that trade on NSE. See

[http://www.nseindia.com/content/indices/ind\\_cnxfmcg.htm](http://www.nseindia.com/content/indices/ind_cnxfmcg.htm)

### *CNX Millenium Index*

[http://www.nseindia.com/content/indices/ind\\_cnxmill.htm](http://www.nseindia.com/content/indices/ind_cnxmill.htm)

*CNX PSE Index*

The CNX PSE index is a benchmark for the Public Sector Enterprises, and was established in 1991. The index captures the performance of the PSE segment of the Indian market. See

[http://www.nseindia.com/content/indices/ind\\_cnxpse.htm](http://www.nseindia.com/content/indices/ind_cnxpse.htm)

*CNX MNC Index*

The CNX MNC Index consists of 15 listed companies in which the foreign shareholding is over 50% and/or the management control is vested in the foreign company. See

[http://www.nseindia.com/content/indices/ind\\_cnxmnc.htm](http://www.nseindia.com/content/indices/ind_cnxmnc.htm)

*CNX IBG Index*

The CNX Bank Index is an index comprised of the 12 most liquid and large capitalized Indian Banking stocks. See

[http://www.nseindia.com/content/indices/ind\\_cnxbank.htm](http://www.nseindia.com/content/indices/ind_cnxbank.htm)

*S&P CNX Industry Indices*

[http://www.nseindia.com/content/indices/ind\\_indusindices.htm](http://www.nseindia.com/content/indices/ind_indusindices.htm)

LISTING 1.4: CONSTITUENTS LIST OF INDUSTRY INDICES

---

No:	Industry Name:
1	S&P CNX ABRASIVES
2	S&P CNX AIRCONDITIONERS
3	S&P CNX ALUMINIUM
4	S&P CNX AUTO ANCILLARIES
5	S&P CNX AUTOMOBILES - 2 AND 3 WHEELERS
6	S&P CNX AUTOMOBILES - 4 WHEELERS
7	S&P CNX BANKS
8	S&P CNX BEARINGS
9	S&P CNX BREW/DISTILLERIES
10	S&P CNX CABLES - TELECOM
11	S&P CNX CASTINGS/FORGINGS
12	S&P CNX CEMENT AND CEMENT PRODUCTS
13	S&P CNX CHEMICALS - INORGANIC
14	S&P CNX CHEMICALS - ORGANIC
15	S&P CNX CHEMICALS - SPECIALITY
16	S&P CNX CIGARETTES
17	S&P CNX COMPRESSORS / PUMPS
18	S&P CNX COMPUTERS - HARDWARE

---

19	S&P CNX COMPUTERS - SOFTWARE
20	S&P CNX CONSTRUCTION
21	S&P CNX CONSUMER DURABLES
22	S&P CNX CYCLES
23	S&P CNX DETERGENTS
24	S&P CNX DIESEL ENGINES
25	S&P CNX DIVERSIFIED
26	S&P CNX DYES AND PIGMENTS
27	S&P CNX ELECTRICAL EQUIPMENT
28	S&P CNX ELECTRODES
29	S&P CNX ELECTRONICS - INDUSTRIAL
30	S&P CNX ENGINEERING
31	S&P CNX FASTNERS
32	S&P CNX FERTILISERS
33	S&P CNX FINANCE
34	S&P CNX FINANCE - HOUSING
35	S&P CNX FINANCIAL INSTITUTION
36	S&P CNX FOOD AND FOOD PROCESSING
37	S&P CNX GEMS, JEWELLERY AND WATCHES"
38	S&P CNX HOTELS
39	S&P CNX GAS
40	S&P CNX LEATHER AND LEATHER PRODUCTS
41	S&P CNX MEDIA & ENTERTAINMENT
42	S&P CNX METALS
43	S&P CNX MINING
44	S&P CNX MISCELLANEOUS
45	S&P CNX OIL EXPLORATION/PRODUCTION
46	S&P CNX PACKAGING
47	S&P CNX PAINTS
48	S&P CNX PAPER AND PAPER PRODUCTS
49	S&P CNX PERSONAL CARE
50	S&P CNX PESTICIDES AND AGROCHEMICALS
51	S&P CNX PETROCHEMICALS
52	S&P CNX PHARMACEUTICALS
53	S&P CNX PLASTIC AND PLASTIC PRODUCTS
54	S&P CNX POWER
55	S&P CNX PRINTING AND PUBLISHING
56	S&P CNX REFINERIES
57	S&P CNX REFRACTORIES
58	S&P CNX SHIPPING
59	S&P CNX SOLVENT EXTRACTION
60	S&P CNX STEEL AND STEEL PRODUCTS
61	S&P CNX SUGAR
62	S&P CNX TEA AND COFFEE
63	S&P CNX TELECOMMUNICATION - EQUIPMENT
64	S&P CNX TELECOMMUNICATION - SERVICES
65	S&P CNX TEXTILE MACHINERY
66	S&P CNX TEXTILE PRODUCTS
67	S&P CNX TEXTILES - COTTON
68	S&P CNX TEXTILES - SYNTHETIC
69	S&P CNX TRADING
70	S&P CNX TRANSMISSION TOWERS
71	S&P CNX TRAVEL AND TRANSPORT
72	S&P CNX TYRES

---



### *Customized Indices*

The customized indices can be sub-sets of existing indices or a completely new index. Some of the indices that can be constructed include for example sector indices, individual business group indices, portfolios, or industry indices. See

[http://www.nseindia.com/content/indices/ind\\_custindices.htm](http://www.nseindia.com/content/indices/ind_custindices.htm)

## 1.5 REFERENCES AND INDEX FACT SHEETS

NSE offers fact sheets to some selected indices. These include the S&P CNX 500, the S&P CNX Nifty, the CNX Nifty Junior, the CNX Midcap, the CNX IT sector Index, the CNX 100 Index, and the CNX Bank Index. Here are the links to download the brochures.

[http://www.nseindia.com/content/indices/ind\\_cnx\\_500.pdf](http://www.nseindia.com/content/indices/ind_cnx_500.pdf)  
[http://www.nseindia.com/content/indices/ind\\_cnx\\_nifty.pdf](http://www.nseindia.com/content/indices/ind_cnx_nifty.pdf)  
[http://www.nseindia.com/content/indices/ind\\_cnx\\_junior.pdf](http://www.nseindia.com/content/indices/ind_cnx_junior.pdf)  
[http://www.nseindia.com/content/indices/ind\\_cnx\\_midcap.pdf](http://www.nseindia.com/content/indices/ind_cnx_midcap.pdf)  
[http://www.nseindia.com/content/indices/ind\\_cnx\\_it.pdf](http://www.nseindia.com/content/indices/ind_cnx_it.pdf)  
[http://www.nseindia.com/content/indices/ind\\_cnx\\_100.pdf](http://www.nseindia.com/content/indices/ind_cnx_100.pdf)  
[http://www.nseindia.com/content/indices/ind\\_cnx\\_bank.pdf](http://www.nseindia.com/content/indices/ind_cnx_bank.pdf)

## 1.6 NSE LEGAL ISSUES

Historical data can be used and downloaded on the basis of the indices and index components. This Ebook gives in the following hints how to use the available information from the NSE Internet portal for educational purposes, for teachers and researchers who have no other access to financial market data and their universities cannot afford a feed from a commercial provider. It is not allowed to use the code presented in this book for any commercial purposes. It is also forbidden to distribute code and code snippets from this eBook without written permission from the Rmetrics Association. It is not allowed to use the code and code snippets in any case which violates the NSE legal issues.

Please note, historical data can only be used within the legal framework specified by NSE. Before you continue to use this eBook, visit the NSE Internet portal and read carefully the *Legal Issues* of NSE to use their data.

## CHAPTER 2

# DOWLOADING NSE INDICES

```
> library(fBasics)
> library(fImport)
```

### 2.1 THE DOWNLOAD URL

The following table lists the download names for the 15 indices which can be downloaded from NSE

LISTING 2.1: NSE SYMBOLS FOR INDEX TIME SERIES

---

Name:	Description:
"S&P CNX NIFTY"	S&P CNX Nifty Index
"CNX NIFTY JUNIOR"	CNX Nifty Junior
"S&P CNX DEFTY"	S&P CNX Defty Index
"S&P CNX 500"	S&P CNX 500 Index
"CNX MIDCAP 200"	CNX Midcap 200 Index
"CNX MIDCAP"	CNX Midcap Index
"CNX IT"	CNX IT Index
"BANK NIFTY"	BANK NIFTY Index
"CNX 100"	CNX 100 Index
"NIFTY MIDCAP 50"	NIFTY MIDCAP 50 Index
"S&P ESG INDIA INDEX"	S&P ESG India Index
"S&P CNX NIFTY SHARIAH"	S&P CNX NIFTY Shariah Index
"S&P CNX 500 SHARIAH"	S&P CNX 500 Shariah Index
"CNX INFRASTRUCTURE"	CNX Infrastructure Index
"CNX REALTY"	CNX Realty Index

---

The download URL is then composed in a two step process given the name of the desired index.

*Example: Compose the URL of the CNX IT Index*

First let us download the web page which keeps the download URL

```
> NAME <- "CNX IT"
> from <- Sys.Date() - 5 * 366
> to <- Sys.Date()
> FROM <- paste(substr(format(from), 9, 10), substr(format(from),
  6, 7), substr(format(from), 1, 4), sep = "-")
> TO <- paste(substr(format(to), 9, 10), substr(format(to), 6,
  7), substr(format(to), 1, 4), sep = "-")
> URL1 <- composeURL("www.nseindia.com//marketinfo/indices/histdata/historicalindices.jsp?",
  "IndexType=", NAME, "&FromDate=", FROM, "&ToDate=", TO)
```

Then we extract the download URL from this page

```
> URL = character(0)
> while (length(URL) == 0) {
  urlInfo <- read lynx(URL1)
  URL <- substring(indexGrep("/content/indices/histdata/",
    urlInfo), 7)
}
> URL
[1] "http://www.nseindia.com/content/indices/histdata/CNX%20IT25-05-2005-29-05-2010.csv"
```

## 2.2 DOWNLOADING AN INDEX TIME SERIES

In the previous section we have composed the download URL. Now we are ready to fetch the data

```
> Download <- NA
> class(Download) <- "try-error"
> while (class(Download) == "try-error") {
  Download <- try(read lynx(URL), silent = TRUE)
}
> Download <- gsub("\\", "", Download)[-1]
> Download <- Download[Download != ""]
> head(Download)
[1] "25-May-2005, 2791.60, 2844.20, 2772.80, 2828.55, 10108349, 741.59"
[2] "26-May-2005, 2770.65, 2903.55, 2769.40, 2888.95, 17011201, 1137.63"
[3] "27-May-2005, 2903.85, 2926.45, 2880.00, 2896.55, 13883763, 829.14"
[4] "30-May-2005, 2883.90, 2925.25, 2881.95, 2899.55, 10198935, 750.13"
[5] "31-May-2005, 2900.45, 2938.45, 2890.00, 2933.00, 8714418, 586.67"
[6] "01-Jun-2005, 2933.50, 2933.65, 2894.35, 2900.25, 11120182, 584.56"
```

In the next step we transform the downloaded data into an object of class `timeSeries`

```
> CNX.IT <- timeSeries(data = dataSplit(Download, split = ",",
  col = c(2:5, 7)), charvec = format(charvecSplit(Download,
  split = ",", col = 1, format = "%d-%B-%Y")), units = c(paste(NAME,
  c("0", "H", "L", "C"), sep = "."), "Turnover"))
> start(CNX.IT)
```

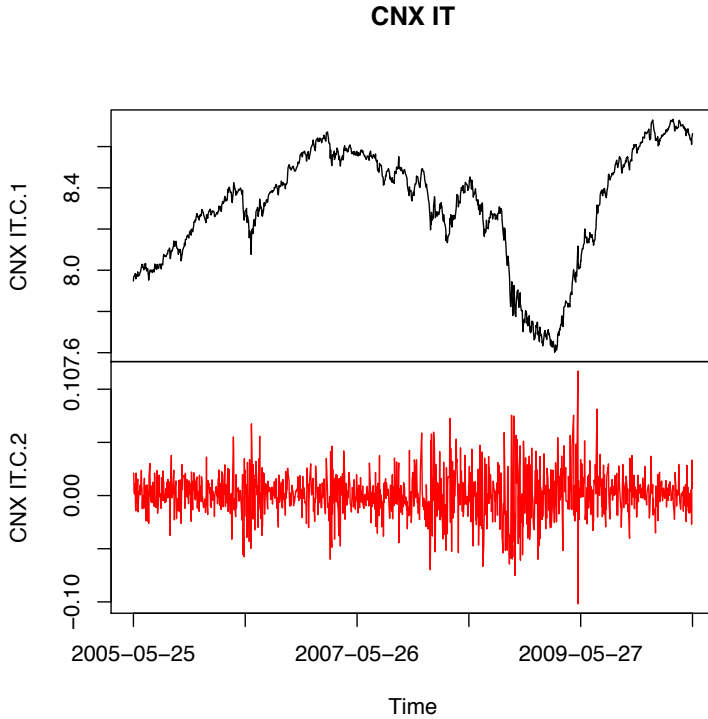


FIGURE 2.1: CNX IT - Log Index and log Returns

```

GMT
[1] [2005-05-25]
> tail(CNX.IT)
GMT
      CNX IT.0 CNX IT.H CNX IT.L CNX IT.C Turnover
2010-05-21  5618.1  5639.6  5554.0  5606.4   677.62
2010-05-24  5622.7  5702.4  5622.6  5638.5   584.48
2010-05-25  5592.4  5592.4  5446.8  5488.1   651.19
2010-05-26  5528.8  5702.4  5528.8  5674.4   677.73
2010-05-27  5671.6  5740.9  5623.3  5713.6  1232.41
2010-05-28  5721.2  5792.2  5709.1  5783.6   660.00

```

Have a look on the data and plot the series of the log indices and log returns of the closing prices.

```

> CNX.IT2 <- cbind(log(CNX.IT[, 4]), returns(CNX.IT[, 4]))
> plot(CNX.IT2, main = "CNX IT")

```

2.3 THE FUNCTION `nseIndexDownload()`

We can put together the above code snippets and write a function to download indices from the NSE.

```
> nseIndexDownload <- function(name, from = Sys.Date() - 366, to = Sys.Date()) {
  NAME = gsub(" ", "%20", name)
  FROM <- paste(substr(format(from), 9, 10), substr(format(from),
    6, 7), substr(format(from), 1, 4), sep = "-")
  TO <- paste(substr(format(to), 9, 10), substr(format(to),
    6, 7), substr(format(to), 1, 4), sep = "-")
  URL1 <- composeURL("www.nseindia.com/marketinfo/indices/histdata/historicalindices.jsp?",
    "IndexType=", NAME, "&FromDate=", FROM, "&ToDate=", TO)
  URL <- character(0)
  while (length(URL) == 0) {
    urlInfo = read.lynx(URL1)
    URL <- substring(indexGrep("/content/indices/histdata/",
      urlInfo), 7)
  }
  Download <- NA
  class(Download) <- "try-error"
  while (class(Download) == "try-error") {
    Download <- try(read.lynx(URL), silent = TRUE)
  }
  Download <- gsub("\\", "", Download)[-1]
  Download <- Download[Download != ""]
  tS <- timeSeries(data = dataSplit(Download, split = ",",
    col = c(2:5, 7)), charvec = format(charvecSplit(Download,
    split = ",", col = 1, format = "%d-%B-%Y")), units = c(paste(NAME,
    c("O", "H", "L", "C"), sep = "."), "Turnover"))
  tS
}
```

*Exercise: Download the S&P CNX 500 Index*

Try it, and download the time series for the S&P CNX 500 Index.

```
> NAME <- "S&P CNX 500"
```

## CHAPTER 3

# DOWLOADING NSE EQUITIES

```
> library(fBasics)
> library(fImport)
```

### 3.1 THE DOWNLOAD URL

Every index and equity has an internal code, we call it NAME. For example the NSE Symbol for the NTPC equity is NTPC.

```
> NAME <- "NTPC"
```

Next we have also to specify the download range

*Example: Compose the URL for a Fixed Period Download Range*

There are several options to specify a fixed download range. These include

LISTING 3.1: FIXED RANGES FOR DOWNLOADS FROM NSE

---

day	1 Day
week	7 Days
15days	15 Days
1month	30 Days
3month	90 Days
12month	365 Days
24month	24 Months

---

Let us construct the URL to fetch the download URL for the data from this file

```

> NAME <- "NTPC"
> RANGE <- "week"
> URL1 <- composeURL("www.nseindia.com/marketinfo/sym_map/symbolMapping.jsp?",
  "dataType=priceVolume", "&symbol=", NAME, "&segmentLink=3",
  "&symbolCount=1", "&dateRange=", RANGE, "&series=EQ")
> urlInfo <- read.lynx(URL1)

```

Since it takes some time until the URL information file is downloaded we pass the command through a while loop until the download URL is fetched

```

> URL = character(0)
> while (length(URL) == 0) {
  URL <- substring(indexGrep("/datafiles/", urlInfo), 7)
}

```

*Example: Compose the URL for a General Range*

```

> from <- Sys.Date() - 366
> to <- Sys.Date()

```

With these settings we can compose the download URL. First we convert the period into the date format as used by NSE. For the download we need two URLs, the first for creating the data set and the second for downloading the data. We create the data on the server side calling the function `read.lynx()`

```

> NAME <- "NTPC"
> FROM <- paste(substr(from, 9, 10), substr(from, 6, 7), substr(from,
  1, 4), sep = "-")
> TO <- paste(substr(to, 9, 10), substr(to, 6, 7), substr(to, 1,
  4), sep = "-")
> URL1 <- composeURL("www.nseindia.com/marketinfo/sym_map/symbolMapping.jsp?",
  "dataType=priceVolume", "&symbol=", NAME, "&segmentLink=3",
  "&symbolCount=1", "&dateRange=day", "&series=EQ", "&fromDate=",
  FROM, "&toDate=", TO)
> URL = character(0)
> while (length(URL) == 0) {
  urlInfo <- read.lynx(URL1)
  URL <- substring(indexGrep("/datafiles/", urlInfo), 7)
}
> URL
[1] "http://www.nseindia.com/content/equities/scripvol/datafiles/15-05-2009-TO-16-05-2010NTPCEQN.csv"

```

## 3.2 DOWNLOADING EQUITY TIME SERIES

Knowing the URL we can download the historical data records of the time series. The best way to do it is to use the function `read.csv()`.

