

〔資 料〕

## R による金融データベースへのアクセス

—Yahoo! の場合—

地 道 正 行  
村 上 卓

## I はじめに

近年のコンピュータ環境のめざましい発展によって、ある程度規模の大きなデータを様々な方法によって分析することが容易になってきており、手法の複雑さやデータ量の多さは、通常の教育・研究に関するコンピュータ環境においても障壁はなくなりつつある<sup>1)</sup>。一方、分析の主体であるデータをどのように取得するかについては、市販のデータベース等は価格の面での問題もあり、それほど容易ではなかった。しかしながら、近年“Yahoo! Finance”等のインターネットを通じた株価などの金融情報の提供が行われており、Web ブラウザなどを利用することによって、手軽にデータを取得することができるようになってきている。ただし、ここでもデータを取得する際にその操作手順に関する「手間」の問題があり、効率的に取得することについて議論をする価値は十分にあると思われる。本稿では、この点をふまえて、フリーのデータ解析環境である R とそのアドオンパッケージである Rmetrics を利用することによってインターネットを介して提供されている金融データベースへアクセスし、効率的にデータを取得するためのいくつかの情報を提供することを目的とする。また、日本における“Yahoo! ファイナンス”サイト

1) 本稿におけるデータは、企業の株価や財務諸表などで扱われる数万件程度の規模のものを想定している。

への R を用いたアクセスについては独自の対応策を紹介する。なお、熊谷、舟尾 (2007-b) でも本稿とは異なった株価データの取得法が紹介されているのであわせて参照されたい。

本稿の構成は以下のようなものである。まず、II 節では R と Rmetrics について紹介した後、Rmetrics を用いて米国の “Yahoo! Finance” サイトへアクセスし、株価のデータを取得するための方法をデモンストレーションを交えて述べる。次に、III 節で、日本における “Yahoo! ファイナンス” サイトへ R を用いてアクセスするために作成した関数について述べ、実際に日経平均株価や企業の株価のデータをダウンロードする方法を述べる。最後に IV 節でまとめと今後の展望について述べる。なお、本稿で扱うソフトウェアのバージョンは、R 2.6.1, Rmetrics 260.72 を利用していることに注意しよう。

## II R と Rmetrics

### 1. R とは

R はデータ解析とグラフィックに関する環境 (environment) の一つであり、当初はオークランド大学統計学科の R. Gentleman と R. Ihaka によって開発されていたけれども、現在は “The R Core Team” を中心に保守や改良などが行われている<sup>2)</sup>。R は各種の OS<sup>3)</sup> に対応しており、Windows、Linux、Mac OS などの代表的なものに関してはバイナリ形式のファイルが提供されている。R は FSF<sup>4)</sup> によって制定された GNU<sup>5)</sup> General Public License<sup>6)</sup> の意

2) R の開発プロジェクトの URL は <http://www.r-project.org/> である。

3) Operating System の略。

4) Free Software Foundation の略。R. M. Stallman を中心とするコンピュータ・プログラムの複写や再配布の制限、理解と修正の制限をなくすことを目的にした団体。

5) Gnu is Not Unix の略。Unix 互換のソフトウェア環境を全てフリーソフトウェアで実装することを目的とした FSF によって進められているプロジェクト。

6) 一般公衆利用許諾契約書のこと。単に、GPL と略される。なお、GPL における自由 (free) の意味を <http://www.gnu.org/philosophy/free-sw.ja.html> から以下に引用する：

- ・目的を問わず、プログラムを実行する自由 (第 0 の自由)。
- ・プログラムがどのように動作しているか研究し、そのプログラムにあなたの必要に応じて修正を加え、採り入れる自由 (第 1 の自由)。ソースコードが入手

味でのフリーソフトウェアであり、再配布が許されているなど緩い条件のもとで利用可能である。このことから、大学などの教育・研究機関において非常に有益なソフトウェアであるといえる。Rは、最寄りのCRAN<sup>7)</sup>サイトからインターネット経由で入手可能であり、日本語版のインストールに関してはRjpWiki<sup>8)</sup>サイトから情報を得ることができる。なお、Rのインストール用CDが付属した入門書も出版されているので、初学者はこのような文献を利用するのが簡単であろう。(たとえば、舟尾、高浪(2006)。)また、PDF、HTMLなどの各種のファイル形式でオンラインマニュアルがRに標準で付属しており、英語で記述されているという点を除けば最も手軽で有用なものである。Rに関する書籍としては近年多くのもので出版されており、たとえば、Everitt & Hothorn(2006)は入門のみならず応用の面でも優れた良書である。また、Crawley(2007)は多変量解析、時系列解析などを含む基本的な統計手法を網羅した包括的な文献であり900頁を超える大部のものである。さらに、Rを用いた豊富な事例研究に関しては岡田ら(2004)が、機械学習やデータマイニングなどの最新の手法について言及したものに熊谷、舟尾(2007-a, b)、金(2007)があり、プログラミングに関するものとしてはLigges(2004)、間瀬(2007)が参考になる。なお、設計は同じであるけれども、異なった実装であるS言語<sup>9)</sup>に関する文献もRの利用に際して参考

---

可能であることはこの前提条件となります。

- ・身近な人を助けられるよう、コピーを再頒布する自由(第2の自由)。
- ・プログラムを改良し、コミュニティ全体がその恩恵を受けられるようあなたの改良点を公衆に発表する自由(第3の自由)。ソースコードが入手可能であることはここでも前提条件となります。

- 7) Comprehensive R Archive Networkの略。R関連のソフトウェアやパッケージ、マニュアルなどを集めたインターネット上のサイト。<http://www.r-project.org/>が大本であり、全世界にミラーサーバーがある。
- 8) Rに関する情報交換を目的としたWikiサイト。<http://www.okada.jp.org/RWiki/>
- 9) J. W. Tukeyによって探索的データ解析(Exploratory Data Analysis; EDA)が提唱され、それを実行するための環境がJ. M. Chambersによって設計され、Sと名付けられた。さらにR. A. BekkerとJ. M. ChambersによってSシステム(後にS言語)が、R. GentlemanとR. IhakaによってR言語がその実装として与えられた。なお、市販のものとしてS-PLUSがInsightful社から販売されている。

になり、Chambers & Hastie (1991)、Chambers (1998) は S 言語を使ったデータ解析やモデリングに関して、また、Venables & Ripley (2002) は S、R を使ったデータ解析、統計解析全般にわたる豊富な情報を与えてくれる。

## 2. Rmetrics とは

Rmetrics<sup>10)</sup> は、金融市場の分析や金融機関の評価のために開発された世界初のオープン・ソース・ソリューションであり、チューリヒ工科大<sup>11)</sup> の D. Würtz によって金融工学の講義のために作成されたものである。Rmetrics は、R のアドオンパッケージとして提供されており、以下のような枠組みから構成される：

