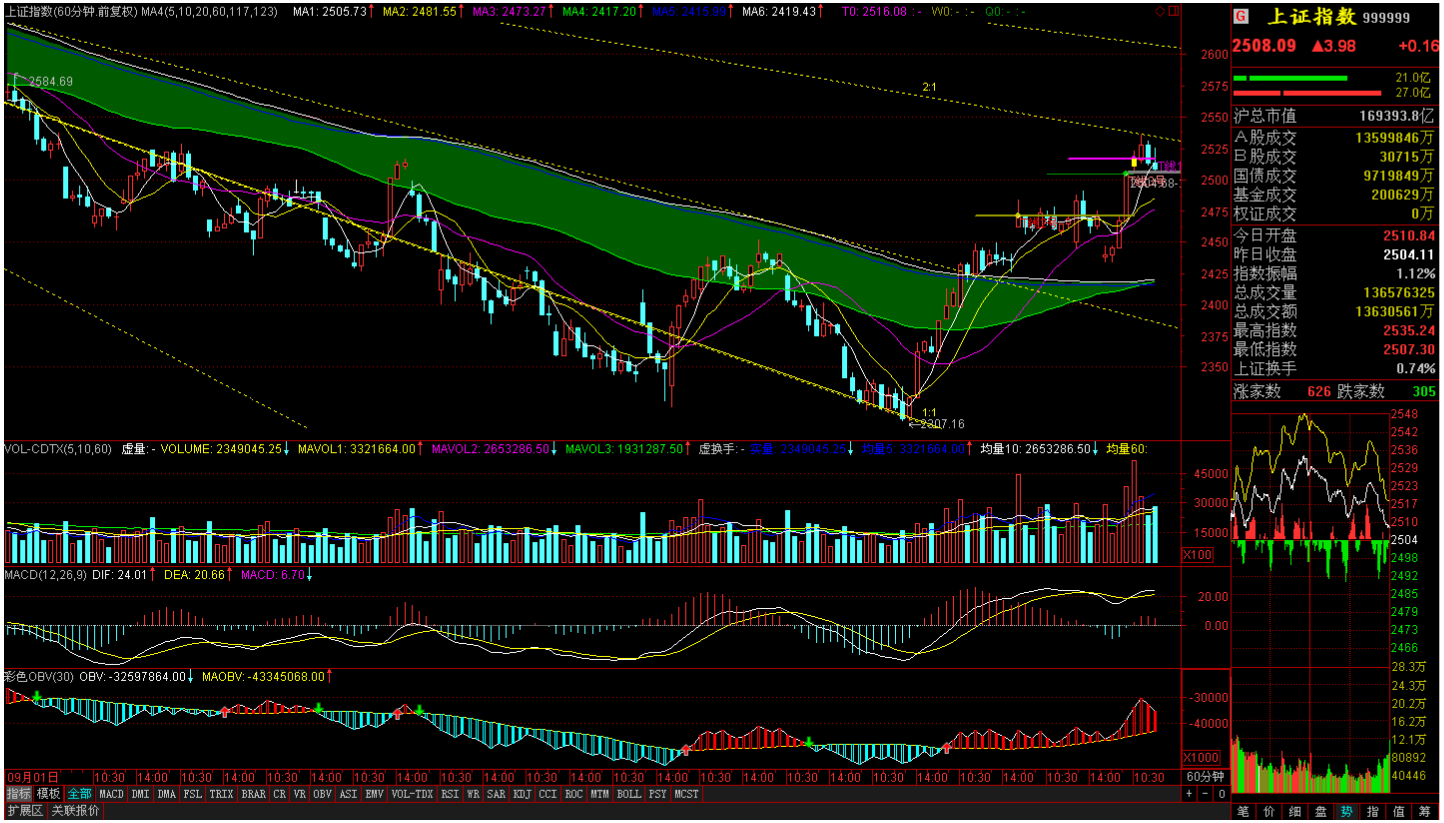
ATAGURU 炼数成金


上证指数(60分钟.前复权) MA4(5,10,20,60,117,123) MA1: 2505.73↑ MA2: 2481.55↑ MA3: 2473.27↑ MA4: 2417.20↑ MA5: 2415.99↑ MA6: 2419.43↑ T0: 2516.08 :- W0: - :- Q0: - :-
2584.69
2:1
1:1
2307.16
2600
2575
2550
2525
2500
2475
2450
2425
2400
2375
2350
VOL-CDTX(5,10,60) 虚量: - VOLUME: 2349045.25↓ MAVOL1: 3321664.00↑ MAVOL2: 2653286.50↓ MAVOL3: 1931287.50↑ 虚换手: - 实量: 2349045.25↓ 均量5: 3321664.00↑ 均量10: 2653286.50↓ 均量60:
45000
30000
15000
X100
MACD(12,26,9) DIF: 24.01↑ DEA: 20.66↑ MACD: 6.70↓
20.00
0.00
彩色OBV(30) OBV: -32597864.00↓ MAOBV: -43345068.00↑
-30000
-40000
X1000
09月01日
10:30
14:00
60分钟
指标 模板 全部 MACD DMI DMA FSL TRIX BRAR CR VR OBV ASI EMV VOL-TDX RSI WR SAR KDJ CCI ROC MTM BOLL PSY MCST
扩展区 关联报价
2548
2542
2536
2529
2523
2517
2510
2504
2498
2492
2485
2479
2473
2466
28.3万
24.3万
20.2万
16.2万
12.1万
80892
40446
笔 价 细 盘 势 指 值 筹

G 上证指数 999999

2508.09 ▲3.98 +0.16

21.0亿
27.0亿

沪总市值	169393.8亿
A股成交	13599846万
B股成交	30715万
国债成交	9719849万
基金成交	200629万
权证成交	0万
今日开盘	2510.84
昨日收盘	2504.11
指数振幅	1.12%
总成交量	136576325
总成交额	13630561万
最高指数	2535.24
最低指数	2507.30
上证换手	0.74%