*Example: Download the NTPC Time Series*

Now download the .csv file with the NCTP data for the last 366 days. We use the R function `R.csv()`. The connection to the NSE data server is sometimes very weak, and so we pass the `read.csv()` function through a while loop until the data file will be transferred.

```
> Download <- NA
> class(Download) <- "try-error"
> while (class(Download) == "try-error") {
  Download <- try(read.Lynx(URL), silent = TRUE)
}
> Download <- gsub("\\\"", "", Download)[-1]
> Download <- Download[Download != ""]
```

The file has 14 columns. Let us extract Open, High, Low, Close and Turnover

```
> data = dataSplit(Download, split = ",", col = c(5:7, 9, 11))
> charvec = charvecSplit(Download, split = ",", col = 3, format = "%d-%B-%Y")
> tS <- timeSeries(data = data, charvec = format(charvec), units = c(paste(NAME,
  c("O", "H", "L", "C"), sep = "."), "Turnover"))
> start(tS)
GMT
[1] [2009-05-15]
> tail(tS)
GMT
      NTPC.O NTPC.H NTPC.L NTPC.C Turnover
2010-05-07 204.05 206.00 201.20 202.60 7200734
2010-05-10 204.00 205.95 203.00 205.25 1780205
2010-05-11 206.00 206.00 203.15 204.90 2485430
2010-05-12 204.50 206.50 204.05 206.25 1968672
2010-05-13 205.50 208.45 205.50 207.75 3376878
2010-05-14 207.75 208.20 205.15 205.85 2555638
```

3.3 THE FUNCTION `nseEquityDownload()`

The code snippets above for downloading can be summarized in a function `nseEquityDownload()`. The next example shows how to write such a function

*Example: Write the Download Function*

```
> nseEquityDownload <- function(name, from = Sys.Date() - 366,
  to = Sys.Date()) {
  URL = character(0)
  NAME = name
  FROM = paste(substr(format(from), 9, 10), substr(format(from),
    6, 7), substr(format(from), 1, 4), sep = "-")
  TO = paste(substr(format(to), 9, 10), substr(format(to),
    6, 7), substr(format(to), 1, 4), sep = "-")
```



```

URL1 = composeURL("www.nseindia.com/marketinfo/sym_map/symbolMapping.jsp?",
  "dataType=priceVolume", "&symbol=", NAME, "&segmentLink=3",
  "&symbolCount=1", "&dateRange=day", "&series=EQ", "&fromDate=",
  FROM, "&toDate=", TO)
while (length(URL) == 0) {
  urlInfo <- read lynx(URL1)
  URL <- substring(indexGrep("/datafiles/", urlInfo), 7)
}
Download <- NA
class(Download) <- "try-error"
while (class(Download) == "try-error") {
  Download <- try(read.lynx(URL), silent = TRUE)
}
Download <- gsub("\\", "", Download)[-1]
Download <- Download[Download != ""]
data = dataSplit(Download, split = ",", col = c(5:7, 9, 11))
charvec = charvecSplit(Download, split = ",", col = 3, format = "%d-%B-%Y")
tS <- timeSeries(data = data, charvec = format(charvec),
  units = c(paste(NAME, c("O", "H", "L", "C"), sep = "."),
  "Turnover"))
tS
}

```

### *Example: Download the NTPC Time Series Data*

Use the function to download "NTPC" for the last 5 years

```

> NTPC <- nseEquityDownload("NTPC", from = Sys.Date() - 5 * 366)
> colnames(NTPC)
[1] "NTPC.O"   "NTPC.H"   "NTPC.L"   "NTPC.C"   "Turnover"
> start(NTPC)
GMT
[1] [2005-05-12]
> end(NTPC)
GMT
[1] [2010-05-14]

```

Plot the log closing price, the log returns, the High-Low spread, and the turnover

```

> NTPC.TS <- cbind(logPrice = log(NTPC[, 4]), logReturns = returns(NTPC[,
  4]), HLSpred = NTPC[, 2] - NTPC[, 3], Turnover = NTPC[, 5])
> plot(NTPC.TS, main = "NTPC")

```

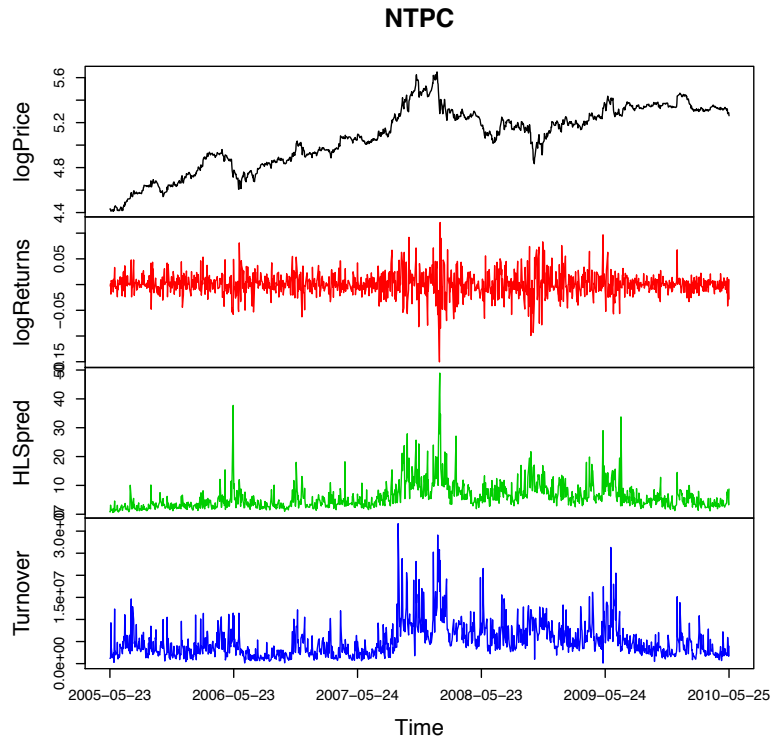


FIGURE 3.1: NTPC Equity Series

## CHAPTER 4

# DOWLOADING NSE DEBT SERIES

```
> library(fBasics)
> library(fImport)
```

### 4.1 THE DOWNLOAD URL

Historical Data on the debt market can be accessed following the menu Link: DEBT on NSE's home page. Three segments of the debt market data are available

---

#### LISTING 4.1: NSE DEBT SEGMENTS

---

Corporate Bonds  
Wholesale Debt Market  
Retail Debt Market

---

#### *The Wholesale Debt Market*

The *Wholesale Debt Market*, WDM, segment deals in fixed income securities and is rapidly gaining ground in an environment that has largely focussed on equities. The WDM segment of NSE started operations in June 1994. For more information we refer to

[http://www.nseindia.com/content/debt/debt\\_introduction.htm](http://www.nseindia.com/content/debt/debt_introduction.htm)

*The MIBID and MIBOR Download URL*

The historical time series files for the MIBIB and MIBOR rates can be downloaded as zipped .csv files from the following web page

```
http://www.nseindia.com/marketinfo/eod_information/bidbor.jsp
```

The files for the different maturities, overnight, two weeks, one month, and one quarter are

```
http://www.nseindia.com/content/debt/Overnight.zip
http://www.nseindia.com/content/debt/14day.zip
http://www.nseindia.com/content/debt/1month.zip
http://www.nseindia.com/content/debt/3month.zip
```

You have to download these files, to unzip them, and finally to clean the .csv

## 4.2 DOWNLOAD DEBT TIME SERIES

*Example: Load the Overnight Rates*

Compose the download URL for the overnight rates

```
> DEBT <- "Overnight"
> URL <- composeURL("www.nseindia.com/content/debt/", DEBT, ".zip")
> URL
[1] "http://www.nseindia.com/content/debt/Overnight.zip"
```

Download the zip file and unzip it

```
> download.file(URL, paste(DEBT, "zip", sep = "."))
> unzip(paste(DEBT, "zip", sep = "."))
```

Then convert the first spread sheet of the file into a .csv file

```
> sheet = 1
> package.dir <- .path.package("fImport")
> perl.dir <- file.path(package.dir, "perl")
> xls2csv <- file.path(perl.dir, "xls2csv.pl")
> xls <- paste(DEBT, "xls", sep = ".")
> csv <- paste(DEBT, "csv", sep = ".")
> cmd <- paste("perl", xls2csv, xls, csv, sheet, sep = " ")
> system(cmd, intern = TRUE)
[1] "Loading Overnight.xls ..."
[2] ""
[3] "Original Filename :Overnight.xls"
[4] "Number of Sheets :1"
[5] "Author          :agodkar"
[6] ""
[7] "Writing Sheet number 1 ('Overnight') to file 'Overnight.csv'"
[8] "Minrow=0 Maxrow=3714 Mincol=0 Maxcol=255"
[9] " (Ignored 1 blank lines.)"
[10] ""
```

Load the .csv file, clean it up

```
> Download = read.lynx("Overnight.csv")
> download = gsub("[\\\" ]", "", Download)
> download = indexGrep("^[0-9]", download)
> download = download[-grep("[A-Z]", download)]
```

and convert it into a timeSeries object

```
> ON <- timeSeries(data = dataSplit(download, split = ",", col = 2:5),
  charvec = charvecSplit(download, split = ",", format = "%d-%b-%y"),
  units = paste("ON", c("MIBID", "MIBID.SD", "MIBOR", "MIBOR.SD"),
  sep = "-")
> start(ON)
GMT
[1] [1998-06-15]
> tail(ON)
GMT
      ON-MIBID ON-MIBID.SD ON-MIBOR ON-MIBOR.SD
2010-05-22   3.83      0.0113   3.90      0.0088
2010-05-24   4.08      0.0216   4.21      0.0128
2010-05-25   4.03      0.0156   4.14      0.0166
2010-05-26   4.00      0.0036   4.11      0.0101
2010-05-28   4.60      0.0580   4.72      0.0665
2010-04-01    NA         NA      NA         NA
```

The first column is the overnight MIBID and the third the overnight MIBOR rate. The second and third columns are their standard deviations. Now let us plot the time series

```
> plot(ON, main = "MIBID and MIBOR Overnight Rates")
```

### 4.3 THE FUNCTION `nseDebtDownload()`

We can put together the code snippets from the previous section and write a download function for the debt instruments.

*Exercise: Write a Download Function*

Hint: Have a look how we have written the functions `nseIndexDownload()` and `nseEquityDownload()`.

### 4.4 NSE CLOSING DATES

Note, the download files of the overnight rates also contain all NSE holidays and closing days of the exchange, i. e. the dates of missing data records. We can use this information to build a BSE closing calendar.

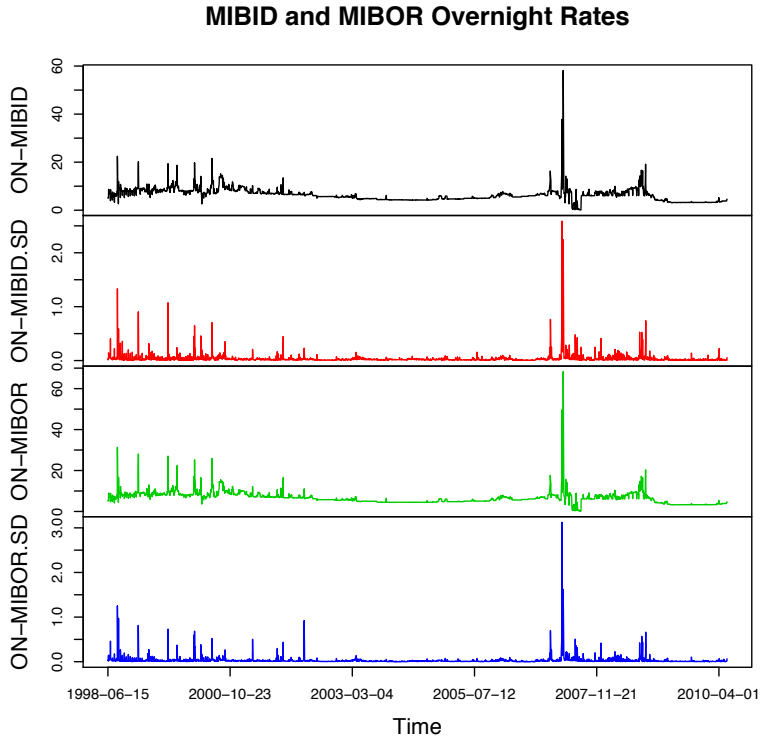


FIGURE 4.1: Overnight Interest Rates

*Example: Holiday Calendar 2005*

```

> download = gsub("\\", "", Download)
> download = gsub(" ", "", download)
> download = gsub(" ", " ", download)
> download = gsub(", ", " ", download)
> download = indexGrep("[A-Z]", download)
> download = download[-c(1, grep("MIBID/MIBOR", download))]

```

After we have extracted the closing days we can compose the calendar starting January 2000.

```

> charvec <- charvecSplit(download, split = " ", format = "%d-%b-%y")
> holidays <- tableSplit(download, split = " ", col = 2)
> holidays <- gsub(" ", "", holidays)
> holidayListing <- indexGrep("^2", paste(charvec, holidays))

```

Let us print the Calendar for the year 2005

```

> holidays2005 <- indexGrep("^2005", holidayListing)

```

```
> holidays2005
```

```
[1] "2005-01-21 Bakri ID"  
[2] "2005-01-26 Republic Day"  
[3] "2005-03-08 Mahashivratri"  
[4] "2005-03-25 Good Friday"  
[5] "2005-03-26 Holi (2nd day)"  
[6] "2005-04-09 Gudhi Padva"  
[7] "2005-04-14 Dr. Babasaheb Ambedkar Jayanti"  
[8] "2005-04-18 Ram Navami"  
[9] "2005-04-22 Mahavir Jayanti"  
[10] "2005-05-23 Buddha Pournima"  
[11] "2005-08-15 Independence Day"  
[12] "2005-08-20 Parsi New Year"  
[13] "2005-09-07 Ganesh Chaturthi"  
[14] "2005-10-12 Dasara "  
[15] "2005-11-01 Diwali Amavasya (Laxmi Pujan)"  
[16] "2005-11-05 Ramzan-Id (ID-Ul-Fitar)"  
[17] "2005-11-15 Guru Nanak Jayanti"
```

## PART II

# NSE LISTING FUNCTIONS



# CHAPTER 5

## INDEX CATEGORIZATION

```
> library(fBasics)
> library(fImport)
```

### 5.1 CATEGORIZATION SUMMARY

The Index categorization of the NSE Indices can be found on the web page

[http://www.nseindia.com/content/indices/ind\\_majorindices.htm](http://www.nseindia.com/content/indices/ind_majorindices.htm)

The NSE Equity Indices are categorized as major and other indices. In the following we briefly summarize the categorization scheme given on the mentioned web page, for details we recommend to visit the page.