- A) データセット、ユーティリティ関数
- B) 年代順オブジェクト (chronological object)、時系列
- C) リターンとボラティリティの分析
- D) テクニカル分析、意志決定、予測
- E) 極値理論、コピュラ (copula)、リスクマネジメント
- F) 金融手法の評価
- G) 資産管理、ポートフォリオ分析、最適化

なお、詳細については Rmetrics に付属のオンラインマニュアルやヘルプを参照されたい。

Rmetrics を利用するためには、R を起動した後に以下のように関数 library を利用してパッケージ Rmetrics をロードする必要がある<sup>12)</sup>：

10) <http://www.itp.phys.ethz.ch/econophysics/R/index.html>

11) Swiss Federal Institute of Technology in Zurich (ETH Zurich)

12) Rmetrics は R をインストールする際に標準ではインストールされないなので、別途、最寄りの CRAN サイトからダウンロードし、インストールされたい。

```

> library(Rmetrics)
  要求されたパッケージ fEcofin をロード中です
  要求されたパッケージ fUtilities をロード中です
  要求されたパッケージ RUnit をロード中です
  ⋮
  ⋮ (略)
  ⋮
Package fPortfolio (260.72) loaded.
Rmetrics - Portfolio Selection and Optimization
Rmetrics, (C) 1999-2007, Diethelm Wuertz, GPL

```

ここで、> は R のプロンプトである<sup>13)</sup>。

### 3. Rmetrics による “Yahoo! Finance” サイトへのアクセス

Rmetrics には、経済、金融に関するデータをインターネット経由で取得するためのいくつかの関数が用意されている。例えば、yahooImport という関数を利用することによって米国における “Yahoo! Finance” のサイト<sup>14)</sup> から株価に関するデータをダウンロードできる<sup>15)</sup>。

具体的な例として、2006年1月3日から2008年1月10日までの IBM の NYSE<sup>16)</sup> における日次の株価を取得することを考える。Rmetrics をロード後、以下のように入力する：

```

> query <- "s=IBM&a=0&b=3&c=2006&d=0&e=10&f=2008&z=IBM&x=.csv"
> IBM <- yahooImport(query)

```

1行目の入力は、データベースへの問合せ (query) のための引数 (argument) を文字列 (character) のオブジェクト<sup>17)</sup> query として事前に用意し

13) R では、関数などの命令をプロンプト > に続いて入力し、**Enter** キーをタイプすることによって命令を実行し、その結果が出力されるという形式で処理を行う。従って、R は「対話型」、「関数型」言語の一つといえる。

14) <http://quote.yahoo.com/>

15) このほかにも、経済時系列データを提供するサイトである Economagic (<http://www.economagic.com/>) からデータをダウンロードする関数である economagicImport, economagicSeries などが用意されている。

16) New York Stock Exchange (ニューヨーク証券取引所) の略。

17) R はすべての「対象」をオブジェクト (object) として扱う、いわゆるオブジェクト指向言語 (object oriented language) である。

ている。ここで “<-” は、代入（付値; assignment）することを表す演算子である。次に、2行目の入力は、1行目で得られたオブジェクト `query` を関数 `yahooImport` の引数に与えることによって実行し、その結果をオブジェクト `IBM` に代入している。この入力によって、R から HTTP<sup>18)</sup> によってコネクションが “Yahoo! Finance” サイトに張られ、株価に関するデータがダウンロードされる。結果として得られるオブジェクト `IBM` は、509行7列の行列形式<sup>19)</sup> の株価に関するデータを含んでいる<sup>20)</sup>。関数 `head` を使って最初の6日分の株価データを表示させると以下のようなになる：

```
> head(IBM@data)
      X.Y..m..d Open High  Low Close  Volume Adj.Close
509 2006-01-03 82.45 82.55 80.81 82.06 11715100      79.85
508 2006-01-04 82.20 82.50 81.33 81.95  9832800      79.74
507 2006-01-05 81.40 82.90 81.25 82.50  7213400      80.28
506 2006-01-06 83.95 85.03 83.41 84.95  8196900      82.66
505 2006-01-09 83.90 84.25 83.38 83.73  6851100      81.48
504 2006-01-10 83.15 84.12 83.12 84.07  5700000      81.81
```

次に、ここで得られた株価のオブジェクトを関数 `as.timeSeries` を使って時系列データを扱うために `Rmetrics` で用意されている専用のクラス `timeSeries`<sup>21)</sup> に属するオブジェクト `IBM.ts` に変換し、そのオブジェクトの2列目のデータ (High; 「高値」を表す。) の時系列プロットを関数 `plot`<sup>22)</sup>

18) Hyper Text Transfer Protocol (ハイパーテキスト転送プロトコル) の略。

19) より正確にはデータフレーム (data frame) オブジェクトである。

20) オブジェクト指向言語の特徴の一つにクラス (class) という概念がある。Rにおいてクラスはオブジェクトを定義するために利用され、スロット (slot) という仕組みを使って幾つかのクラス固有の情報を持つように設計されている。ここで与えられたオブジェクト `IBM` は、株価データを `data` というスロットに格納しており、スロットをオブジェクトから抜き出すには、演算子 “@” を用いて `IBM@data` のようにする。また、このオブジェクトは、このほかにも `call`、`param`、`title`、`description` など、合計5つのスロットをもっている。なお、このオブジェクト自体は `fWEBCDATA` というクラスに属しており、関数 `class` を使ってクラス名を調べることができる。また、オブジェクトの構造 (structure) は、関数 `str` を使って詳細に調べることができる。

21) このクラスは多変量時系列データ (multivariate time series data) を扱うためのものである。

22) 実際には、`Rmetrics` に付属の関数 `plot.timeSeries` を利用してプロットが行われ

を使って行くと以下のように与えられる：

```
> IBM.ts <- as.timeSeries(IBM@data)
> plot(IBM.ts[,2],type="l",xlab="Year",ylab="High")
> title("IBM Stock Price")
```

このような入力によって、図1のようなグラフが専用のウィンドウ<sup>23)</sup>に描かれる。

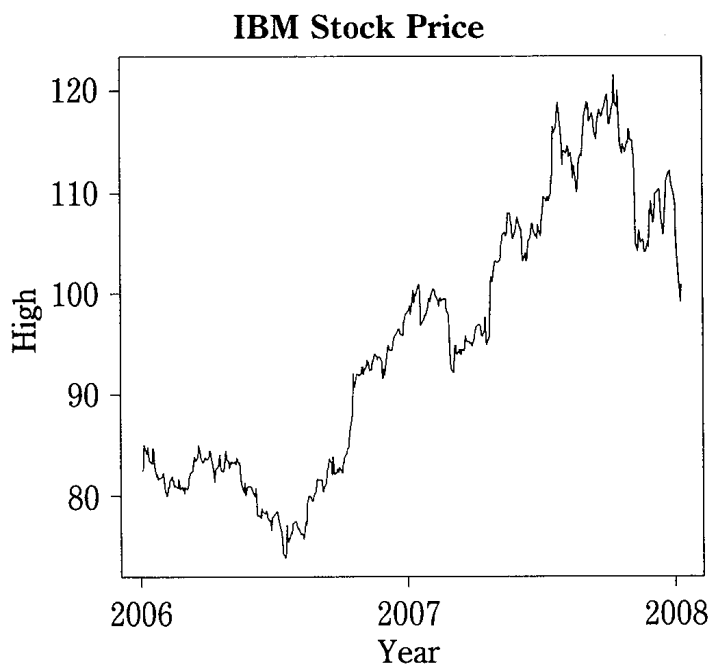


図1：2006年1月3日から2008年1月10日までのIBMのNYSEにおける高値（日次）の時系列プロット

オブジェクト `IBM.ts` には、高値の他にも始値（Open）、安値（Low）、終値（Close）、出来高（Volume）、補正終値（Adj.Close）などの情報が含まれているので、これらを3行2列の行列形式にプロットする関数