涨家数 626 跌家数 305

R金融数据分析之quantmod包 第1周

DATAGURU专业数据分析社区

Mahout机器学习平台　讲师 黄志洪

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

http://edu.dataguru.cn

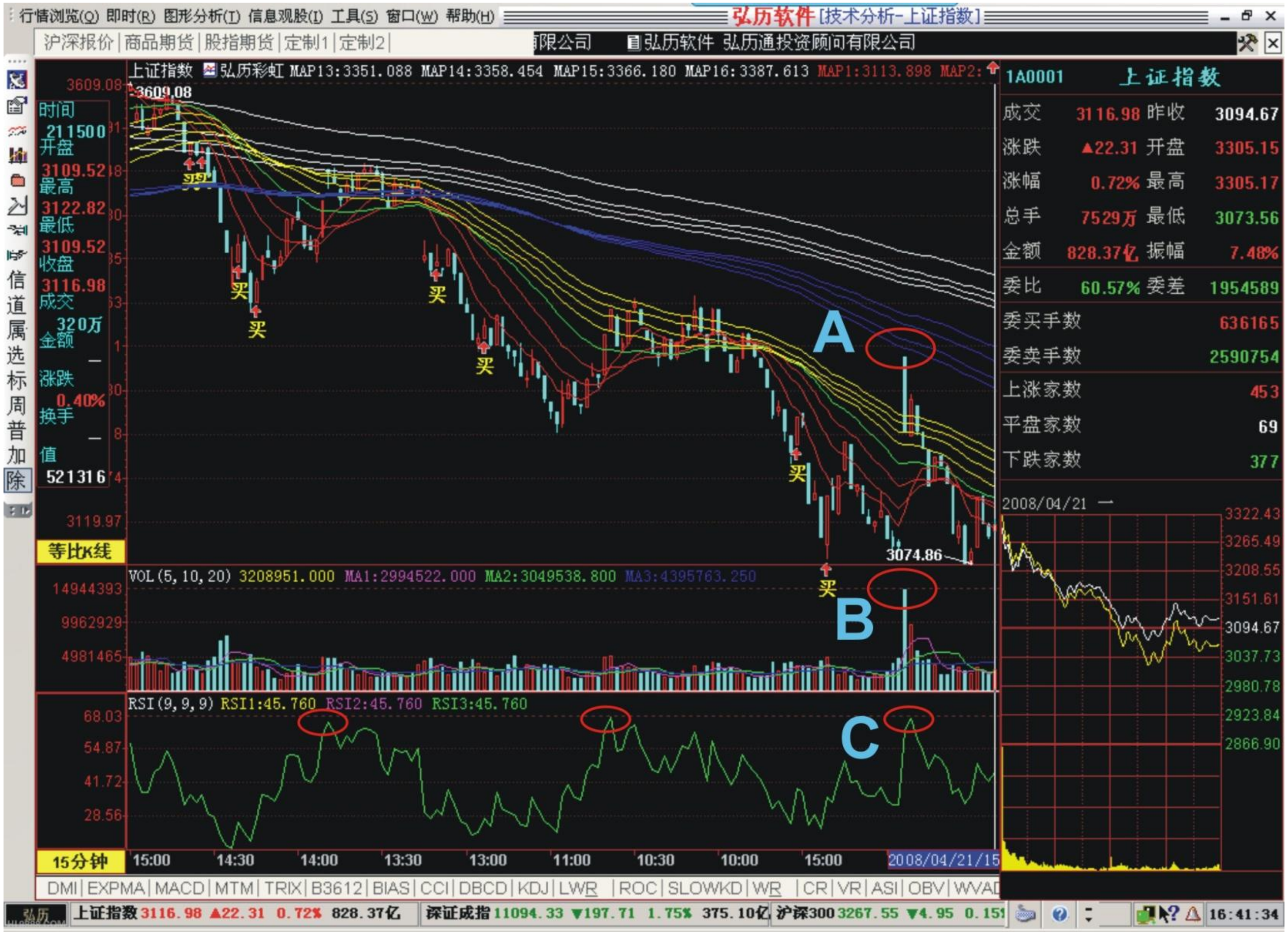
关注炼数成金企业微信

- 提供全面的数据价值资讯，涵盖商业智能与数据分析、大数据、企业信息化、数字化技术等，各种高性价比课程信息，赶紧掏出您的手机关注吧！

金融行情数据分析

- 从传统的股市交易图表说起
- 量化投资
- 统计套利
- 算法交易
- 高频交易

K线图



行情浏览(Q) 即时(R) 图形分析(T) 信息观股(I) 工具(S) 窗口(W) 帮助(H) 弘历软件 [技术分析-上证指数]

沪深报价 | 商品期货 | 股指期货 | 定制1 | 定制2 | 限公司 弘历软件 弘历通投资顾问有限公司

上证指数 弘历彩虹 MAP13:3351.088 MAP14:3358.454 MAP15:3366.180 MAP16:3387.613 MAP1:3113.898 MAP2:

1A0001	上证指数		
成交	3116.98	昨收	3094.67
涨跌	▲22.31	开盘	3305.15
涨幅	0.72%	最高	3305.17
总手	7529万	最低	3073.56
金额	828.37亿	振幅	7.48%
委比	60.57%	委差	1954589
委买手数			636165
委卖手数			2590754
上涨家数			453
平盘家数			69
下跌家数			377

时间	211500
开盘	3109.52
最高	3122.82
最低	3109.52
收盘	3116.98
成交	320万
金额	—
涨跌	0.40%
换手	—
值	521316