### 5.2 MAJOR INDEX CATEGORIES

The categories for the *Major Indices* are

LISTING 5.1: NSE MAJOR INDEX CATEGORIZATION SCHEME

---

```
S&P CNX Nifty
CNX Nifty Junior
CNX 100
S&P CNX 500
CNX Midcap
Nifty Midcap 50
S&P CNX Defty
```

---

### 5.3 OTHER INDEX CATEGORIES

The categories for the *Other Indices* are

---

LISTING 5.2: NSE OTHER INDEX CATEGORIZATION SCHEME

---

CNX IT Index  
CNX Bank Index  
CNX FMCG Index  
CNX PSE Index  
CNX MNC Index  
CNX Service Sector Index  
S&P CNX Industry Indices  
Customised Indices  
CNX Energy Index  
CNX Pharma Index  
CNX Infrastructure Index  
CNX PSU BANK Index  
CNX Realty Index  
S&P CNX Nifty Shariah / S&P CNX 500 Shariah  
S&P ESG India Index

---

## CHAPTER 6

# NSE MARKET INDEX LISTINGS

```
> library(fBasics)
> library(fImport)
```

### 6.1 THE NIFTY INDEX LISTING

In the case of the NIFTY index we download and create a listing for the equities in the following way. First let us visit the Web page from where we get the information. Start at the NSE home page and follow the Link: Indices, Link: IISL Indices, Link: S&P CNX Nifty, and Link: List of S&P CNX Nifty stocks. You will end up at the web page

```
http://www.nseindia.com/content/indices/ind\_niftylist.csv
```

#### *Example: NIFTY Index Listing*

To create a listing we download this page using the function `readLines()`

```
> URL <- composeURL("www.nseindia.com/content/indices/ind_niftylist.csv")
> Download <- readLines(URL)
```

The first few lines of the downloaded `*.csv` file are

```
> head(Download, 10)
[1] ""
[2] "Constituents list of S&P CNX Nifty"
[3] ""
[4] "Company Name,Industry,Symbol,Series,ISIN Code"
[5] ""
[6] "ABB Ltd.,ELECTRICAL EQUIPMENT,ABB,EQ,INE117A01022"
[7] "ACC Ltd.,CEMENT AND CEMENT PRODUCTS,ACC,EQ,INE012A01025"
```

```
[8] "Ambuja Cements Ltd.,CEMENT AND CEMENT PRODUCTS,AMBUJACEM,EQ,INE079A01024"
[9] "Axis Bank Ltd.,BANKS,AXISBANK,EQ,INE238A01026"
[10] "Bharat Heavy Electricals Ltd.,ELECTRICAL EQUIPMENT,BHEL,EQ,INE257A01018"
```

Then we skip the first 5 lines and create a table from the third, fifth, and first column using the function `tableSplit()`

```
> download <- tableSplit(Download, split = ",", col = c(3, 5, 1))
> head(download[-(1:5), ])
      [,1]      [,2]      [,3]
[1,] "ABB"      "INE117A01022" "ABB Ltd."
[2,] "ACC"      "INE012A01025" "ACC Ltd."
[3,] "AMBUJACEM" "INE079A01024" "Ambuja Cements Ltd."
[4,] "AXISBANK" "INE238A01026" "Axis Bank Ltd."
[5,] "BHEL"      "INE257A01018" "Bharat Heavy Electricals Ltd."
[6,] "BPCL"      "INE029A01011" "Bharat Petroleum Corporation Ltd."
```

The table can be beautified when we transform it into a data frame and add appropriate column names

```
> niftyListing <- as.data.frame(download[-(1:5), ], stringsAsFactors = FALSE)
> colnames(niftyListing) <- c("Symbol", "ISIN", "Description")
> niftyListing
      Symbol      ISIN      Description
1      ABB INE117A01022      ABB Ltd.
2      ACC INE012A01025      ACC Ltd.
3  AMBUJACEM INE079A01024  Ambuja Cements Ltd.
4  AXISBANK INE238A01026      Axis Bank Ltd.
5      BHEL INE257A01018  Bharat Heavy Electricals Ltd.
6      BPCL INE029A01011  Bharat Petroleum Corporation Ltd.
7  BHARTIARTL INE397D01024  Bharti Airtel Ltd.
8      CAIRN INE910H01017  Cairn India Ltd.
9      CIPLA INE059A01026  Cipla Ltd.
10     DLF INE271C01023  DLF Ltd.
11     GAIL INE129A01019  GAIL (India) Ltd.
12    HCLTECH INE860A01027  HCL Technologies Ltd.
13    HDFCBANK INE040A01018  HDFC Bank Ltd.
14  HEROHONDA INE158A01026  Hero Honda Motors Ltd.
15  HINDALCO INE038A01020  Hindalco Industries Ltd.
16  HINDUNILVR INE030A01027  Hindustan Unilever Ltd.
17     HDFC INE001A01028  Housing Development Finance Corporation Ltd.
18     ITC INE154A01025  I T C Ltd.
19  ICICIBANK INE090A01013  ICICI Bank Ltd.
20     IDEA INE669E01016  Idea Cellular Ltd.
21  INFOSYSTCH INE009A01021  Infosys Technologies Ltd.
22     IDFC INE043D01016  Infrastructure Development Finance Co. Ltd.
23  JPASSOCIAT INE455F01025  Jaiprakash Associates Ltd.
24  JINDALSTEL INE749A01030  Jindal Steel & Power Ltd.
25  KOTAKBANK INE237A01010  Kotak Mahindra Bank Ltd.
26     LT INE018A01030  Larsen & Toubro Ltd.
27     M&M INE101A01026  Mahindra & Mahindra Ltd.
28  MARUTI INE585B01010  Maruti Suzuki India Ltd.
29     NTPC INE733E01010  NTPC Ltd.
30     ONGC INE213A01011  Oil & Natural Gas Corporation Ltd.
```

31	POWERGRID	INE752E01010	Power Grid Corporation of India Ltd.
32	PNB	INE160A01014	Punjab National Bank
33	RANBAXY	INE015A01028	Ranbaxy Laboratories Ltd.
34	RELCAPITAL	INE013A01015	Reliance Capital Ltd.
35	RCOM	INE330H01018	Reliance Communications Ltd.
36	RELIANCE	INE002A01018	Reliance Industries Ltd.
37	RELINFRA	INE036A01016	Reliance Infrastructure Ltd.
38	RPOWER	INE614G01033	Reliance Power Ltd.
39	SIEMENS	INE003A01024	Siemens Ltd.
40	SBIN	INE062A01012	State Bank of India
41	SAIL	INE114A01011	Steel Authority of India Ltd.
42	STER	INE268A01031	Sterlite Industries (India) Ltd.
43	SUNPHARMA	INE044A01028	Sun Pharmaceutical Industries Ltd.
44	SUZLON	INE040H01021	Suzlon Energy Ltd.
45	TCS	INE467B01029	Tata Consultancy Services Ltd.
46	TATAMOTORS	INE155A01014	Tata Motors Ltd.
47	TATAPOWER	INE245A01013	Tata Power Co. Ltd.
48	TATASTEEL	INE081A01012	Tata Steel Ltd.
49	UNITECH	INE694A01020	Unitech Ltd.
50	WIPRO	INE075A01022	Wipro Ltd.

*Example: Download the TATASTEEL Time Series*

Now we are ready to download the time series for all equities which are part of the Nifty Index.

```
> symbols <- niftyListing[, 1]
> symbols
 [1] "ABB"      "ACC"      "AMBUJACEM" "AXISBANK" "BHEL"
 [6] "BPCL"    "BHARTIARTL" "CAIRN"     "CIPLA"    "DLF"
[11] "GAIL"    "HCLTECH"  "HDFCBANK"  "HEROHONDA" "HINDALCO"
[16] "HINDUNILVR" "HDFC"     "ITC"       "ICICIBANK" "IDEA"
[21] "INFOSYSTCH" "IDFC"     "JPASSOCIAT" "JINDALSTEL" "KOTAKBANK"
[26] "LT"      "M&M"     "MARUTI"    "NTPC"     "ONGC"
[31] "POWERGRID" "PNB"     "RANBAXY"   "RELCAPITAL" "RCOM"
[36] "RELIANCE" "RELINFRA" "RPOWER"    "SIEMENS"   "SBIN"
[41] "SAIL"     "STER"    "SUNPHARMA" "SUZLON"    "TCS"
[46] "TATAMOTORS" "TATAPOWER" "TATASTEEL" "UNITECH"   "WIPRO"
```

As an example let us download the TATASTEEL time series

```
> TATASTEEL = nseEquityDownload(name = "TATASTEEL", from = Sys.Date() -
  5 * 366)
```

and plot the absolute values of the returns and the turnover

```
> TATASTEEL.TS = na.omit(cbind(logReturns = abs(100 * returns(TATASTEEL[,
  4])), Turnover = TATASTEEL[, 5]))
> plot(TATASTEEL.TS, main = "TATASTEEL")
```

Finally compute the correlation between the absolute values of the log returns and the turnover

```
> cor(TATASTEEL.TS)
```

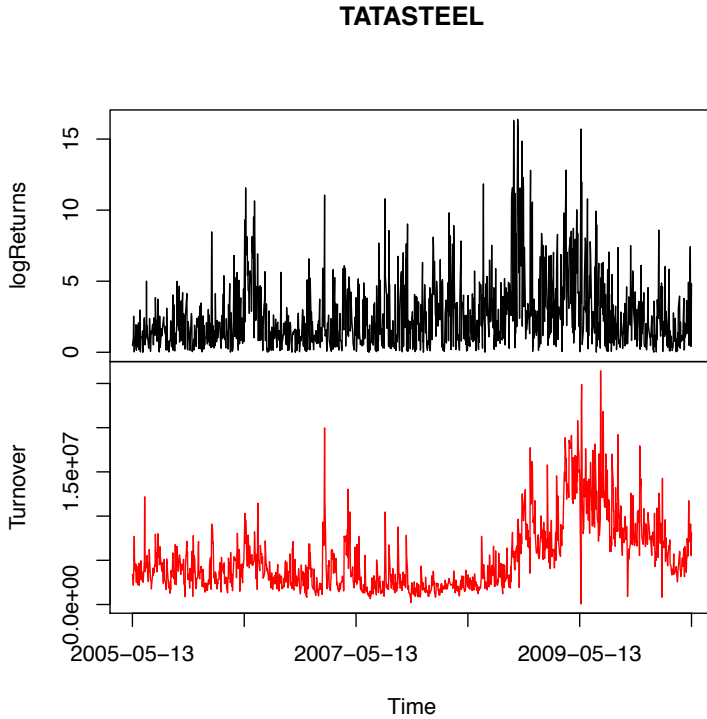


FIGURE 6.1: TATASTEEL - Log Returns and Turnover

```

      logReturns Turnover
logReturns  1.00000  0.37523
Turnover    0.37523  1.00000

```

## 6.2 THE FUNCTION `nseListing()`

We can now compose the code snippets into a function `nseListing()` to create an index listing

```

> nseListing <- function(index = "nifty") {
  URL <- composeURL("www.nseindia.com/content/indices/ind_",
    index, "list.csv")
  Download <- readLines(URL)
  download <- tableSplit(Download, split = ",", col = c(3,
    5, 1))
  Listing <- as.data.frame(download[-(1:5), ], stringsAsFactors = FALSE)
  colnames(Listing) <- c("Symbol", "ISIN", "Description")
  Listing
}

```

```
}

```

Try it to download the ISIN codes of the Nifty components

```
> ISIN = sort(nseListing("nifty")[, 2])
> ISIN
 [1] "INE001A01028" "INE002A01018" "INE003A01024" "INE009A01021" "INE012A01025"
 [6] "INE013A01015" "INE015A01028" "INE018A01030" "INE029A01011" "INE030A01027"
[11] "INE036A01016" "INE038A01020" "INE040A01018" "INE040H01021" "INE043D01016"
[16] "INE044A01028" "INE059A01026" "INE062A01012" "INE075A01022" "INE079A01024"
[21] "INE081A01012" "INE090A01013" "INE101A01026" "INE114A01011" "INE117A01022"
[26] "INE129A01019" "INE154A01025" "INE155A01014" "INE158A01026" "INE160A01014"
[31] "INE213A01011" "INE237A01010" "INE238A01026" "INE245A01013" "INE257A01018"
[36] "INE268A01031" "INE271C01023" "INE330H01018" "INE397D01024" "INE455F01025"
[41] "INE467B01029" "INE585B01010" "INE614G01033" "INE669E01016" "INE694A01020"
[46] "INE733E01010" "INE749A01030" "INE752E01010" "INE860A01027" "INE910H01017"
```

### 6.3 THE MAJOR INDEX LISTING FILES

#### *Summary of Major Listing Files*

For all major and other indices we can proceed in the same way. We have to find out the names of the NSE listing files. These are for the major indices

LISTING 6.1: NSE MAJOR INDEX LISTING FILES

---

Index:	Description:
nifty	NIFTTY
jrnifty	Junior NIFTY
cnx100	cnx 100
cnx500	S&P CNX 500
cnxmidcap	CNX Midcap
niftymidcap50	NIFTY Midcap 50
defty	S&P CNX Defty

---

*Example: Get the Listing for the CNX Junior Index*

```
> nseListing(index = "jrnifty")
      Symbol      ISIN      Description
1  ADANIENT INE423A01024 Adani Enterprises Ltd.
2  ABIRLANUVO INE069A01017 Aditya Birla Nuvo Ltd.
3  ANDHRABANK INE434A01013 Andhra Bank
4  ASHOKLEY INE208A01029 Ashok Leyland Ltd.
5  ASIANPAINT INE021A01018 Asian Paints Ltd.
6  BAJAJ - AUTO INE917I01010 Bajaj Auto Ltd.
7  BANKBARODA INE028A01013 Bank of Baroda
8  BANKINDIA INE084A01016 Bank of India
9  BEL INE263A01016 Bharat Electronics Ltd.
```

10	BHARATFORG	INE465A01025	Bharat Forge Ltd.
11	BIOCON	INE376G01013	Biocon Ltd.
12	CANBK	INE476A01014	Canara Bank
13	COLPAL	INE259A01022	Colgate Palmolive (India) Ltd.
14	CONCOR	INE111A01017	Container Corporation of India Ltd.
15	CORPBANK	INE112A01015	Corporation Bank
16	CROMPGREAV	INE067A01029	Crompton Greaves Ltd.
17	CUMMINSIND	INE298A01020	Cummins India Ltd.
18	DRREDDY	INE089A01023	Dr. Reddy's Laboratories Ltd.
19	FEDERALBNK	INE171A01011	Federal Bank Ltd.
20	GMRINFRA	INE776C01039	GMR Infrastructure Ltd.
21	GLAXO	INE159A01016	Glaxosmithkline Pharmaceuticals Ltd.
22	GLENMARK	INE935A01035	Glenmark Pharmaceuticals Ltd.
23	HINDPETRO	INE094A01015	Hindustan Petroleum Corporation Ltd.
24	HDIL	INE191I01012	Housing Development and Infrastructure Ltd.
25	IDBI	INE008A01015	IDBI Bank Ltd.
26	IFCI	INE039A01010	IFCI Ltd.
27	IBREALEST	INE069I01010	Indiabulls Real Estate Ltd.
28	INDHOTEL	INE053A01029	Indian Hotels Co. Ltd.
29	IOB	INE565A01014	Indian Overseas Bank
30	JSWSTEEL	INE019A01020	JSW Steel Ltd.
31	LICHSGFIN	INE115A01018	LIC Housing Finance Ltd.
32	LUPIN	INE326A01029	Lupin Ltd.
33	MRPL	INE103A01014	Mangalore Refinery & Petrochemicals Ltd.
34	MPHASIS	INE356A01018	Mphasis Ltd.
35	MUNDRAPORT	INE742F01034	Mundra Port and Special Economic Zone Ltd.
36	OFSS	INE881D01027	Oracle Financial Services Software Ltd.
37	PATNI	INE660F01012	Patni Computer Systems Ltd.
38	PFC	INE134E01011	Power Finance Corporation Ltd.
39	PUNJLLOYD	INE701B01021	Punj Lloyd Ltd.
40	RNRL	INE328H01012	Reliance Natural Resources Ltd.
41	SESAGOA	INE205A01025	Sesa Goa Ltd.
42	SYNDIBANK	INE667A01018	Syndicate Bank
43	TTML	INE517B01013	Tata Teleservices (Maharashtra) Ltd.
44	TECHM	INE669C01028	Tech Mahindra Ltd.
45	TORNTPOWER	INE813H01021	Torrent Power Ltd.
46	ULTRACEMCO	INE481G01011	UltraTech Cement Ltd.
47	UNIONBANK	INE692A01016	Union Bank of India
48	UNIPHOS	INE628A01036	United Phosphorus Ltd.
49	MCDOWELL-N	INE854D01016	United Spirits Ltd.
50	ZEEL	INE256A01028	Zee Entertainment Enterprises Ltd.