`plot.yahoo` を作成し、それを用いて描画すると図2のように与えられる：

---

る。これは、関数 `plot` が引数として与えられたオブジェクトのクラスを自動的に判断し、適切と思われる関数を呼び出す仕組み（メソッド（method）といわれる。）によって実現されている。このような関数（ここでは `plot`）を総称関数（generic function）よぶ。

23) Rを利用してプロットを行う場合、初期の描画時に、“R Graphics”という専用ウィンドウが画面上にポップアップする。

```

> plot.yahoo <- function(obj, ttl)
+ {
+   ts.obj <- as.timeSeries(obj@data)
+   labels <- dimnames(ts.obj)[[2]]
+   par(mfcol=c(3,2))
+   for (i in 1:6) plot(ts.obj[,i], type="l", xlab="Year", ylab=labels[i])
+   par(mfcol=c(1,1))
+   title(ttl)
+ }
> plot.yahoo(IBM, ttl="IBM Stock Prices")

```

ここで、par はプロットに関するパラメータ (parameter) を設定するための関数であり、非常に細かい箇所まで設定することが可能である。なお、詳細は R のヘルプや舟尾 (2005) を参照のこと。

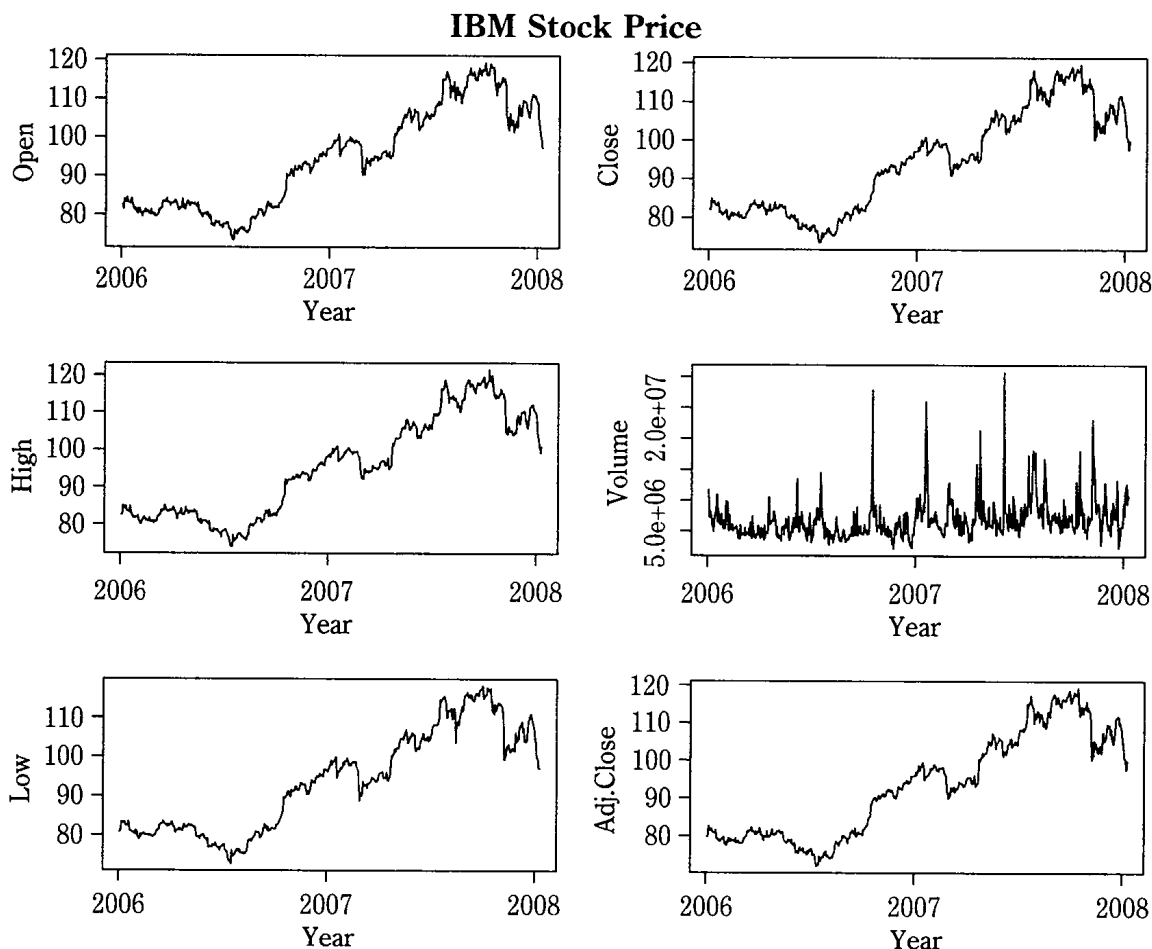


図 2 : 2006年 1 月 3 日から 2008年 1 月 10 日までの IBM の NYSE における株価の時系列プロット : 1 列目を上から下へ順に、始値 (Open)、高値 (High)、安値 (Low) のプロットが並んでおり、同様に 2 列目は終値 (Close)、出来高 (Volume)、補正終値 (Adj.Close) のプロットが並んでいる。



### Ⅲ “Yahoo! ファイナンス” サイトへのアクセス

#### 1. “Yahoo! Finance” と “Yahoo! ファイナンス” の相違点

Rmetrics はインターネット経由で株価などの情報を得ることができるという点で非常に便利な機能を提供してくれるけれども、残念ながら日本における “Yahoo! ファイナンス” サイト<sup>24)</sup> へのアクセスには対応していない。これは “Yahoo! Finance” のサイトがデータをスプレッドシート (csv 形式) のファイルとして提供してくれているのに対し、“Yahoo! ファイナンス” のサイトは現時点では提供していないことによる。この問題に対して “Yahoo! ファイナンス” サイトにアクセスし、株価などのデータを取得するための関数を作成することを考える。