VOL(5,10,20) 3208951.000 MA1:2994522.000 MA2:3049538.800 MA3:4395763.250

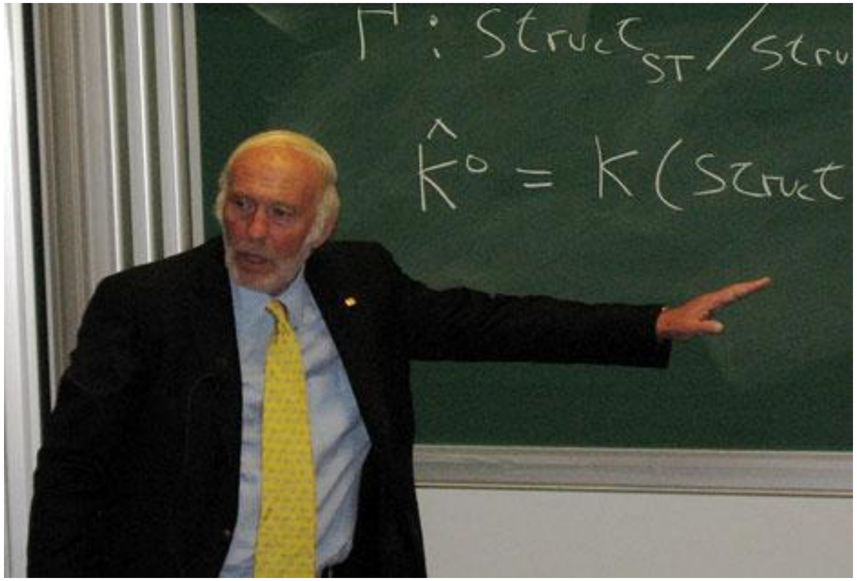
RSI(9,9,9) RSI1:45.760 RSI2:45.760 RSI3:45.760

DMI | EXPMA | MACD | MTM | TRIX | B3612 | BIAS | CCI | DBCD | KDJ | LWR | ROC | SLOWKD | WR | CR | VR | ASI | OBV | WVAD

上证指数 3116.98 ▲22.31 0.72% 828.37亿 深证成指 11094.33 ▼197.71 1.75% 375.10亿 沪深300 3267.55 ▼4.95 0.15 16:41:34

量化投资

- 什么是**量化投资**？
- 量化投资区别于传统操盘的特点
- 西蒙斯的大奖章基金
- 量化投资就是画图看图吗？
- 在中国这样的政策市，量化有效吗？



统计套利

- 统计套利是一种基于模型的套利策略，它从资产的历史交易数据找寻规律，发现两个或者两个以上的资产之间存在的套利机会，然后通过模型拟合资产价格的变化规律，设定交易阀值，通过计算机程序根据市场的实时信息自动发出交易信号而进行套利。
- 成对交易，即价差交易，是统计套利最常用的策略，指在构建某一资产多头的同时，构建另一种资产的空头，并在将来某一时刻同时了结两资产的头寸。这是一种市场中性策略，可以免疫市场风险，通过捕捉两个或者多个资产之间的相对错误定价机会来获得低风险收益。
- 主成分分析法，该策略通过分析与股票收益率相关的多种因素，建立回归模型，通过分析资产实际价格和模型预测价格之间的差异来获利。当实际资产价格高于模型预测价格时，则说明该资产被高估了，卖出该资产，待到实际资产价格与模型预测价格相等时，再买入该资产以平掉之前的空头头寸。反之则进行相反操作。