## 6.4 THE OTHER INDEX LISTING FILES

### *Summary of Other Listing Files*

For the other indices the listing file names are

LISTING 6.2: NSE OTHER INDEX LISTING FILES

---

Index:	Description:
cnxit	CNX IT Index



---

cnxbank	CNX Bank Index
cnxfmcg	CNX FMCG Index
cnxpse	CNX PSE Index
cnxmnc	CNX MNC Index
cnxssi	CNX Service Sector Index
indusindices	S&P CNX Industry Indices
cusindices	Customised Indices
cnxenergy	CNX Energy Index
cnxparma	CNX Pharma Index
cnxinfra	CNX Infrastructure Index
cnxpsubank	CNX PSU BANK Index
realty	CNX Realty Index
sharia	S&P CNX Nifty Shariah / S&P CNX 500 Shariah
ESG	S&P ESG India Index

---

*Example: Get the CNX Bank Index Listing*

Have a look on the bank index

```
> cnxbankListing <- nseListing(index = "cnxbank")
> cnxbankListing
```

	Symbol	ISIN	Description
1	AXISBANK	INE238A01026	Axis Bank Ltd.
2	BANKBARODA	INE028A01013	Bank of Baroda
3	BANKINDIA	INE084A01016	Bank of India
4	CANBK	INE476A01014	Canara Bank
5	HDFCBANK	INE040A01018	HDFC Bank Ltd.
6	ICICIBANK	INE090A01013	ICICI Bank Ltd.
7	IDBI	INE008A01015	IDBI Bank Ltd.
8	KOTAKBANK	INE237A01010	Kotak Mahindra Bank Ltd.
9	ORIENTBANK	INE141A01014	Oriental Bank of Commerce
10	PNB	INE160A01014	Punjab National Bank
11	SBIN	INE062A01012	State Bank of India
12	UNIONBANK	INE692A01016	Union Bank of India

PART III

RELATED DATA FROM OTHER  
SOURCES

## CHAPTER 7

# DATA FROM BOMBAY STOCK EXCHANGE

```
> library(fBasics)
> library(fImport)
```

### 7.1 BSE INTERNET PORTAL

The Bombay Stock Exchange is the oldest stock exchange in Asia, it was established in 1875. BSE is the world's number 1 exchange in terms of the number of listed companies. The BSE Index, SENSEX, is India's stock benchmark index and tracked worldwide. It constitutes 30 stocks representing 12 major sectors. BSE provides an efficient and transparent market for trading in equity, debt instruments and derivatives. The BSE Internet Portal can be accessed through the following link

```
http://www.bseindia.co.in
```

### 7.2 DOWNLOADING EQUITY PRICES

Time series data can be downloaded on monthly ranges, the starting date is one month back given the current date through the variables `fdate`, `fmonth` and `fyear`. The equity is specified in the variables `text1` and `main1`. With this information we can compose the download URL. For *Hindalco Industries Ltd* the symbol or so called *Scrip Code* is "500440"

```
> NAME = "500440"
> fdate = 31
> fmonth = 12
> fyear = 2009
> URL <- composeURL("www.bseindia.com/histdata/stockprc2.asp",
```

```
"?text1=", NAME, "&flag=sp", "&main1=", NAME, "&fdate=31",
"&fmonth=12", "&fyear=2009", "&main2=1", "&default1=1")
```

The URL becomes

```
> strsplit(gsub("&", " &", URL, perl = TRUE), " ")[[1]]
[1] "http://www.bseindia.com/histdata/stockprc2.asp?text1=500440"
[2] "&flag=sp"
[3] "&main1=500440"
[4] "&fdate=31"
[5] "&fmonth=12"
[6] "&fyear=2009"
[7] "&main2=1"
[8] "&default1=1"
```

Since the link is longer than one line, the output was splitted at the ampersands.

Then we download the data records using the function `read.lynx()` and extract the data records

```
> Download = read.lynx(URL)
> download = indexGrep("^ [0-9]", Download)
> download = gsub(" ", " ", download)
> download = gsub(" ", " ", download)
> download = download[-grep("http", download)]
> download = matrix(download, ncol = 2, byrow = TRUE)
> download = apply(download, 1, paste, collapse = " ")
```

The output of the downloaded data records has now the form

```
> head(download)
[1] " 30 November 2009 134.00 139.85 134.00 138.05 138.31 1991468 11385 275,447,073.00 5.85 4.05"
[2] " 1 December 2009 138.80 142.50 138.80 142.10 141.20 1768141 10877 249,662,107.00 3.70 3.30"
[3] " 2 December 2009 143.90 145.60 142.05 142.85 143.70 3170760 9368 455,639,665.00 3.55 -1.05"
[4] " 3 December 2009 143.95 147.70 143.65 146.30 146.31 1836891 9001 268,761,851.00 4.05 2.35"
[5] " 4 December 2009 146.00 150.35 145.00 147.30 148.31 2010912 11306 298,240,292.00 5.35 1.30"
[6] " 7 December 2009 148.15 148.75 140.25 142.55 142.67 2300479 12268 328,200,162.00 8.50 -5.60"
```

To download historical data for several months, we first write a download function for one month,

```
> bseMonthlyDownload <- function(name, fmonth, fyear) {
  URL <- composeURL("www.bseindia.com/histdata/stockprc2.asp",
    "?text1=", name, "&flag=sp", "&main1=", name, "&fdate=1",
    "&fmonth=", fmonth, "&fyear=", fyear, "&main2=1", "&default1=1")
  Download = read.lynx(URL)
  download = indexGrep("^ [0-9]", Download)
  download = gsub(" ", " ", download)
  download = gsub(" ", " ", download)
  download = download[-grep("http", download)]
  download = matrix(download, ncol = 2)
  download = matrix(download, ncol = 2, byrow = TRUE)
  download = apply(download, 1, paste, collapse = " ")
  return(download)
}
```

and then we loop over the desired months.

```
> NAME <- "500440"
> SEQ <- timeSequence(Sys.Date() - 183, Sys.Date() + 31, by = "month")
> Years <- atoms(SEQ)$Y
> Months <- atoms(SEQ)$m
> Download <- NULL
> for (i in 1:length(SEQ)) {
  Download = c(Download, bseMonthlyDownload(name = NAME, Months[i],
    Years[i]))
}
```

To obtain a clean file, we remove leading blanks, transform the date into the format "%d-%m-%Y", convert double blanks in single blanks, and remove the “comma” field separators

```
> download = sub("^ ", "", Download)
> download = sub(" ", "-", download)
> download = sub(" ", "-", download)
> download = gsub(" ", " ", download)
> download = gsub(" ", " ", download)
> download = gsub(",", "", download)
```

In the last step we split the matrix into a numeric data part and a character time stamp part to create the time series object.

```
> data <- dataSplit(download)
> charvec <- charvecSplit(download, format = "%d-%B-%Y")
> units <- c("Open", "High", "Low", "Close", "Average", "Shares",
  "Trades", "Turnover", "SpreadHL", "SpreadC0")
> HINDALCO <- timeSeries(data, charvec, units = units)
```

Let us have a look on the open, high, low, close, and turnover columns of the time series.

```
> start(HINDALCO)
GMT
[1] [2009-10-01]
> head(HINDALCO[, c(1:4, 8)])
GMT
      Open  High  Low  Close  Turnover
2009-10-01 130.0 130.00 125.70 126.70 175249866
2009-10-05 125.0 125.50 116.85 118.15 195716046
2009-10-06 122.9 125.95 118.50 125.30 384538292
2009-10-07 127.0 129.95 126.15 127.25 404491823
2009-10-08 128.0 132.00 127.00 129.25 298230389
2009-10-09 130.1 131.75 127.15 128.20 187777813
```

As a further example we plot the absolute values of the log returns which measures volatility and the high-low spreads.

```
> HINDALCO.TS <- cbind(Volatility = abs(returns(HINDALCO[, 4])),
  hLSpread = HINDALCO[, 9])
> plot(HINDALCO.TS, main = "HINDALCO")
```

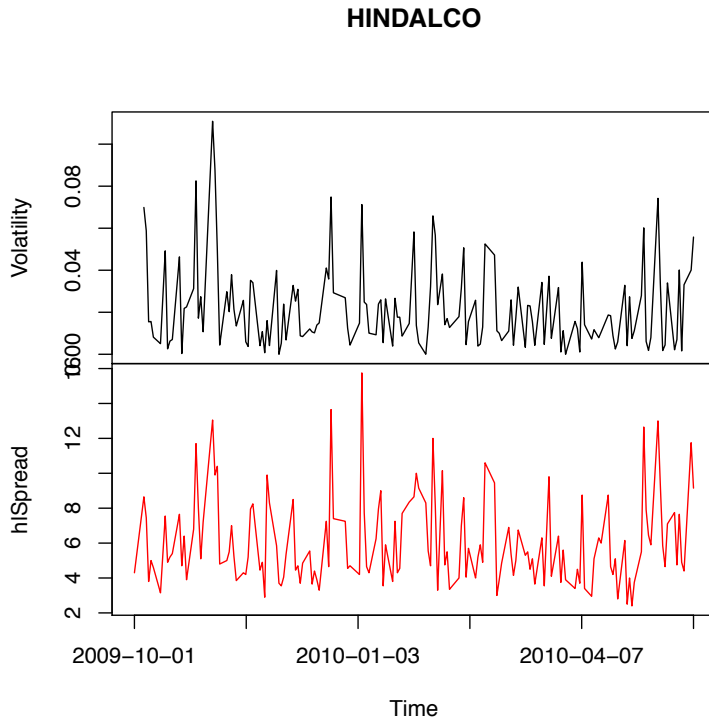


FIGURE 7.1: Volatility and High-Low Spreads for HINDALCO

Note, there is an evident correlation between these two variables.

```
> cor(na.omit(HINDALCO.TS))
      Volatility hlSpread
Volatility  1.00000  0.72887
hlSpread    0.72887  1.00000

> plot(series(HINDALCO.TS), pch = 19, col = "steelblue", main = "HINDALCO")
```

### 7.3 THE FUNCTION `bseDownload()`

From the code snippets of the previous section we can write a download function `bseDownload()`.

```
> bseDownload <- function(name, from = Sys.Date() - 366, to = Sys.Date()) {
  SEQ = timeSequence(from, to + 31, by = "month")
  Years = atoms(SEQ)$Y
  Months = atoms(SEQ)$m
```

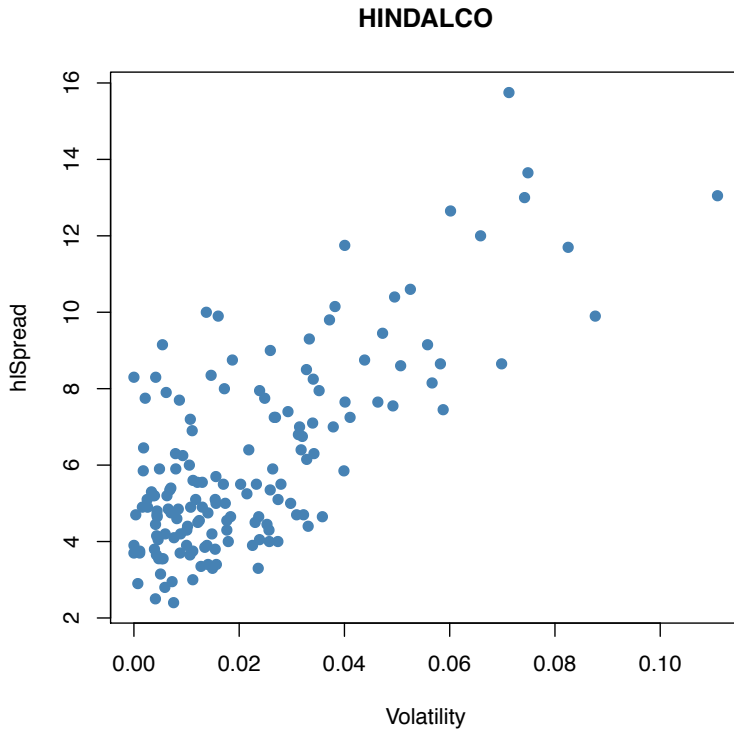


FIGURE 7.2: Volatility vs High-Low Spreads for HINDALCO

```
.bseMonthlyDownload <- function(name, fmonth, fyear) {
  URL <- composeURL("www.bseindia.com/histdata/stockprc2.asp",
    "?text1=", name, "&flag=sp", "&main1=", name, "&fdate=1",
    "&fmonth=", fmonth, "&fyear=", fyear, "&main2=1",
    "&default1=1")
  Download = read.lynx(URL)
  download = indexGrep("^ [0-9]", Download)
  download = gsub(" ", " ", download)
  download = gsub(" ", " ", download)
  download = download[-grep("http", download)]
  download = matrix(download, ncol = 2)
  download = matrix(download, ncol = 2, byrow = TRUE)
  download = apply(download, 1, paste, collapse = " ")
  return(download)
}
Download = NULL
for (i in 1:length(SEQ)) {
  Download = c(Download, .bseMonthlyDownload(name, Months[i],
    Years[i]))
}
```

```

}
download = sub("^ ", "", download)
download = sub(" ", "-", download)
download = sub(" ", "-.", download)
download = gsub(" ", " ", download)
download = gsub(" ", " ", download)
download = gsub(" ", " ", download)
data = dataSplit(download)
charvec = charvecSplit(download, format = "%d-%B-%Y")
units = c("Open", "High", "Low", "Close", "Average", "Shares",
         "Trades", "Turnover", "SpreadHL", "SpreadCO")
tS = timeSeries(data, charvec, units = units)
tS = tS[time(tS) >= as.timeDate(from), ]
tS = tS[time(tS) <= as.timeDate(to), ]
return(tS)
}