#### 2. 株価情報の取得に関する R 関数

ここでは、以下のような R の関数 `fetchYahooJapanFinance` を作成した。なお、この関数のソースコードは付録を参照のこと。

```
fetchYahooJapanFinance(code, begin, end, unit, what, kindness)
```

各引数は以下のようなものである：

`code` 銘柄コードや証券コードを与える。デフォルト<sup>25)</sup> は "998407" (日経平均株価) である。

`begin` データを取得したい期間の始め。デフォルトは "2008-1-1" (実行年の1月1日) である。

`end` データを取得したい期間の終わり。デフォルトはこの関数を実行した日が指定される。(たとえば、"2008-1-10"。)

`unit` 日次 (daily)、週次 (weekly)、月次 (monthly)。デフォルトは

24) <http://quote.yahoo.co.jp/>

25) 引数の省略時の値を指す。「債務不履行」などの金融関連の専門用語との混乱はないであろう。

"daily" (日次) である。

what “株価”、“信用残”、“為替” のデータのいずれかを指定 (“株価” と “為替” は同じ)。デフォルトは “株価” (株価時系列データ) である。

kindness 休止時間 (秒) を指定。デフォルトは 0 (0 秒) である。

なお、この関数の処理の流れを図 3 に与える。

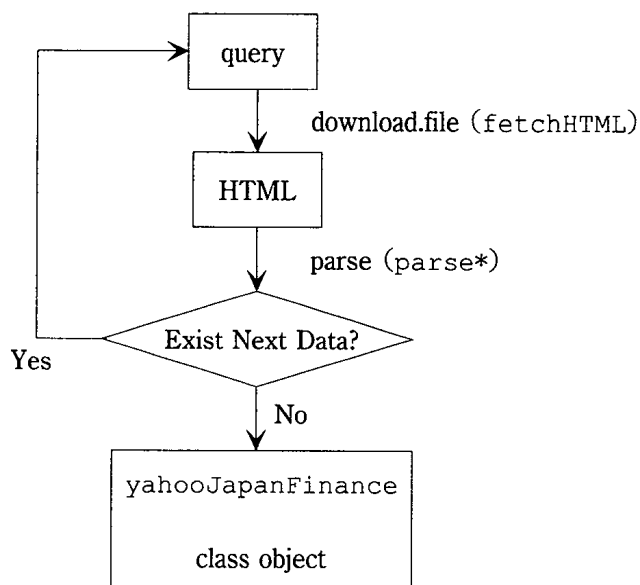


図 3 : 関数 `fetchYahooJapanFinance` の処理の流れ

### 3. 例：日経平均株価の取得とその時系列プロット

関数 `fetchYahooJapanFinance` はデフォルトで日経平均株価を取得できる。では実際に2007年1月4日から2008年1月22日までの日経平均株価を取得する例を以下に与えよう。

```

> nikkei <- fetchYahooJapanFinance(begin="2007-1-4", end="2008-1-22")
> nikkei
{ 株価時系列データ }
コード:          998407.0
名称. 市場:      日経平均株価 . 大証
間隔 (unit):     d (d:daily, w:weekly, m:monthly)
取得期間:       2007-01-04 -- 2008-01-22
データ数:       257
slot names:     @nameData @dataKind @data
  
```

この取得したデータのオブジェクト `nikkei` の時系列プロットは関数 `plot` を使って描くことができる<sup>26)</sup>。(図4も参照のこと。)

```
> plot(nikkei)
```

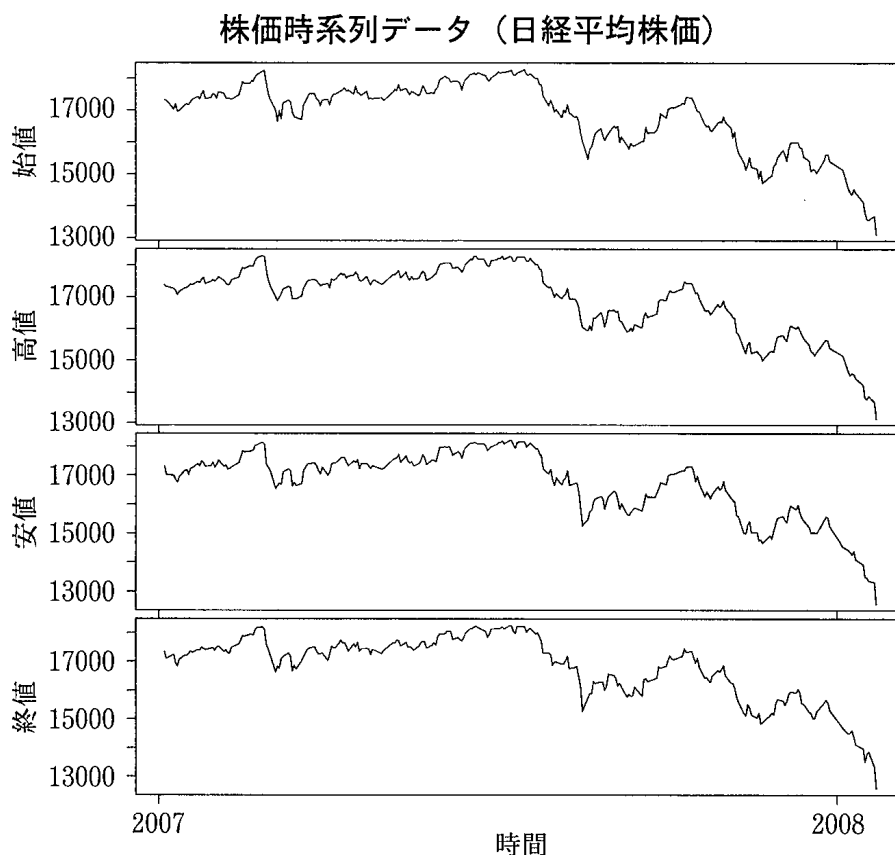


図4：2007年1月4日から2008年1月22日現在までの日経平均株価（始値・高値・安値・終値）の時系列プロット

#### 4. 例：楽天の株価の取得とその時系列プロット

特定企業の株価の情報を取得する例を与える。ここでは、企業として「楽天」を選び、期間としては2007年1月4日から2008年1月22日としよう。関数 `fetchYahooJapanFinance` で「楽天」の日経会社コード（4755）を示す "4755" を引数 `code` に与え、引数 `begin` に "2007-1-4", `end` に "2008-1-22" を与えることによって取得できる：

26) ここでは、`yahooJapanFinance` というクラスを定義し、そのクラスに対して `plot` メソッドを定義することによって描画している。詳しくは付録を参照されたい。

```

> rakuten <- fetchYahooJapanFinance(code="4755", begin="2007-1-4", end="2008-1-22")
> rakuten
{ 株価時系列データ }
コード:          4755.Q
名称. 市場:      楽天 (株) . JASDAQ
間隔 (unit):     d (d:daily, w:weekly, m:monthly)
取得期間:       2007-01-04 -- 2008-01-22
データ数:       257
slot names:     @nameData @dataKind @data