```

Try it, for the last two calendar weeks.

```

> Sys.Date()
[1] "2010-05-25"
> bseDownload("500440", from = Sys.Date() - 14)
GMT
      Open  High  Low  Close  Average  Shares  Trades  Turnover
2010-05-11 178.00 178.00 168.70 169.70  171.90 2531589  15658 435186466
2010-05-12 170.90 172.85 167.00 170.00  169.81 1888792  12978 320745147
2010-05-13 171.90 172.90 168.25 169.25  170.63  692782   5195 118208847
2010-05-14 168.00 169.60 162.50 163.60  165.81 1017829  6891 168765570
2010-05-17 160.25 166.25 158.50 163.95  163.03 1318351  9339 214929630
2010-05-18 164.00 165.00 160.25 162.80  162.41 1344632  8398 218376598
2010-05-19 160.00 162.75 155.10 156.40  159.28 1269337  8420 202173905
2010-05-20 157.25 159.15 154.25 156.65  156.74 1579041  8344 247494032
2010-05-21 152.50 153.30 148.90 151.55  151.12 1843109 10414 278522028
2010-05-24 154.00 156.00 144.25 145.60  150.67 1540533  9920 232107771
2010-05-25 142.00 144.90 135.75 137.70  140.67 2903269 18542 408406229
      SpreadHL  SpreadCO
2010-05-11     9.30    -8.30
2010-05-12     5.85    -0.90
2010-05-13     4.65    -2.65
2010-05-14     7.10    -4.40
2010-05-17     7.75     3.70
2010-05-18     4.75    -1.20
2010-05-19     7.65    -3.60
2010-05-20     4.90    -0.60
2010-05-21     4.40    -0.95
2010-05-24    11.75    -8.40
2010-05-25     9.15    -4.30

```

## 7.4 DOWNLOADING BSE INDICES

### *BSE Major Indices*

The 7 major indices are



LISTING 7.1: BSE MAJOR INDICES

SENSEX	MIDCAP	SMLCAP	BSE - 100	BSE - 200
BSE - 500	BSE IPO			

*BSE Sector Indices*

The 13 sector indices are

LISTING 7.2: BSE SECTOR INDICES

AUTO	BANKEX	CD	CG	FMCG
HC	IT	METAL	OIL&GA	POWER
PSU	REALTY	TEC		

*BSE Dollar Indices*

The three Dollar indices are

LISTING 7.3: BSE DOLLEX INDICES

DOLLEX - 30	DOLLEX - 100	DOLLEX - 200
-------------	--------------	--------------

## 7.5 BSE EQUITY LISTINGS

The components of the BSE indices can be obtained from the following source

<http://www.bseindia.com/downloads/abindices/file/Indices.zip>

You can download and unzip the file, it contains a MS .xls file with 11 work sheets for the following indices

LISTING 7.4: BSE EQUITY LISTING

SENSEX	BSE - 100	BSE - 200	BSE - 500
BSE - TECK	BSE - PSU	Sectoral	BANKEX
BSE - Mid Cap	BSE - Small Cap	BSE IPO	

*Example: SENSEX Index Listing*

This is a slightly modified listing from the zipped indices file which shows the 30 components of the SENSEX together with its Scrip Codes.

LISTING 7.5: BSE SENSEX INDEX LISTING

Code:	Description:	Sector:	Factor:
500410	ACC Ltd.	Housing Related	0.55
500103	Bharat Heavy Electricals Ltd.	Capital Goods	0.35
532454	Bharti Airtel Ltd.	Telecom	0.35
532868	DLF Ltd.	Housing Related	0.25
500300	Grasim Industries Ltd.	Diversified	0.75
500010	HDFC	Finance	0.90
500180	HDFC Bank Ltd.	Finance	0.80
500182	Hero Honda Motors Ltd.	Transport	0.50
500440	Hindalco Industries Ltd.	Metal & Mining	0.70
500696	Hindustan Unilever Ltd.	FMCG	0.50
532174	ICICI Bank Ltd.	Finance	1.00
500209	Infosys Technologies Ltd.	IT	0.85
500875	ITC Ltd.	FMCG	0.70
532532	Jaiprakash Associates Ltd.	Housing Related	0.55
500510	Larsen & Toubro Limited	Capital Goods	0.90
500520	Mahindra & Mahindra Ltd.	Transport	0.75
532500	Maruti Suzuki India Ltd.	Transport	0.50
532555	NTPC Ltd.	Power	0.20
500312	ONGC Ltd.	Oil & Gas	0.20
532712	Reliance Communications Ltd.	Telecom	0.35
500325	Reliance Industries Ltd.	Oil & Gas	0.50
500390	Reliance Infrastructure Ltd.	Power	0.65
500112	State Bank of India	Finance	0.45
500900	Sterlite Industries Ltd.	Metal & Mining	0.45
524715	Sun Pharma Industries Ltd.	Healthcare	0.40
532540	Tata Consultancy Services Ltd.	IT	0.30
500570	Tata Motors Ltd.	Transport	0.60
500400	Tata Power Company Ltd.	Power	0.70
500470	Tata Steel Ltd.	Metal & Mining	0.70
507685	Wipro Ltd.	IT	0.20

as. of March 29, 2010

You can use the Scrip Codes to download the historical time series using the function `bseDownload()`.

## 7.6 BSE SCRIP CODE AND SCRIP CODE SEARCH

*Indiamart* provides an Internet investment portal

<http://finance.indiamart.com>

which comes with a search engine for financial instruments traded at BSE. Thus we can write a search tool to find scrip codes and scrip name.

*Example: Search for "ABB"*

Let us search the instruments related to the search string "ABB", a company name. First create the URL for the search page

```
> SEARCH = "ABB"
> URL = composeURL("finance.indiamart.com/cgi-bin/stocksearch.cgi",
  "option=company", "&company_name=", SEARCH, "&which_exchange=bse")
> search = read.lynx(URL)
```

then crop the appropriate lines from the downloaded file

```
> ans = search[-(1:grep("Scrip ID", search))]
> ans = ans[(1:(grep("^$", ans)[1]) - 1)]
```

Then we clean the search file from undesired characters

```
> ans = gsub("^ ", "", ans)
> ans = gsub(" -", "", ans)
> ans = gsub(" \\[.\\.\\]View", "", ans)
> ans = gsub("\\[.\\.\\]", "", ans)
```

and remove obsolete lines

```
> Keywords = c("^PLAN-", "^DIVIDEND", "^OPTION", "^INVESTMENT")
> for (Keyword in Keywords) {
  index = grep(Keyword, ans)
  if (length(index) > 0)
    ans = ans[-grep(Keyword, ans)]
}
```

In the next step we convert the character vector into a three-column matrix

```
> ans = matrix(ans, byrow = TRUE, ncol = 3)
```

and abbreviate the description strings to 30 characters

```
«search6» ans[, 2] = substr(ans[, 2], 1, 30)
```

Finally we convert the search result into a data frame

```
> ans = as.data.frame(ans[, c(1, 3, 2)], stringsAsFactors = FALSE)
> colnames(ans) <- c("ScripCode", "SripName", "Description")
> ans
```

	ScripCode	SripName	Description
1	500002	ABB	ABB LTD.
2	500488	ABBOT	ABBOTT INDIA LTD.

The first column holds the scrip code, the second the scrip name, and the last a short description.

*BSE Search Function*

Use the above code snippets to write a search function

```

> bseSearch <- function(search) {
  URL = composeURL("finance.indiamart.com/cgi-bin/stocksearch.cgi",
    "?option=company", "&company_name=", search, "&which_exchange=bse")
  search = read lynx(URL)
  ans = search[-(1:grep("Scrip ID", search))]
  ans = ans[(1:(grep("^$", ans)[1]) - 1)]
  ans = gsub("^ ", "", ans)
  ans = gsub(" -", "", ans)
  ans = gsub("\\[\\.\\.\\]View", "", ans)
  ans = gsub("\\\\[\\.\\.\\]", "", ans)
  Keywords = c("^PLAN-", "^DIVIDEND", "^OPTION", "^INVESTMENT")
  for (Keyword in Keywords) {
    index = grep(Keyword, ans)
    if (length(index) > 0)
      ans = ans[-grep(Keyword, ans)]
  }
  ans = matrix(ans, byrow = TRUE, ncol = 3)
  ans[, 2] = substr(ans[, 2], 1, 30)
  ans = ans[, c(1, 3, 2), drop = FALSE]
  ans = as.data.frame(ans, stringsAsFactors = FALSE, drop = FALSE)
  colnames(ans) <- c("ScripCode", "SripName", "Description")
  ans
}

```

### Try it

```

> bseSearch("TATA")

```

	ScripCode	SripName	Description
1	500770	TATACHEM	TATA CHEMICALS LTD.
2	532301	TATACOFFEE	TATA COFFEE LTD.
3	500483	TATACOMM	TATA COMMUNICATIONS LTD.
4	532540	TCS	TATA CONSULTANCY SERVICES LTD.
5	500408	TATAELXSI	TATA ELXSI LTD.
6	533114	TFMP25AHG	TATA FIXED MATURITY PLAN SERIE
7	533115	TFMP25AHDP	TATA FIXED MATURITY PLAN SERIE
8	533116	TFMP25AHDR	TATA FIXED MATURITY PLAN SERIE
9	533111	TFMP25ARG	TATA FIXED MATURITY PLAN SERIE
10	533112	TFMP25ARDP	TATA FIXED MATURITY PLAN SERIE
11	533113	TFMP25ARDR	TATA FIXED MATURITY PLAN SERIE
12	533117	TFMP25ASG	TATA FIXED MATURITY PLAN SERIE
13	533118	TFMP25ASDP	TATA FIXED MATURITY PLAN SERIE
14	533119	TFMP25ASDR	TATA FIXED MATURITY PLAN SERIE
15	501301	TATAINVEST	TATA INVESTMENT CORPORATION LT
16	513434	TATAMETALI	TATA METALIKS LTD.
17	500570	TATAMOTORS	TATA MOTORS LTD.
18	533123	T25BRIPG	TATA MUTUAL FUND
19	533125	T25BRIPD	TATA MUTUAL FUND
20	533126	T25BHIPG	TATA MUTUAL FUND
21	533128	T25BHIPD	TATA MUTUAL FUND
22	533129	T25BSHIPG	TATA MUTUAL FUND
23	533182	TFMP26AG	TATA MUTUAL FUND
24	533183	TFMP26AQD	TATA MUTUAL FUND
25	533184	TFMP26APD	TATA MUTUAL FUND
26	533084	SM1AD	TATA MUTUAL FUND - TATA SMART
27	533085	SM1AG	TATA MUTUAL FUND - TATA SMART

28	533086	SM1BD TATA MUTUAL FUND - TATA SMART
29	533087	SM1BG TATA MUTUAL FUND - TATA SMART
30	500400	TATAPOWER TATA POWER CO.LTD.
31	513010	TATASPONGE TATA SPONGE IRON LTD.
32	500470	TATASTL TATA STEEL LTD.
33	500800	TATATEA TATA TEA LTD.
34	532371	TTML TATA TELESERVICES (MAHARASHTRA
35	570001	TATAMTDVR TATAMOTORS-DVR-A-ORDY
36	504961	TATAYODO TAYO ROLLS LTD.

Note, now we can use the scrip codes to download data from the BSE Internet portal.

## 7.7 BSE HOLIDAY CALENDAR

Missing data in the downloaded fields are due to the BSE holidays. Here is a holiday calendar for the year 2001

LISTING 7.6: BSE HOLIDAY CALENDAR 2010

	Holidays	Date	Day
1	New Year	1st January 2010	Friday
2	Republic Day	26th January 2010	Tuesday
3	Mahashivratri	12th February 2010	Friday
4	Holi (2nd Day)	1st March 2010	Monday
5	Ram Navmi	24th March 2010	Wednesday
6	Good Friday	2nd April 2010	Friday
7	Dr. B. Ambedkar Jayanti	14th April 2010	Wednesday
8	Ramzan Id	10th September 2010	Friday
9	Diwali A. (Laxmi Puja)	5th November 2010	Friday
10	Bakri-Id	17th November 2010	Wednesday
11	Moharum	17th December 2010	Friday

<http://www.bseindia.com/about/listholi.asp>

This allows us to write holiday function, exactly in the same way as those already available in the Rmetrics timeSeries package.

## CHAPTER 8

# DATA FROM FINANCE YAHOO

```
> library(fBasics)
> library(fImport)
```

### 8.1 FINANCE YAHOO INTERNET PORTAL

The Internet portal of Yahoo Finance

<http://finance.yahoo.com>

offers a huge number of historical time series for download. These cover a wide range of different financial market instruments, such as equities, interest rate instruments, funds and their indices. The historical data are provided in the form of CSV files. The datasets usually contain time series for the previous ten years, and some have even longer histories.

For the Indian market Finance Yahoo has its own Internet portal

<http://in.finance.yahoo.com>

### 8.2 DOWNLOADING CNX INDICES

CNX indices are listed on the following Yahoo web page of the Indian server

<http://in.finance.yahoo.com/indices?e=cnx>

For some of the indices a listing of equity symbols can be created. The following table shows the Yahoo symbols for the indices, and tells if component listings are available or not

LISTING 8.1: MSCI INDICES FROM ONVISTA'S INTERNET PORTAL

Symbol	Components	Name
^NSEI	NO	S&P CNX NIFTY
^NSED	NO	S&P CNX DEFTY
^NSMIDCP	YES	CNX NIFTY Junior
^CRSLDX	YES	S&P CNX 500
^CRSMID	YES	CNX MIDCAP 200
^NSEBANK	YES	BANK NIFTY
^CNXIT	YES	CNX IT

Use default settings to download the NIFTY index. In this case the URL takes a very simple string. The download results in

*Example: Download the NIFTY Index*

```
> NAME = "^NSEI"
> URL <- composeURL("ichart.finance.yahoo.com/table.csv?s=", NAME)
> NSEI.CSV = read.csv(URL)
> head(NSEI.CSV)
      Date  Open  High  Low  Close Volume Adj.Close
1 2010-05-28 5005.6 5077.2 5005.6 5066.6      0 5066.6
2 2010-05-27 4915.1 5016.6 4897.6 5003.1      0 5003.1
3 2010-05-26 4807.3 4925.4 4807.3 4917.4      0 4917.4
4 2010-05-25 4945.3 4946.6 4786.4 4806.8      0 4806.8
5 2010-05-24 4944.3 5029.6 4923.4 4943.9      0 4943.9
6 2010-05-21 4946.7 4946.7 4842.3 4931.1      0 4931.1
```

Now let us transfer the downloaded data into an object of class `timeSeries`

```
> NSEI <- rev(as.timeSeries(NSEI.CSV))
> colnames(NSEI) <- paste("NSEI.", c("O", "H", "L", "C", "V", "AC"),
  sep = ".")
> start(NSEI)
GMT
[1] [2002-08-12]
> tail(NSEI)
GMT
      NSEI.O NSEI.H NSEI.L NSEI.C NSEI.V NSEI.AC
2010-05-21 4946.7 4946.7 4842.3 4931.1      0 4931.1
2010-05-24 4944.3 5029.6 4923.4 4943.9      0 4943.9
2010-05-25 4945.3 4946.6 4786.4 4806.8      0 4806.8
2010-05-26 4807.3 4925.4 4807.3 4917.4      0 4917.4
2010-05-27 4915.1 5016.6 4897.6 5003.1      0 5003.1
2010-05-28 5005.6 5077.2 5005.6 5066.6      0 5066.6
```

Plot the adjusted Closing price and the volume

```
> plot(NSEI[, 6:5], main = "CNX Nifty")
```

The Volume data has many zero entries, and evenmore they are not provided after beginning of 2007.

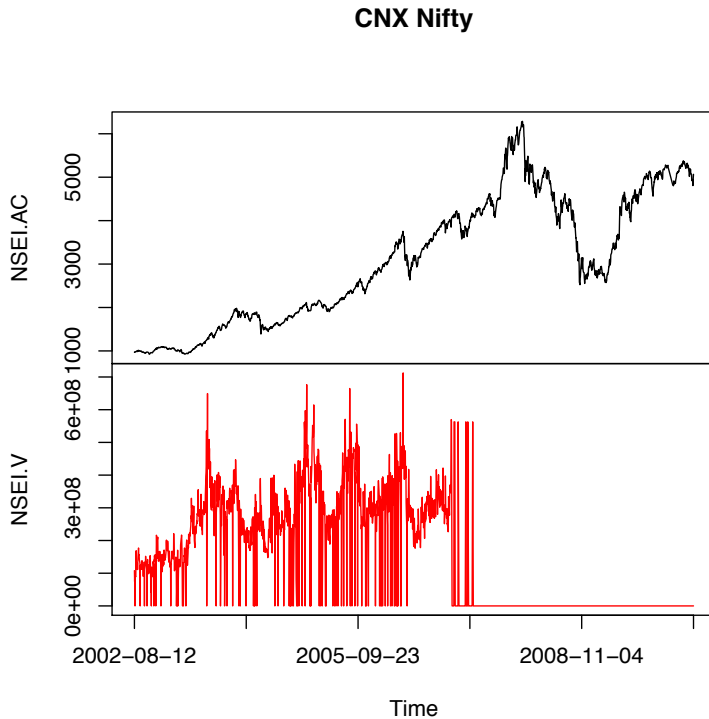


FIGURE 8.1: Adjusted Closing Price and Volume of CNX Nifty

### 8.3 THE FUNCTION yahooSeries()

To download the time series for the indices we can use the function `yahooSeries()` from the Rmetrics package `fImport`

```
> args(yahooSeries)
function (symbols, from = NULL, to = Sys.timeDate(), nDaysBack = 366,
  ...)
NULL
```

*Example: Downloading CNX Indices from Yahoo*

As an example let us download the NIFTY index from Finance Yahoo. Try it for the last 10 years

```
> NSEI = yahooSeries("^NSEI", from = Sys.Date() - 10 * 366)
> head(NSEI)
```



```
GMT
      ^NSEI.Open ^NSEI.High ^NSEI.Low ^NSEI.Close ^NSEI.Volume
2010-05-28      5005.6      5077.2      5005.6      5066.6           0
2010-05-27      4915.1      5016.6      4897.6      5003.1           0
2010-05-26      4807.3      4925.4      4807.3      4917.4           0
2010-05-25      4945.3      4946.6      4786.4      4806.8           0
2010-05-24      4944.3      5029.6      4923.4      4943.9           0
2010-05-21      4946.7      4946.7      4842.3      4931.1           0
      ^NSEI.Adj.Close
2010-05-28          5066.6
2010-05-27          5003.1
2010-05-26          4917.4
2010-05-25          4806.8
2010-05-24          4943.9
2010-05-21          4931.1
```

```
> tail(NSEI)
```

```
GMT
      ^NSEI.Open ^NSEI.High ^NSEI.Low ^NSEI.Close ^NSEI.Volume
2002-08-19      979.40      983.55      978.20      979.85      89119400
2002-08-16      970.75      981.20      968.35      979.25      92417900
2002-08-15      969.65      969.65      969.65      969.65           0
2002-08-14      976.05      976.25      967.25      969.65     103412400
2002-08-13      970.40      977.20      969.90      976.05      74423000
2002-08-12      961.15      971.55      961.15      969.85     108092200
      ^NSEI.Adj.Close
2002-08-19          979.85
2002-08-16          979.25
2002-08-15          969.65
2002-08-14          969.65
2002-08-13          976.05
2002-08-12          969.85
```

### Example: Creating a CNX BANK Listing with Yahoo Symbols

First let us download the index components

```
> NAME = "^NSEBANK"
> URL <- composeURL("in.finance.yahoo.com/d/quotes.csv?s=@", NAME,
  "&f=slldtltlclohgv&e=.csv&h=1")
> Components <- read.csv(URL, stringsAsFactors = FALSE)[, 1]
> Components
 [1] "BANKINDIA.NS" "CANBK.NS"      "CORPBANK.NS"   "HDFCBANK.NS"   "ICICIBANK.NS"
 [6] "KOTAKBANK.NS" "ORIENTBAN.NS" "PNB.NS"        "SBIN.NS"       "UNIONBANK.NS"
[11] "UTIBANK.NS"
```

and then the series for "BANKINDIA.NS"

```
> BANKINDIA <- yahooSeries("BANKINDIA.NS")
> BANKINDIA <- yahooSeries("BANKINDIA.NS")
> colnames(BANKINDIA) <- abbreviate(colnames(BANKINDIA))
> start(BANKINDIA)
GMT
[1] [2009-05-29]
```

```
> tail(BANKINDIA)
```

```
GMT
```

```

      BANKINDIA.NS.O BANKINDIA.NS.H BANKINDIA.NS.L BANKINDIA.NS.C
2009-06-05          343.9          354.9          342.15          344.15
2009-06-04          336.0          351.9          331.20          344.05
2009-06-03          335.0          341.8          325.20          339.00
2009-06-02          344.9          350.0          320.15          332.00
2009-06-01          340.0          350.0          340.00          348.95
2009-05-29          332.0          341.0          326.00          338.50
      BANKINDIA.NS.V BANKINDIA.NS.A
2009-06-05        1524400          211.78
2009-06-04        1861300          211.72
2009-06-03        3260900          208.62
2009-06-02        3108100          204.31
2009-06-01        2140900          214.74
2009-05-29        2085200          208.31

```

The function `abbreviate()` abbreviates the column names.

#### 8.4 DOWNLOADING BSE INDICES

BSE indices from the Bombay Stock Exchange are listed on the following Yahoo web page of the Indian server

```
http://in.finance.yahoo.com/indices?e=cnx
```

For some of the indices a listing of equity symbols can be created. The following table shows the Yahoo symbols for the indices, and tells if component listings are available or not

LISTING 8.2: MSCI INDICES FROM ONVISTA'S INTERNET PORTAL

Symbol	Components	Historical	Name
^BSESN			BSE 30 SENSEX
BSE-AUTO.B0			AUTO INDEX
BSE-BANK.B0			BANKEX
BSE-100.B0			BSE 100
BSE-200.B0			BSE 200
BSE-500.B0			BSE 500
BSE-CG.B0			BSE CAPITAL GOODS
BSE-CD.B0			BSE CONS. DURABLES
BSE-FCMG.B0			BSE FMCG
BSE-HC.B0			BSE HEALTHCARE
BSE-IT.B0			BSE INFOTECH
BSE-MIDCAP.B0			BSE Mid-Cap
BSE-PSU.B0			BSE PUBLIC SECTOR
BSE-SMLCAP.B0			BSE Small-Cap
BSE-TECK.B0			BSE TECK INDEX
BSE-METAL.B0			METAL INDEX
BSE-OILGAS.B0			OIL & GAS INDEX

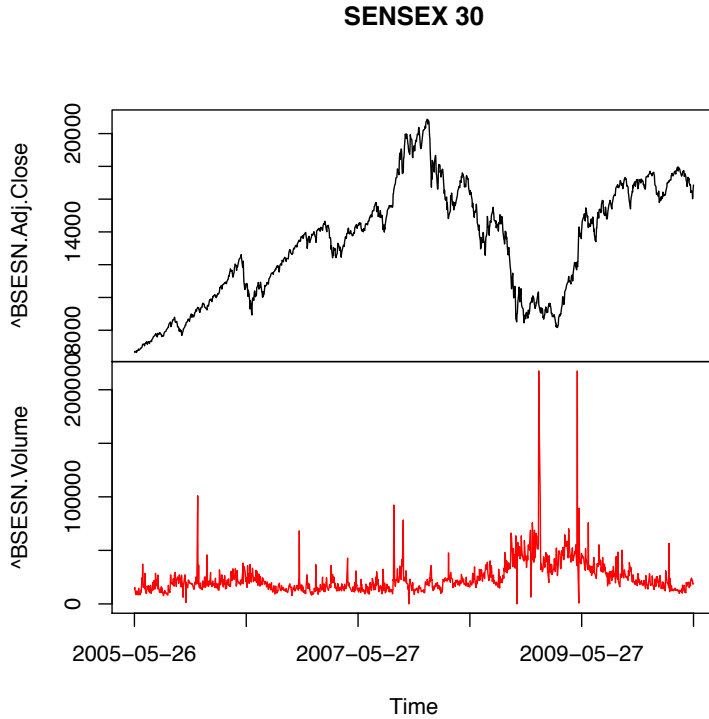


FIGURE 8.2: Adjusted Close and Volume for the Sensex Index

*Example: Downloading BSE30 Index from Yahoo*

To download a time series for the BSE indices we can again use the function `yahooSeries()` from the Rmetrics package `fImport`. Note, only for the BSE 30 SENSEX historical data records are available. Download the data for the last 5 years

```
> BSE30 <- yahooSeries("^BSESN", nDaysBack = 5 * 366)
> colnames(BSE30)
[1] "^BSESN.Open"      "^BSESN.High"      "^BSESN.Low"       "^BSESN.Close"
[5] "^BSESN.Volume"    "^BSESN.Adj.Close"
```

and plot the adjusted Closing Price and the Volume

```
> plot(BSE30[, 6:5], main = "SENSEX 30")
```

Note, there are no component listings available for BSE indices on Yahoo, but we can use those from the Bombay Stock Exchange.

*Example: Downloading NTPC Prices from Yahoo*

The symbols for the BSE listed companies are made from the BSE Scrip Name and the suffix ".BO", eg. "NTPC.BO". As an example we download the maximum available set of data from Yahoo

```
> NTPC = yahooSeries("NTPC.BO", from = Sys.Date() - 10 * 366)
> start(NTPC)
GMT
[1] [2004-11-05]
> tail(NTPC)
GMT
      NTPC.BO.Open NTPC.BO.High NTPC.BO.Low NTPC.BO.Close NTPC.BO.Volume
2004-11-12      74.90      75.25      74.50      74.60      1652800
2004-11-11      75.70      75.70      74.35      74.45      4880600
2004-11-10      75.55      75.90      75.00      75.05      5615800
2004-11-09      75.90      76.05      74.55      75.20      15346200
2004-11-08      76.50      78.45      75.00      75.20      24002100
2004-11-05      70.00      76.15      68.00      75.55      103332400
      NTPC.BO.Adj.Close
2004-11-12      66.14
2004-11-11      66.01
2004-11-10      66.54
2004-11-09      66.67
2004-11-08      66.67
2004-11-05      66.98
```

and plot the time series

```
> plot(NTPC, main = "NTPC")
```

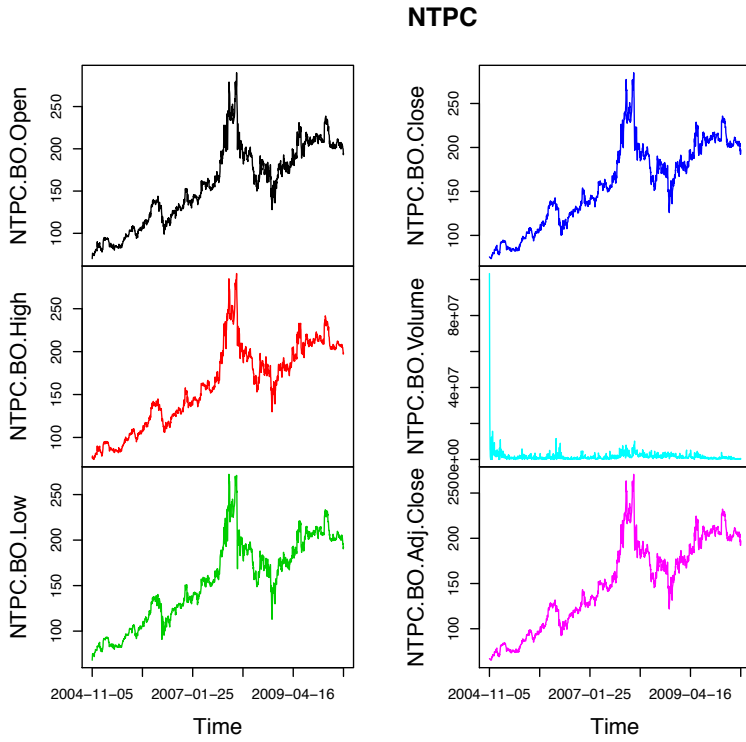


FIGURE 8.3: NTPC Data from Finance Yahoo

## CHAPTER 9

# DATA FROM ONVISTA

```
> library(fBasics)
> library(fImport)
> library(fAssets)
```

### 9.1 ONVISTA INTERNET PORTAL

The Internet Portal of OnVista,

<http://www.onvista.de>

offers like Yahoo Finance a huge amount on historical time series for several kinds of financial instruments. OnVista is a German portal. The offered time series are going back to a maximum period of 10 years. The categories or asset classes of time series available for download from the OnVista portal include

LISTING 9.1: CATEGORY LINKS ON THE ONVISTA SERVER

---

Link: Aktien	Equities
Link: Indizes	Indices
Link: Fonds	Funds
Link: ETFs	ETFs
Link: Anleihen	Bonds
Link: Devisen	Currencies
Link: Rohstoffe	Commodities

---

Each category of assets has its own server. For the indices the server address is

<http://index.onvista.de>

On this page one access further links, e.g. the MSCI indices amongst others.

## 9.2 MSCI INDIA INDEX

MSCI Country Indices can be downloaded from the German Internet Portal Onvista

<http://index.onvista.de/msci-indizes.html>

### *Example: Download MSCI India*

To Download the MSCI India Index from OnVista one has to know the internal NOTATION identification number. In addition one can specify the download range in units of months. Let us download the series for the last 5 years.

```
> NOTATION = "3194472"
> RANGE = "60M"
> URL = composeURL("index.onvista.de/quote_history.html", "?ID_NOTATION=",
  NOTATION, "&RANGE=", RANGE)
> Download = read.lynx(URL)
> head(Download, 15)
[1] ""
[2] " Onvista Stand: 25. 05. 2010"
[3] ""
[4] " Einzelkursabfrage"
[5] " MSCI INDIA (STRD, UHD)"
[6] ""
[7] " Typ: Aktienindex Familie: Morgan Stanley Capital Intern. ISIN:"
[8] " WKN/ID: Symbol: Land: Indien"
[9] ""
[10] " Einzelkursabfrage"
[11] " Datum Eröffnung Tief Hoch Schluss"
[12] " 24.05.10 668,998 668,998 668,998 668,998"
[13] " 21.05.10 666,981 666,981 666,981 666,981"
[14] " 20.05.10 670,749 670,749 670,749 670,749"
[15] " 19.05.10 667,704 667,704 667,704 667,704"
```

### Clean the file

```
> download = indexGrep("^ [0-9][0-9]\\.[0-9][0-9]", Download)
> download = gsub("\\.", "", download)
> download = gsub(", ", ".", download)
> download = gsub(" ", " ", download)
> download = gsub(" ", " ", download)
> download = gsub("^ ", "", download)
```

### Compose to time series object

```

> data <- dataSplit(download)[, col = 4, drop = FALSE]
> charvec <- charvecSplit(download, format = "%d%m%y")
> MSCI <- timeSeries(data, charvec, units = "MSCI India")
> start(MSCI)
GMT
[1] [2005-05-23]
> tail(MSCI)
GMT
          MSCI India
2005-05-31    275.70
2005-05-30    273.95
2005-05-27    275.39
2005-05-26    274.32
2005-05-25    271.16
2005-05-23    269.96

```

Plot the log of the index and the log returns

```

> MSCI2 <- cbind(log(MSCI), 100 * returns(MSCI))
> plot(MSCI2, main = "MSCI India")

```

### 9.3 THE FUNCTION `msciOnvistaDownload()`

In this section we show how to write a function to download the MSCI indices from Onvista. You have to know the NOTATION identifiers

*Example: Write a Download Function*

```

> msciOnvistaDownload <- function(notation, range = "1M") {
  URL = composeURL("index.onvista.de/quote_history.html", "?ID_NOTATION=",
    notation, "&RANGE=", range)
  Download = read.lynx(URL)
  download = indexGrep("^ [0-9][0-9]\\.[0-9][0-9]", Download)
  download = gsub("\\.", "", download)
  download = gsub(",", ".", download)
  download = gsub(" ", " ", download)
  download = gsub(" ", " ", download)
  download = gsub(" ", " ", download)
  data <- dataSplit(download)[, col = 4, drop = FALSE]
  charvec <- charvecSplit(download, format = "%d%m%y")
  MSCI <- timeSeries(data, charvec, units = "MSCI Index")
  return(MSCI)
}

```

*Example: Download MSCI India, Emerging Markets and World Indices*

Use the `msciOnvistaDownload()` function and download the MSCI country index for India, the MSCI Emerging Market index, and the MSCI World index.