```

この取得したデータのオブジェクト rakuten の時系列プロットは以下のように入力することによって与えられる。(図5も参照のこと。)

```
> plot(rakuten)
```

株価時系列データ (楽天株)

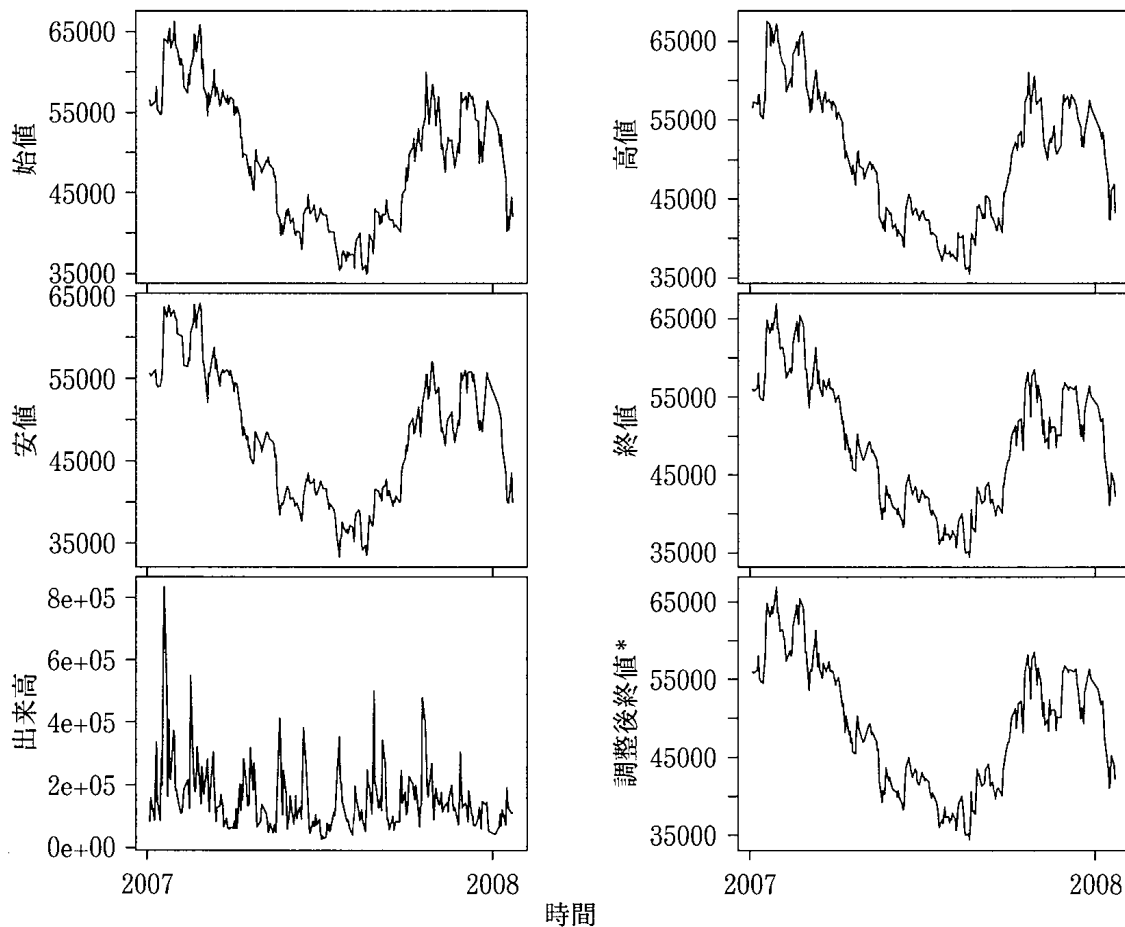


図5：楽天株の2007年1月4日から2008年1月22日現在までの株価（始値・高値・安値・終値・出来高・調整後終値\*）の時系列プロット

## 5. コードと銘柄の検索

一般に、名称（銘柄など）に関するキーワード（例えば“任天堂”など）はわかっているけれども、日経会社コードがわからないため、コードや名称をもとに検索する機能があれば便利である。そのような要求に対して作成した関数が `searchCode` である。この関数の使用法は以下の通りである。

```
searchCode(scn, kindness=0)
```

ここで、`scn` は検索したいコードや名称を指定する。（日本語も可。）

例えば、“任天堂”というキーワードから「任天堂(株)」の日経会社コードを得るためには以下のように入力すればよい。

```
> searchCode("任天堂")
      コード 名称
[1,] "7974" "任天堂(株)"
```

この場合は、“任天堂”というキーワードをもつ銘柄が一つしかないため、このような結果となったが、該当する銘柄が複数存在する場合には、それらすべてについての情報が得られる。例えば“香港”というキーワードで検索すると以下のように“香港オープン”や“香港ドル”などを含む6個の銘柄に関する情報が表示される。

```
> searchCode("香港")
      コード      名称
[1,] "HKDJPY=X" "香港  ドル"
[2,] "01312927" "(オーロラ F) 香港投資 F"
[3,] "0131504A" "(野村ブル・ベア S3) 香港ブル 3"
[4,] "0131604A" "(野村ブル・ベア S3) 香港ベア 3"
[5,] "0631292C" "香港オープン"
[6,] "79311074" "三井住友・中国 A 株・香港株オープン"
```

なお、この関数は日経会社コードから銘柄も検索できることに注意しよう。

## IV おわりに

本稿では、Rを用いて米国における“Yahoo! Finance”サイトと日本にお

ける“Yahoo! ファイナンス”サイトから効率よく株価データを取得するための方法について述べた。特に、“Yahoo! ファイナンス”サイトに関しては独自に作成した関数によってデータを取得する工夫を紹介した。この関数の改良点としては、日経会社コード（仮引数）をベクトル（実引数）で与えることによって複数の会社の株価データを同時に取得することがある。

今後の課題としては、R と XBRL<sup>27)</sup> が協調することによって、EDINET<sup>28)</sup> などのサイトから有価証券報告書に関する情報を取得することを支援するアドオンパッケージの開発や、日経 NEEDS などの商業用のデータベースも視野に入れたデータの効率的な取得などがあるであろう。

筆者 地道正行：関西学院大学商学部教授

村上 卓：関西学院大学大学院商学研究科博士課程前期課程

付記：本稿は2007年8月7日に大阪教育大学で開催された国際数理科学協会年次大会での報告（タイトル：「Rによる金融・財務データベースへのアクセス」）における原稿に加筆・修正を加えたものである。

## 謝辞

XBRL の存在については本学商学部の井上達男先生からご教示いただきました。また、S、R の開発に関する歴史的背景などの詳細に関しては、オークランド大学統計学科の R. Ihaka 先生からご教示いただきました。ここに感謝の意を表します。