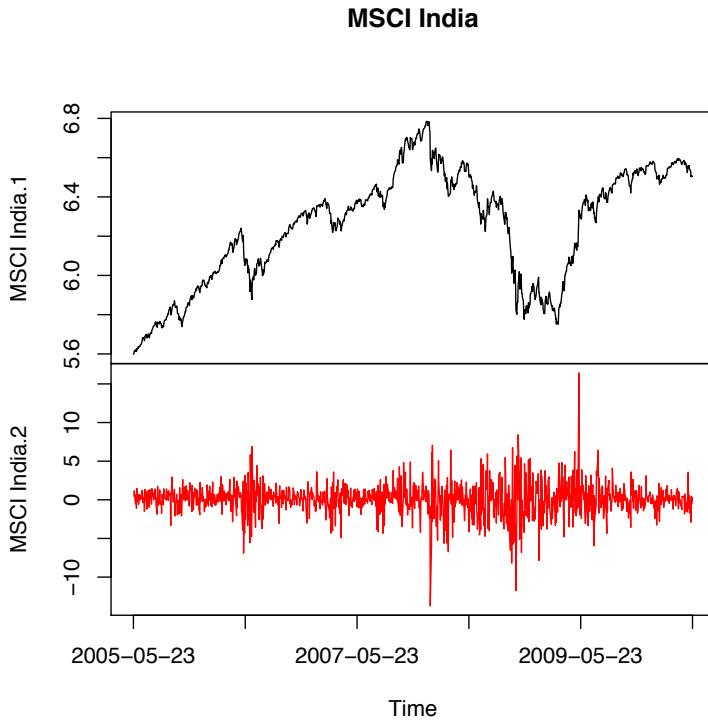


FIGURE 9.1: Log-Index and Log&gt;Returns of MSCI India Index

```
> MSCI.INDIA = msci0nvistaDownload(notation = "3194472", range = "120M")
> MSCI.EMMA = msci0nvistaDownload(notation = "1643097", range = "120M")
> MSCI.WORLD = msci0nvistaDownload(notation = "1157217", range = "120M")
```

Bind the three series together in a common `timeSeries` object

```
> MSCI = cbind(INDIA = log(MSCI.INDIA), EMMA = log(MSCI.EMMA),
              WORLD = log(MSCI.WORLD))
> start(MSCI)
GMT
[1] [2000-05-17]
> tail(MSCI)
GMT
          INDIA  EMMA  WORLD
2010-05-17 6.5345 6.8456 6.9859
2010-05-18 6.5329 6.8456 6.9716
2010-05-19 6.5038 6.8140 6.9662
2010-05-20 6.5084 6.7827 6.9264
2010-05-21 6.5028 6.7875 6.9413
```

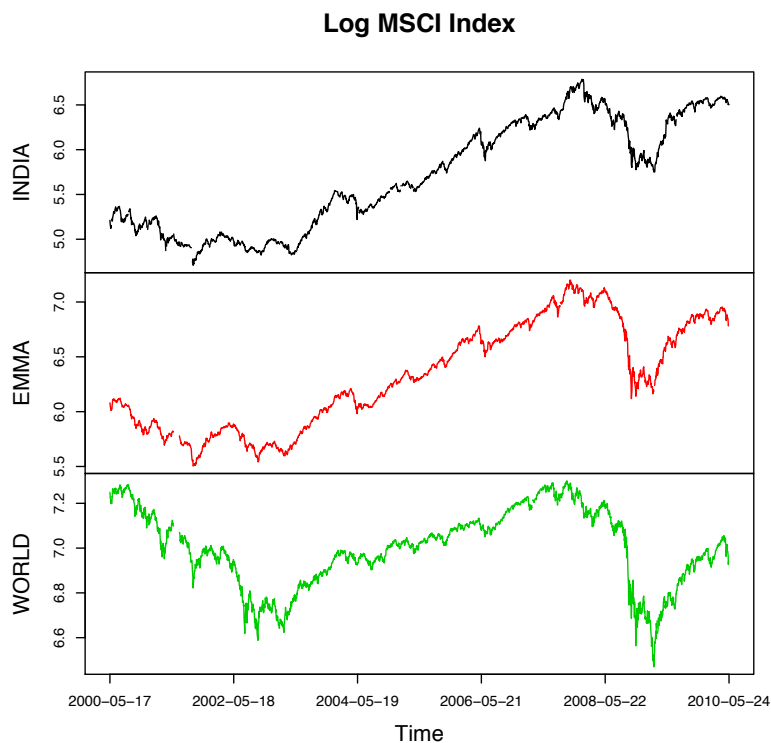


FIGURE 9.2: Log MSCI Index

```
2010-05-24 6.5058    NA    NA
```

and plot them

```
> plot(MSCI, main = "Log MSCI Index")
```

Have a look on the correlations between the three markets

```
> pairs(returns(MSCI), pch = 19, col = "steelblue")
```

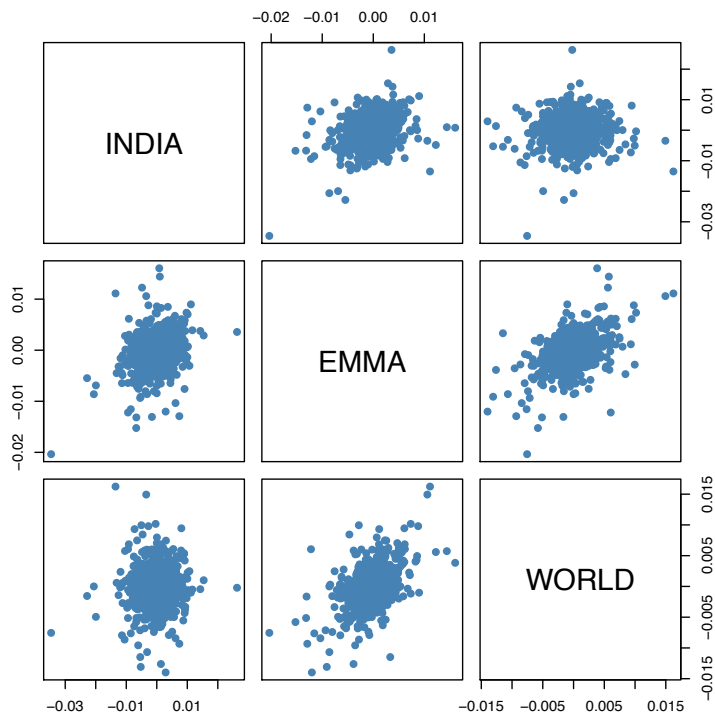


FIGURE 9.3: Correlation Plot of MSCI Returns

## CHAPTER 10

# DATA FROM OANDA

```
> library(fImport)
```

### 10.1 OANDA INTERNET PORTAL

The Foreign Exchange Internet portal of Oanda

```
http://www.oanda.com
```

is a platform for FX trading. The portal allows the access to historical prices of currencies and thus to download exchange rates for the Indian Rupee.

To download exchange rates from Oanda's trading platform we have only to know the ISO codes for the desired countries. On the Oanda platform one can find a page which lists the ISO codes and a page for searching the standard 3-letter currency codes for any country. The links are

```
http://www.oanda.com/help/currency-iso-code-country  
http://www.oanda.com/help/currency-iso-code  
http://www.oanda.com/currency/currency-code
```

### 10.2 THE DOWNLOAD URL

The link to download data from the web site is composed from the home currency CCY1, from the foreign currency CCY2, and from the start FROM and end T0 date. The following example shows how to compose an URL.

*Example: Compose the URL for the EUR/USD Currency Pair*

Let us start with the settings for currency pair and the download period

```
> CCY1 <- "INR"
> CCY2 <- "EUR"
> FROM <- Sys.Date() - 366
> TO <- Sys.Date()
```

Now we can compose the URL

```
> fromDate <- format(FROM, "%m-%d-%y")
> toDate <- format(TO, "%m-%d-%y")
> URL <- composeURL("www.oanda.com/convert/fxhistory?lang=en",
  fromDate, toDate, "&date_fmt=us", "&exch=", CCY1, "&expr2=",
  CCY2, "&margin_fixed=0&SUBMIT=Get+Table&format=CSV&redirected=1")
```

Since the URL is too long to get nicely printed, we have the character string splitted in parts

```
> strsplit(gsub("&", " &", URL, perl = TRUE), " ")[[1]]
[1] "http://www.oanda.com/convert/fxhistory?lang=en"
[2] "&date1=05%2F27%2F09"
[3] "&date=05%2F28%2F10"
[4] "&date_fmt=us"
[5] "&exch=INR"
[6] "&expr2=EUR"
[7] "&margin_fixed=0"
[8] "&SUBMIT=Get+Table"
[9] "&format=CSV"
[10] "&redirected=1"
```

*Example: Download the USDEUR Rates*

Then take the URL, download the time series using the Lynx reader, and extract the the records which contain a date format "mm/dd/yyyy"

```
> Download <- read.Lynx(URL)
> Download <- indexGrep("^[:space:]]*.../.../...", Download, perl = TRUE)
```

With the second R command we have extracted the lines from the file which start with a calendar date. In the next step we show how to extract the date and data records form the downloaded file

```
> INREUR <- timeSeries(data = dataSplit(Download, split = ","),
  charvec = charvecSplit(Download, split = ",", format = "%m/%d/%Y"),
  units = paste(CCY1, CCY2, sep = "/"))
```

and remove records with missing values

```
> INREUR <- na.omit(INREUR)
```

Let us print the starting date and the most recent rates

```

> start(INREUR)
GMT
[1] [2009-05-27]

> tail(INREUR)
GMT
      INR/EUR
2010-05-23 0.01698
2010-05-24 0.01700
2010-05-25 0.01720
2010-05-26 0.01716
2010-05-27 0.01725
2010-05-28 0.01737

```

### 10.3 THE FUNCTION `oandaDownload()`

The code snippets above can be used to write a download function

*Example: Write a Download Function*

```

> oandaDownload <- function(ccy1, ccy2, from = Sys.Date() - 366,
  to = Sys.Date()) {
  fromDate <- format(from, "%m%%2F%d%%2F%y")
  toDate <- format(to, "%m%%2F%d%%2F%y")
  URL <- composeURL("www.oanda.com/convert/fxhistory?lang=en",
    fromDate, toDate, "%date_fmt=us", "%exch=", ccy1, "%expr2=",
    ccy2, "%margin_fixed=0&SUBMIT=Get+Table&format=CSV&redirected=1")
  download <- read lynx(URL)
  download <- indexGrep("^[:space:]]*.../.../...", download,
    perl = TRUE)
  data <- dataSplit(download, split = ",")
  charvec <- charvecSplit(download, split = ",", format = "%m/%d/%Y")
  units <- paste(ccy1, ccy2, sep = "")
  tS <- na.omit(timeSeries(data, charvec, units))
  tS
}

```

*Exercise: Download of Major INR Rates*

As an example we create a multivariate time series of the major currencies USD, EUR, GBP, and JPY, against the INR for the last half year (183 days)

```

> CCY <- c("USD", "EUR", "GBP", "JPY")
> FROM <- Sys.Date() - 183
> RATES <- oandaDownload("INR", CCY[1], FROM)
> for (i in 2:4) RATES <- cbind(RATES, oandaDownload("INR", CCY[i],
  FROM))

```

Print the starting date and have a look on the most recent FX rates

```

> start(RATES)

```

```
GMT
[1] [2009-11-26]
> tail(RATES)
GMT
      INRUSD  INREUR  INRGBP  INRJPY
2010-05-23 0.02135 0.01698 0.01475 1.9226
2010-05-24 0.02137 0.01700 0.01477 1.9257
2010-05-25 0.02137 0.01720 0.01480 1.9298
2010-05-26 0.02104 0.01716 0.01468 1.8893
2010-05-27 0.02115 0.01725 0.01469 1.9087
2010-05-28 0.02136 0.01737 0.01472 1.9345
```

1

Plot the rates

```
> plot(log(RATES), main = "Log INR FX Rates")
```

and compute the correlation matrix of the log returns

```
> cor(log(RATES))
      INRUSD  INREUR  INRGBP  INRJPY
INRUSD 1.00000 0.60583 0.66646 0.91174
INREUR 0.60583 1.00000 0.95489 0.65586
INRGBP 0.66646 0.95489 1.00000 0.66632
INRJPY 0.91174 0.65586 0.66632 1.00000
```

---

<sup>1</sup>Note, the download of rates from the Oanda portal is limited to 2000 records. If one likes to download more records, one has to split the download and merge the series.

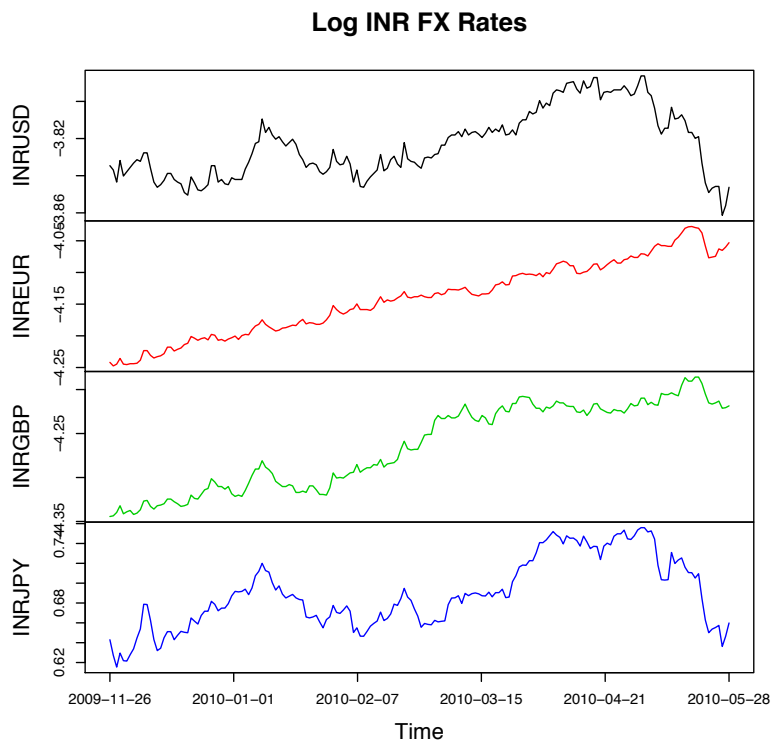


FIGURE 10.1: Log of INR FX Rates



PART IV

APPENDIX

## APPENDIX A

# PACKAGES REQUIRED FOR THIS EBOOK

```
> library(fBasics)
> library(fImport)
```

In the following we briefly describe the packages required for this ebook.

### RMETRICS PACKAGE: `timeDate`

`timeDate` contains R functions to handle time, date and calendar aspects. The S4 `timeDate` class is used in Rmetrics for financial data and time management together with the management of public and ecclesiastical holidays. The class fulfils the conventions of the ISO 8601 standard as well as of the ANSI C and POSIX standards. Beyond these standards, Rmetrics has added the ‘Financial Center’ concept, which allows you to handle data records collected in different time zones and combine them with the proper time stamps of your personal financial centre, or, alternatively, to the GMT reference time. The S4 class can also handle time stamps from historical data records from the same time zone, even if the financial centres changed daylight saving times at different calendar dates. Moreover, `timeDate` is almost compatible with Insightful’s SPlus `timeDate` class. If you move between the two worlds of R and SPlus, you will not have to rewrite your code. This is important for many business applications. The class offers not only date and time functionality, but also sophisticated calendar manipulations for business days, weekends, public and ecclesiastical holidays. `timeSeries` can be downloaded from the CRAN server. Development versions are also available from the R-Forge repository.

```
> listDescription(timeDate)
timeDate Description:
```

```

Package:      timeDate
Version:     2110.88
Revision:
Date:        2009-12-10
Title:       Rmetrics - Chronological and Calendarical Objects
Author:      Diethelm Wuertz and Yohan Chalabi with contributions
             from Martin Maechler, Joe W. Byers, and others
Depends:     R (>= 2.6.0), graphics, utils, stats, methods
Suggests:    RUnit
Maintainer:  Rmetrics Core Team <Rmetrics-core@r-project.org>
Description: Environment for teaching "Financial Engineering and
             Computational Finance"
NOTE:        SEVERAL PARTS ARE STILL PRELIMINARY AND MAY BE CHANGED
             IN THE FUTURE. THIS TYPICALLY INCLUDES FUNCTION AND
             ARGUMENT NAMES, AS WELL AS DEFAULTS FOR ARGUMENTS AND
             RETURN VALUES.
LazyLoad:    yes
LazyData:    yes
License:     GPL (>= 2)
URL:         http://www.rmetrics.org
Built:       R 2.9.1; ; 2009-12-28 09:59:44 UTC; windows

```

#### RMETRICS PACKAGE: timeSeries

`timeSeries` is the Rmetrics package that allows us to work very efficiently with S4 `timeSeries` objects. Let us briefly summarize the major functions available in this package. You can create `timeSeries` objects in several different ways, i.e. you can create them from scratch or you can read them from a file. you can print and plot these objects, and modify them in many different ways. Rmetrics provides functions that compute financial returns from price/index series or the cumulated series from returns. Further modifications deal with drawdowns, durations, spreads, mid-quotes and may other special series. `timeSeries` objects can be subset in several ways. You can extract time windows, or you can extract start and end data records, and you can aggregate the records on different time scale resolutions. Time series can be ordered and resampled, and can be grouped according to statistical approaches. You can apply dozens of math operations on time series. `timeSeries` can also handle missing values.

```

> listDescription(timeSeries)
timeSeries Description:

Package:      timeSeries
Version:     2110.88
Revision:
Date:        2010-01-06
Title:       Rmetrics - Financial Time Series Objects

```

```

Author:      Diethelm Wuertz and Yohan Chalabi
Depends:     R (>= 2.6.0), graphics, grDevices, methods, stats,
             utils, timeDate (>= 2100.86)
Suggests:    robustbase, RUnit
Maintainer:  Rmetrics Core Team <Rmetrics-core@r-project.org>
Description: Environment for teaching "Financial Engineering and
             Computational Finance"
NOTE:        SEVERAL PARTS ARE STILL PRELIMINARY AND MAY BE CHANGED
             IN THE FUTURE. THIS TYPICALLY INCLUDES FUNCTION AND
             ARGUMENT NAMES, AS WELL AS DEFAULTS FOR ARGUMENTS AND
             RETURN VALUES.
LazyLoad:    yes
LazyData:    yes
License:     GPL (>= 2)
URL:         http://www.rmetrics.org
Built:       R 2.9.1; ; 2010-05-03 15:15:55 UTC; windows

```

### RMETRICS PACKAGE: fBasics

fBasics provides basic functions to analyze and to model data sets of financial time series. The topics from this package include distribution functions for the generalized hyperbolic distribution, the stable distribution, and the generalized lambda distribution. Beside the functions to compute density, probabilities, and quantiles, you can find there also random number generators, functions to compute moments and to fit the distributional parameters. Matrix functions, functions for hypothesis testing, general utility functions and plotting functions are further important topics of the package.

```
> listDescription(fBasics)
```

```
fBasics Description:
```

```

Package:      fBasics
Version:      2110.80
Revision:     4727
Date:         2010-02-08
Title:        Rmetrics - Markets and Basic Statistics
Author:       Diethelm Wuertz and Rmetrics core team members, uses
             code builtin from the following R contributed packages:
             gmm from Pierre Chauss, gld from Robert King, gss from
             Chong Gu, nortest from Juergen Gross, HyperbolicDist
             from David Scott, sandwich from Thomas Lumley and Achim
             Zeileis, and fortran/C code from Kersti Aas.
Depends:      R (>= 2.6.0), MASS, methods, timeDate, timeSeries (>=
             2100.84)
Suggests:     akima, spatial, RUnit, tcltk
Maintainer:   Rmetrics Core Team <Rmetrics-core@r-project.org>
Description:  Environment for teaching "Financial Engineering and
             Computational Finance" NOTE: SEVERAL PARTS ARE STILL
             PRELIMINARY AND MAY BE CHANGED IN THE FUTURE. THIS
             TYPICALLY INCLUDES FUNCTION AND ARGUMENT NAMES, AS WELL

```

```

AS DEFAULTS FOR ARGUMENTS AND RETURN VALUES. Please
donate, www.rmetrics.org, to support future activities
of the Rmetrics association.
LazyLoad:      yes
LazyData:      yes
License:        GPL (>= 2)
URL:           http://www.rmetrics.org
Built:         R 2.9.1; i386-pc-mingw32; 2010-05-28 17:49:30 UTC;
               windows

```

## RMETRICS PACKAGE: fImport

fImport provides basic functions to download and import time series from *Yahoo Finance*, the U.S. *Federal Reserve* and from the *Oanda* Foreign Exchange trading platform. The topics include download functions, and reader functions. Amongst the readers we have added R functions to make the functionality of *Lynx*, *Links*, *w3m*, and *wget* available. Furthermore the package provides some useful utilities for grepping and splitting data text files, and to manipulate data from *xls* files.

```

> listDescription(fImport)
fImport Description:

Package:        fImport
Version:        2120.80
Revision:       4826
Date:           2010-04-14
Title:          Rmetrics - Economic and Financial Data Import
Author:         Diethelm Wuertz and many others, see the SOURCE file
Depends:        R (>= 2.6.0), methods, timeDate, timeSeries
Suggests:       RUnit
Maintainer:     Rmetrics Core Team <Rmetrics-core@r-project.org>
Description:    Environment for teaching "Financial Engineering and
                Computational Finance"
NOTE:           SEVERAL PARTS ARE STILL PRELIMINARY AND MAY BE CHANGED
                IN THE FUTURE. THIS TYPICALLY INCLUDES FUNCTION AND
                ARGUMENT NAMES, AS WELL AS DEFAULTS FOR ARGUMENTS AND
                RETURN VALUES.
LazyLoad:       yes
LazyData:       yes
License:        GPL (>= 2)
URL:           http://www.rmetrics.org
Built:          R 2.9.1; ; 2010-05-17 21:28:22 UTC; windows

```

## APPENDIX B

# NSE FUNCTION SUMMARY

We have shown how to write functions to download time series data from NSE's Internet portal and how to write functions to generate equity listings from indices. These functions are summarized in the following table.

LISTING B.1: FUNCTIONS TO DOWNLOAD DATA FROM NSE

---

Functions:	Description
nseEquityDownload	downloads an equity time series from NSE
nseIndexDownload	downloads an index time series from NSE
nseDebtDownload	downloads a debt time series from NSE
nseListing	creates a listing for NSE market indices
bseDownload	downloads a time series from BSE
yahooSeries	downloads a time series from Finance Yahoo
msciOnvistaDownload	downloads MSCI time series from OnVista
oandaDownload	downloads a FX time series from Oanda

---

Typical arguments for these functions are

LISTING B.2: ARGUMENTS FOR DOWNLOAD AND LISTING FUNCTIONS

---

Arguments:	Description:
name, market	an integer value, the market identifier
symbol	a character string, a user defined symbol name
from, to	start and end dates

---

## APPENDIX C

# LYNX TEXT READER

Lynx is a text-only Web browser for use on character terminals. It is released under the GNU General Public License. Supported protocols include HTTP, HTTPS, and FTP amongst others. For an overview of the Lynx Web browser we refer to the Wikipedia Web page

[http://en.wikipedia.org/wiki/Lynx\\_\(web\\_browser\)](http://en.wikipedia.org/wiki/Lynx_(web_browser))

Lynx was originally designed for Unix. Versions are also available for all Microsoft Windows releases, Linux and Mac OS X. You can access the home page of the browser via the link

<http://lynx.isc.org>

There you can find the current development sources, the main help page, the current User Guide. The main help page and the User Guide are online available.

### WINDOWS INSTALLATION

#### *cygwin Win32 Installation*

Our preferred way to use the text browser lynx is to call the function `lynx.exe` from a cygwin installation under Windows. To learn how to install cygwin under Windows we refer to the following links

<http://www.cygwin.com>

<http://en.wikipedia.org/wiki/Cygwin>

cygwin is a Unix-like environment and command-line interface for Microsoft Windows. This environment allows to launch Windows applications from the cygwin environment, as well as to use many cygwin tools and applications within the Windows operating context, this includes the lynx Web browser.

When you have installed `cygwin`, please do not forget to add the location of the `lynx.exe` binary to the search path of your windows environment. Our experience is that `cygwin` is the most stable way under Windows to download data from the Internet<sup>1</sup>.

### *The Minimalist Win32 Version*

A standalone installation will be an alternative option if you like to avoid the time consuming and maybe for you difficult `cygwin` installation. You can find many pointers on the Internet to standalone Windows binaries of `lynx` but none was working properly together with R. We have prepared a zip file with the necessary `lynx` files from the `cygwin` distribution. You can download this file named `lynx4RmetricsWindows.zip` from the `r-forge` server

<https://r-forge.r-project.org/scm/viewvc.php/share/lynx4RmetricsWindows.zip?root=rmetrics>

Click on the (download) link to get the latest revision. Then unzip the downloaded file and copy the `cygwin` folder to your "Computer". i.e. place it under

`C:\cygwin`

Please do not forget to add the location of the `lynx.exe` binary, i.e.

`C:\cygwin\bin`

to the search path of your windows environment. That's all.

### LINUX INSTALLATION

We assume that the user is familiar with his Linux operating system. `lynx` is in almost all cases already installed and the binary is in the search path. So the `read.lynx()` will work out of the box.

### MAC OS X INSTALLATION

For the Mac OS X operating system the `lynx` web browser is available from

[http://www.apple.com/downloads/macosx/unix\\_open\\_source/lynxtextwebbrowser.html](http://www.apple.com/downloads/macosx/unix_open_source/lynxtextwebbrowser.html)

Alternatively, you can install `lynx` using the Macports package management system:

<http://www.macports.org>

---

<sup>1</sup>Note with a full `cygwin` installation you also get access to other web browsers and downloaders like `wget`, `w3m`, or `links`.



## APPENDIX D

# RMETRICS TERMS OF LEGAL USE

### *Grant of License*

*Rmetrics Association* and *Finance Online Zurich* have authorized you to download one copy of this electronic book (ebook). The service includes free updates for the period of one year. *Rmetrics Association* and *Finance Online Zurich* grant you a nonexclusive, nontransferable license to use this eBook according to the terms and conditions specified in the following. This license agreement permits you to install and use the eBook for your personal use only.

### *Restrictions*

You shall not resell, rent, assign, timeshare, distribute, or transfer all or any part of this eBook (including code snippets and functions) or any rights granted hereunder to any other person.

You shall not duplicate this eBook, except for backup or archival purposes. You shall not remove any proprietary notices, labels, or marks from this eBook and transfer to any other party.

The code snippets and functions provided in this book are for teaching and educational research, i.e. for non commercial use. It is not allowed to use the provided code snippets and functions for any commercial use. This includes workshops, seminars, courses, lectures, or any other events. The unauthorized use or distribution of copyrighted or other proprietary content from this eBook is illegal.

### *Intellectual Property Protection*

This eBook is owned by the *Rmetrics Association* and *Finance Online* and is protected by international copyright and other intellectual property laws.

*Rmetrics Association* and *Finance Online Zurich* reserve all rights in this eBook not expressly granted herein. This license and your right to use this eBook terminates automatically if you violate any part of this agreement. In the event of termination, you must destroy the original and all copies of this eBook.

### *General*

This agreement constitutes the entire agreement between you and *Rmetrics Association* and *Finance Online Zurich* and supersedes any prior agreement concerning this eBook. This agreement is governed by the laws of Switzerland.

(C) 2009, 2010 Rmetrics Association. All rights reserved.

## APPENDIX E

# NSE TERMS OF LEGAL USE

### DATA AND INFO VENDING PRODUCTS OF NSE

Information and terms of legal use can be found on NSEs *Data and Info vending Products of NSE* web page.

[http://www.nseindia.com/content/press/prs\\_datavending.htm](http://www.nseindia.com/content/press/prs_datavending.htm)

"The data and info-vending services of National Stock Exchange (NSE) are provided through DotEx International Limited (DoteEx) which is a separate professional set-up dedicated solely for this purpose. DotEx currently provides the following products."

#### LISTING E.1: DOTEX DATA PRODUCTS

---

NSE online Real time Data Feed.  
NSE 5-minute and 2-minute Snapshot Data  
NSE End of Day data  
Historical Data  
Corporate Data

---

For information on the individual products we refer to the web page mentioned above.

#### NSE END OF DAY DATA

For end-of-day data NSE generates binary formatted files at the end of each trading day. Technical specifications for the end-of-day data can be found in the following documents

[http://www.nseindia.com/content/press/EOD\\_CM&FA0.pdf](http://www.nseindia.com/content/press/EOD_CM&FA0.pdf)  
[http://www.nseindia.com/content/press/EOD\\_WDM.pdf](http://www.nseindia.com/content/press/EOD_WDM.pdf)

End-of-day data is also made available through NSEs web interface which we discussed in this eBook.

#### HISTORICAL DATA PROVIDED ON CD/DVDs

For research work in the capital market, derivatives market, and debt market, NSE provides historical trades data, high frequency tick by tick data, on CDs/DVDs.

NSE notes that the historical data "CDs/DVDs are being provided for non-commercial and personal use for research purposes. Under no circumstances should unauthorized copies be made, data transferred to anyone or distributed in any manner.

Although not explicitly stated on their Internet portal, the same terms of legal use will be also valid for the end-of-day data provided on the internet. For further details please contact NSE 0091-22-26598163.

## BIBLIOGRAPHY

Saif Siddiqui, *Examining Associations between S&P CNX Nifty and selected Asian & US Stock Markets*, 36 p., 2009

Dotex International Ltd., *EoD Data Technical Specification, Wholesale Debt Market*, 10 p., 2009

Dotex International Ltd., *EoD Data Technical Specification, Capital Market and Futures and Option Markets*, 15 p., 2009

IISL, *Facilitating Growth*, 8 p., 2009

National Stock Exchange of India, *Indian Stock Securities, A Review*, 221 p., 2009

National Stock Exchange of India, *Fact Book 2009*, 189 p., 2009

# INDEX

## keywords

- as, 42
- character, 70
- data, 75
- end, 70
- for, 70
- index, 70
- integer, 70
- start, 70
- symbol, 70
- time, 70, 75

timeDate, 66

timeSeries, 45, 66, 67

## R classes

timeDate, 66

timeSeries, 11, 21, 47, 57, 67

## R functions

abbreviate, 50

bseDownload, 38, 42

msciOnvistaDownload, 56

nseDebtDownload, 21

nseEquityDownload, 16, 21

nseIndexDownload, 13, 21

nseListing, 30

oandaDownload, 62

R.csv, 16

read.csv, 15, 16

read.lynx, 15, 36, 72

readLines, 27

tableSplit, 28

yahooSeries, 48, 51

## R packages

fBasics, 68

fImport, 48, 51, 69

## ABOUT THE AUTHORS

**Diethelm Würtz** is private lecturer at the Institute for Theoretical Physics, ITP, and for the Curriculum Computational Science and Engineering, CSE, at the Swiss Federal Institute of Technology in Zurich. He teaches Econophysics at ITP and supervises seminars in Financial Engineering at CSE. Diethelm is senior partner of Finance Online, an ETH spin-off company in Zurich, and co-founder of the Rmetrics Association.

**Mahendra Mehta** is currently managing director with Neural Technologies and Software private Limited (NeuralTechSoft) based in Mumbai (India). The company is engaged in Risk Management, Risk Analytics and related software development Mahendra also teaches several Risk Management and related courses in many countries around the world. Mahendra was previously head of analytics at Citibank (India). He holds a doctoral degree in Electrical engineering.

**Andrew Ellis** read neuroscience and mathematics at the University in Zurich. He is working for Finance Online and is currently doing a Student Internship in the Econophysics group at ETH Zurich at the Institute for Theoretical Physics. Andrew is managing the Rmetrics documentation project.

**Yohan Chalabi** has a master in Physics from the Swiss Federal Institute of Technology in Lausanne. He is a PhD student in the Econophysics group at ETH Zurich at the Institute for Theoretical Physics. Yohan is a senior R code developer and a co-maintainer of the Rmetrics packages.