27) eXtensible Business Reporting Language の略。各種財務報告用の情報を作成・流通・利用できるように標準化された XML ベースの言語。XBRL 仕様 (XBRL Specification) は、ソフトウェアやプラットフォームに関係なく、電子的な財務情報の作成や流通・再利用を可能することを目的としている。世界的には、XBRL International が、そして日本では XBRL Japan が標準化や普及活動を行っている。

28) Electronic Disclosure for Investors' NETwork の略。(https://info.edinet.go.jp/) 金融商品取引法に基づく有価証券報告書等の開示書類に関する電子開示システムのこと。金融庁より行政サービスの一環として提供されており、提出された開示書類について、インターネット上においても閲覧が可能である。なお、金融庁は、「有価証券報告書等に関する業務の業務システム最適化計画 (案)」を公表 (2006年1月16日発表) しており、システムの最適化は2008年度初頭を目途に実施される予定となっている。

## 参考文献

- [1] Chambers, J. M. (1998) *Programming with Data: A Guide to the S Language*, Springer-Verlag.  
(邦訳：垂水共之他訳 (2002) 『データによるプログラミング—データ解析言語 S における新しいプログラミング—』, 森北出版.)
- [2] Chambers, J. M. and T. J. Hastie (Editor) (1991) *Statistical Models in S*, Chapman & Hall/CRC.  
(邦訳：柴田里程訳 (1994) 『S と統計モデル—データ科学の新しい波—』, 共立出版.)
- [3] Everitt, B. S. and T. Hothorn (2006) *A Handbook of Statistical Analysis Using R*, Chapman & Hall.
- [4] Crawley, M. J. (2007) *The R Book*, Wiley.
- [5] 舟尾暢男著 (2005) 『The R Tips』, 九天社.
- [6] 舟尾暢男, 高浪洋平共著 (2006) 『データ解析環境「R」』, 工学社.
- [7] 地道正行著 (2001) 『R と Rweb』, 情報科学研究, 第16号, 関西学院大学情報メディア教育センター, pp. 21-43.
- [8] 金明哲著 (2007) 『R によるデータサイエンス—データ解析の基礎から最新手法まで—』, 森北出版株式会社.
- [9] 熊谷悦生, 舟尾暢男共著 (2007-a) 『R で学ぶデータマイニング I—データ解析の視点から—』, 九天社.
- [10] 熊谷悦生, 舟尾暢男共著 (2007-b) 『R で学ぶデータマイニング II—シミュレーションの視点から—』, 九天社.
- [11] Ligges, U. (2004) *Programmieren mit R*, Springer-Verlag.  
(邦訳：石田基広訳 (2006) 『R の基礎とプログラミング技法』, シュプリンガー・ジャパン株式会社.)
- [12] 間瀬茂著 (2007) 『R プログラミングマニュアル』, (新・数理工学ライブラリ, 情報工学=1), 数理工学社.
- [13] 岡田昌史編著 (2004) 『The R book—データ解析環境 R の活用事例集

—』, 九天社.

- [14] R Development Core Team (2007) R Data Import/Export, <http://www.r-project.org/>, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-10-0.
- [15] RjpWiki, <http://www.okada.jp.org/RWiki/>.
- [16] Venables, W. N. and B. D. Ripley (2002) *Modern Applied Statistics with S*, Fourth Edition. Springer, New York.  
(第3版邦訳: 伊藤幹夫他訳 (2001) 『S-PLUS による統計解析』, シュプリンガー・フェアラーク東京.)
- [17] Wüertz, D. (2007) *The Rmetrics package*, <http://www.rmetrics.org/>.

## 付録

以下に本稿で紹介した自作関数等のソースコードを与える。なお、“#”以降はコメント行である。

```
### HTML タグを削除
deleteHTMLtags <- function(html)
{
  html <- gsub("<[^>]+>|^[^<>]+>|<[^>]+>$", "", html)
}
### 3 桁区切りスタイルを解除 (削除)
delete3keta <- function(num, del=",")
{
  num <- grep(paste("[0-9]", del, "+", sep=""), num, value=TRUE)
  num <- gsub(paste("[1-9]{1,3}", del, "((([0-9]{3})", del, ")*)"([0-9]{3}))", sep=""), "\\1\\2\\4", num)
  num <- gsub(del, "", num)
  return(num)
}
### 引数に与えられた日付のフォーマット変更など ###
# %Y%m%d ("YYYY-MM-DD"形式を返す)
parse.date.Ymd <- function (Ymd)
{
  Ymd <- gsub(
    "[^[:digit:]]*([[:digit:]]{4})[^[:digit:]]*([[:digit:]]{1,2})[^[:digit:]]*([[:digit:]]{1,2})[^[:digit:]]*$",
    "\\1-\\2-\\3", Ymd)
  Ymd <- as.Date(Ymd, format="%Y-%m-%d")
  return(Ymd)
}
# %Y%m ("YYYY-MM"形式を返す)
parse.date.Ym <- function (Ym, addDate)
```



```

{
  addDate <- as.numeric(addDate)
  if (1<=addDate && addDate<=31) {
    Ym <- gsub(
      "[^[:digit:]]*([[:digit:]]{4})[^[:digit:]]*([[:digit:]]{1,2})[^[:digit:]]*$",
      "\\1-\\2", Ym)
    if (!addDate) {
      Ym <- as.Date(Ym, format="%Y-%m")
    } else {
      Ym <- paste(Ym, paste("-", addDate, sep=""), sep="") # 日を加えないとプロットできないので
      Ym <- as.Date(Ym, format="%Y-%m-%d")
    }
    return(Ym)
  } else {
    stop("日にちの追加に失敗しました (addDate=numeric). ")
  }
}

### HTML ファイルをダウンロード
fetchHTML <- function(url, fromEncoding, toEncoding="")
{
  htmlfile <- tempfile()
  handle <- download.file(url, htmlfile, method="auto", quiet=TRUE)
  if (handle != 0) {
    unlink(htmlfile)
    warning(paste("download error, status", status, ", instrument", instrument))
    return(NULL)
  }
  htmlLines <- readLines(htmlfile, warn=FALSE)
  #file.remove(htmlfile) # ファイルの削除
  htmlLines <- iconv(htmlLines, fromEncoding, toEncoding)
  return(htmlLines)
}

# URL エンコード
# URLEncode()は、日本語に対応していない
jaURLEncode <- function(jaCharacter, fromEncoding="", toEncoding="")
{
  jaCharacter <- iconv(jaCharacter, fromEncoding, toEncoding)
  jaCharacter <- paste("", charToRaw(jaCharacter), sep="%", collapse="")
  return(jaCharacter)
}

##### Yahoo!ファイナンス #####
##### クラスの定義 #####
setClass(Class="YahooJapanFinance",
  representation = representation(
    nameData = "character", # 会社名, 上場場所, コード
    dataKind = "character",
    data = "data.frame" # データ
  )
)

### メソッド (method) ###

```

```

# "show"
setMethod("show", signature(object = "YahooJapanFinance"),
  function(object){
    dateCol <- 1
    whatData <- object@dataKind[1]
    code <- object@nameData[3]
    name <- object@nameData[1]
    market <- object@nameData[2]
    unit <- object@dataKind[2]
    begin <- format(object@data[1,dateCol], "%Y-%m-%d")
    end <- format(object@data[nrow(object@data),dateCol], "%Y-%m-%d")
    cat("{",whatData,"}", "\n") # [株価時系列データ/信用残時系列データ/株価時系列データ]
    cat("コード:\t\t", code, "\n") # 銘柄コード
    cat("名称.市場:\t", name, ".",market,"\n") # 会社名. 上場場所
    cat("間隔(unit):\t", unit, "(d:daily, w:weekly, m:monthly)", "\n")
    cat("取得期間:\t", begin, "--", end, "\n")
    #cat("\t\t", begin, "-", end, "=", object@data[nrow(n@data),1]-object@data[1,1], "\n")
    cat("データ数:\t", nrow(object@data), "\n")
    cat("slot names:\t", paste("@",slotNames(object),sep=""),"\n") # スロットの名前リスト
    cat("\n")
  }
)
# 時系列"plot"
setMethod("plot", signature(x="YahooJapanFinance", y="missing"), # functionArgNames("plot")
  function(x, y, ...) {
    dateCol <- 1
    ngraph <- ncol(x@data) - 1
    oldPar <- par(no.readonly = TRUE)
    parOma <- c(4.7, 1, 4, 1) + 0.1
    parMar <- c(0, 4, 0.3, 1) + 0.1
    if (ngraph<=5) {
      plotNcol <- 1
      par(mfrow=c(ngraph,plotNcol), oma=parOma, mar=parMar)
      for (i in 2:c(ngraph+1)){
        if (i <= ngraph+1-plotNcol) {
          plot(x@data[,dateCol], x@data[,i], type="l", ylab=colnames(x@data[i]), xaxt="n")
          #Dateクラス, axis.Date, seq.Date
          axis.Date(side=1, x@data[,dateCol], labels=FALSE)
        } else {
          plot(x@data[,dateCol], x@data[,i], type="l", ylab=colnames(x@data[i]))
        }
      }
    }
    else if (ngraph<=10) {
      plotNcol <- 2
      par(mfrow=c(ngraph%/%2,plotNcol), oma=parOma, mar=parMar)
      for (i in 2:c(ngraph+1)) {
        if (i <= ngraph+1-plotNcol) {
          plot(x@data[,dateCol], x@data[,i], type="l", ylab=colnames(x@data[i]), xaxt="n")
          #Dateクラス, axis.Date, seq.Date
          axis.Date(side=1, x@data[,dateCol], labels=FALSE)
        } else {
          plot(x@data[,dateCol], x@data[,i], type="l", ylab=colnames(x@data[i]))
        }
      }
    }
  }
)

```

```

    }
  } else {
    stop("図の数が多すぎます.", call.=FALSE)
  }
  title(paste(x@dataKind[1], " (", x@nameData[1], ") ", sep=""), xlab="時間", outer=TRUE)
  par(oldPar)
}
)

```

```

##### 関数 (function) #####
### Yahoo!ファイナンスのデータ取得 ###
fetchYahooJapanFinance <- function (
  code = "998407",
  begin = paste(format(Sys.time(), "%Y"), "-01-01", sep=""),
  end = format(Sys.Date(), "%Y-%m-%d"),
  unit = c("daily", "weekly", "monthly", "日間", "週間", "月間",
    "日次", "週次", "月次", "デイリー", "ウィークリー", "マンスリー"),
  what = c("株価時系列データ", "信用残時系列データ", "為替時系列データ",
    "stock price", "outstanding balances of margin trading", "exchange"),
  kindness = 0
)
{
  # データの種類
  what <- match.arg(what)
  switch (what,
    "株価時系列データ" =,
    "stock price" = what <- "株価",
    "信用残時系列データ" =,
    "outstanding balances of margin trading" = what <- "信用残",
    "為替時系列データ" =,
    "exchange" = what <- "為替",
    what <- "株価"
  )
  # 間隔 (デイリー/ウィークリー/マンスリー)
  unit <- match.arg(unit)
  switch (unit,
    "daily" =,
    "デイリー" =,
    "日次" =,
    "日間" = unit <- "d",
    "weekly" =,
    "ウィークリー" =,
    "週次" =,
    "週間" = unit <- "w",
    "monthly" =,
    "マンスリー" =,
    "月次" =,
    "月間" = unit <- "m",
    unit <- "d"
  )
  # 取得期間
  begin <- parse.date.Ymd(begin)
  end <- parse.date.Ymd(end)
  begin <- strsplit(as.character(begin), "-")[1]

```

```

end <- strsplit(as.character(end), "-")[[1]]
# データ種
dataKind <- c(what, unit)
# 休止時間 (秒)/ページ
kindness <- as.numeric(kindness)
# 日付列
dateCol <- 1
# 文字コード
fromEncoding <- "EUC-JP"
toEncoding <- ""
# デバッグ
debugCall <- FALSE
#debugCall <- TRUE
#####
### URL 構築 ###
yjf.url <- function(code=NULL, begin, end, unit, y, what="株価", query=0)
{
  url <- "http://table.yahoo.co.jp/"
  if (what=="信用残") {
    what <- "bt?"
    unit <- "d"
  } else {
    what <- "t?"
  }
  if (query==0) {
    # クエリーの構築
    query <- paste(
      "s=", code,
      "&c=", begin[1], "&a=", begin[2], "&b=", begin[3],
      "&f=", end[1], "&d=", end[2], "&e=", end[3],
      "&g=", unit,
      "&q=q",
      "&y=", y, # 次に50件など
      "&z=", code,
      "&x=.csv",
      sep = ""
    )
    url <- paste(url, what, query, sep="")
  } else {
    url <- paste(url, query, sep="")
  }
  return(url)
}
##### parse 関数の定義 #####
### "株価時系列データ", "信用残時系列データ", "為替時系列データ" ###
yjf.parseWhatData <- function(html)
{
  matchChar <- "^.*<td align=\"center\"[^>]*><b><small>([<]+)</small></b></td>.*$"
  html <- grep(matchChar, html, value=TRUE)
  if (length(html)) {
    html <- sub(matchChar, "\\1", html)
    return(html)
  } else {
    stop("指定されたコード(code)に対するデータ(what)はありません。", call.=debugCall)
  }
}

```

```

    }
  }
  ### 銘柄名・上場場所・コード ###
  yjf.parseNameData <- function(html)
  {
    split <- ","
    handle <- grep("^<tr><td><b>", html)
    if (length(handle)) {
      html <- paste(html, collapse="") # 1行に変換
      html <- sub(".+<tr><td><b>([^\<]+)</b>【([^\<:]+): ? ([^\< ]+)</td>.+",
        paste("\\1", split, "\\2", split, "\\3", sep=""), html)
      html <- strsplit(html, split=split)[[1]]
      return(html)
    } else {
      stop("指定されたコード(code)に対するデータはありません。", call.=debugCall)
    }
  }
  ### 時系列データ ###
  # 列名
  yjf.parseTsDataColNames <- function(html)
  {
    split <- ","
    html <- grep("<th><small>([^\<]+)</small></th>", html, value=TRUE)
    if (length(html)) {
      html <- gsub("<th><small>([^\<]+)</small></th>", paste("\\1", split, sep=""), html)
      html <- deleteHTMLtags(html)
      html <- strsplit(html, split=split)
      if (length(html)!=1) { # 信用残
        html <- paste(html, colaspe="")
      } else { # 株価・為替
        html <- html[[1]]
      }
    }
    return(html)
  } else {
    stop("この検索期間(begin~end)のデータはありません。", call.=debugCall)
  }
}
# 時系列データ
yjf.parseTsData <- function(html, unit="d", dateCol=1)
{
  split <- "s"
  # 列名
  tsDataColNames <- yjf.parseTsDataColNames(html)
  # データ
  html <- grep("^align=right {1,2}bgcolor=\"#ffffff\">", html, value=TRUE)
  if (length(html)) {
    html <- gsub("<[^\>]+>([^\<]+)<[^\>]+>", paste("\\1", split, sep=""), html)
    html <- deleteHTMLtags(html)
    html <- strsplit(html, split=split)
    # データフレームに変換
    html <- as.data.frame(html) # listを表のdata.frameにする
    html <- t(html) # t()はmatrixに変換するので↓
    html <- as.data.frame(html, stringsAsFactors=FALSE) # characterをfactorに変換しない
    # 3桁区切りの削除

```

```

html[,-dateCol] <- apply(html[,-dateCol], c(1,2), delete3keta)
html <- data.frame(html[,dateCol], apply(html[,-dateCol], c(1,2), as.numeric))
# row.names()とrownames()は違う
#rownames(html) <- NULL # 2.4.1ではエラー
#dimnames(html)[[1]] <- NULL
row.names(html) <- NULL
colnames(html) <- tsDataColNames
if (unit=="m") {
  html[,dateCol] <- parse.date.Ym(html[,dateCol], addDate=1)
} else {
  html[,dateCol] <- parse.date.Ymd(html[,dateCol])
}
return(html)
} else {
  stop("データを取得できません.", call.=debugCall)
}
}
}
### 次のページがあるか ###
yjf.parseNextPage <- function(html)
{
  matchChar <- "^.*<a href=\"([^\"]+)\">次の[[:digit:]]{1,2}件</a></small></td>.*$"
  html <- grep(matchChar, html, value=TRUE)
  if (length(html)) {
    html <- sub(matchChar, "\\1", html)
    return(html)
  } else{
    return(FALSE)
  }
}

##### fetch #####
tsData <- NULL
nextPage <- 0
while (TRUE) {
  url <- yjf.url(code, begin, end, unit, y=nextPage, what, query=nextPage)
  html <- fetchHTML(url, fromEncoding, toEncoding)
  dataKind[1] <- yjf.parseWhatData(html)
  nameData <- yjf.parseNameData(html)
  tsData <- rbind(tsData, yjf.parseTsData(html, unit=unit, dateCol=dateCol))
  # 次のページあるか
  nextPage <- yjf.parseNextPage(html)
  if (nextPage==FALSE) {
    break
  }
  Sys.sleep(kindness) #休止時間(秒)/1ページ
}
tsData <- tsData[order(tsData[,dateCol]),]
row.names(tsData) <- NULL
## クラスのインスタンスの生成 ##
OBJ = new("YahooJapanFinance",
  nameData=nameData,
  dataKind=dataKind, data=tsData
)
return (OBJ)

```

```

}

### Yahoo!ファイナンスのコード・名称検索 ###
searchCode <- function(scn, kindness=0)
{
  # scnは検索したいコードもしくは名称(銘柄)
  # 文字コード
  fromEncoding <- "EUC-JP"
  toEncoding <- ""
  # 休止時間(秒)/1ページ
  kindness <- as.numeric(kindness)
  # 検索
  scn <- jaURLEncode(scn, toEncoding, fromEncoding)
  # デバッグ
  debugCall <- FALSE
  #debugCall <- TRUE
  #####
  # URLの構築
  yjc.url <- function(scn, query=0)
  {
    url <- "http://quote.yahoo.co.jp"
    if (query==0) {
      url <- paste(url, "/l?s=", scn, sep="")
    } else {
      url <- paste(url, query, sep="")
    }
    return (url)
  }
  # 列名
  parseCodeColNames <- function(html)
  {
    split <- ", "
    html <- grep("<tr class=chartbg>", html, value=TRUE)
    if (length(html)) {
      html <- gsub("<th>([<]+)</th>", paste("\\1", split, sep=""), html)
      html <- deleteHTMLtags(html)
      html <- strsplit(html, split=split)[[1]]
      return(html)
    } else {
      stop("別の検索語を再入力してください。", call.=debugCall)
    }
  }
  # データ(コード, 会社名)
  parseCodeNameData <- function(html)
  {
    split <- ", "
    codeColNames <- parseCodeColNames(html)
    html <- grep("<tr><td><a href=\\\"/q\\\"?s=\\\"", html, value=TRUE)
    if (length(html)) {
      html <- gsub("<td>(<[>]+)*([<]+)(<[>]+)*</td>", paste("\\2", split, sep=""), html)
      html <- deleteHTMLtags(html)
      html <- strsplit(html, split=split)
      html <- t(as.data.frame(html))
    }
  }
}

```

```
html <- apply(html, c(1,2), as.character)
row.names(html) <- NULL
colnames(html) <- codeColNames
return(html)
} else {
  stop("別の検索語を再入力してください.", call.=debugCall)
}
}
# 次のページがあるか
parseNextPage <- function(html)
{
  matchChar <- "^<a href=\"([^\"]+)\" class=\"yjM\">次の[[:digit:]]{1,2}件</a>$"
  html <- grep(matchChar, html, value=TRUE)
  if (length(html)) {
    html <- gsub(matchChar, "\\1", html)
    return(html)
  } else {
    return(FALSE)
  }
}
nextPage <- 0
codeNameData <- NULL
while (TRUE) {
  url <- yjc.url(scn, query=nextPage)
  html <- fetchHTML(url, fromEncoding, toEncoding)
  codeNameData <- rbind(codeNameData, parseCodeNameData(html))
  nextPage <- parseNextPage(html)
  if (nextPage==FALSE) {
    break
  }
  Sys.sleep(kindness) # 休止時間(秒)/1ページ
}
return(codeNameData)
}
```