算法交易

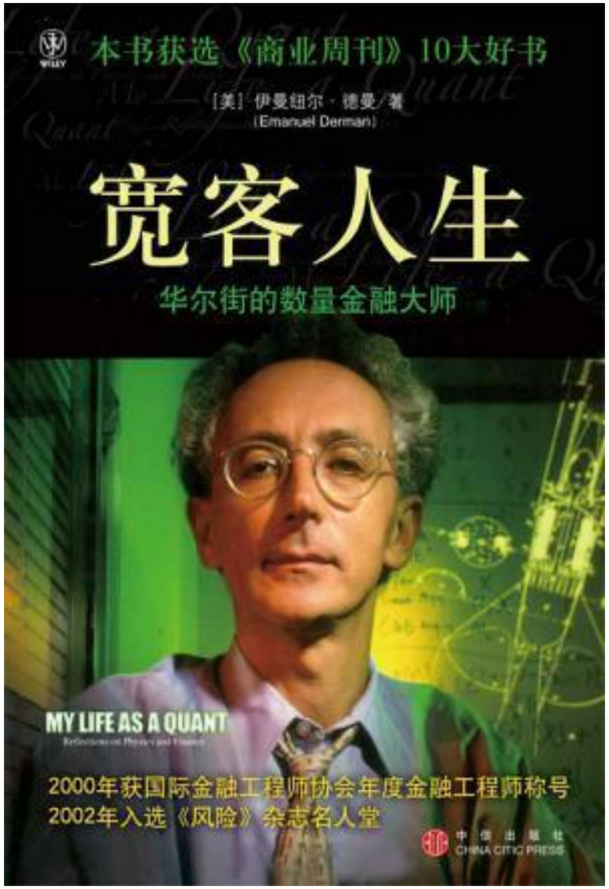
- **算法交易**又称自动交易、黑盒交易或者机器交易，它指的是通过使用计算机程序来发出交易指令的方法。在交易中，程序可以决定的范围包括交易时间的选择、交易的价格，甚至包括最后需要成交的证券数量。
- **被动型算法交易**除利用历史数据估计交易模型的关键参数外，不会根据市场的状况主动选择交易的时机与交易的数量，而是按照一个既定的交易方针进行交易。该策略的核心是减少滑价（目标价与实际成交均价的差）。被动型算法交易最成熟，使用也最为广泛，如在国际市场上使用最多的成交量加权平均价格（VWAP）、时间加权平均价格（TWAP）等都属于被动型算法交易。
- **主动型算法交易**也叫机会型算法交易。这类交易算法根据市场的状况做出实时的决策，判断是否交易、交易的数量、交易的价格等。

高频交易

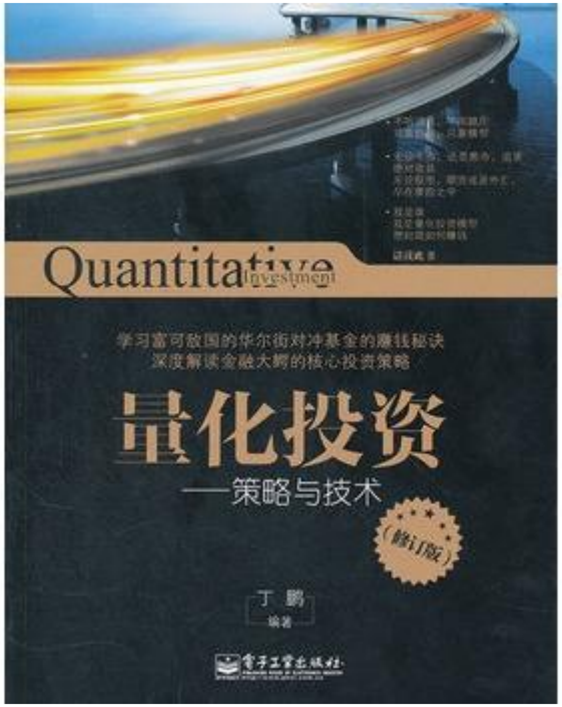
- 高频交易是指从那些人们无法利用的极为短暂的市场变化中寻求获利的计算机化交易，比如，某种证券买入价和卖出价差价的微小变化，或者某只股票在不同交易所之间的微小价差。这种交易的速度如此之快，以至于有些交易机构将自己的“服务器群组”（server farms）安置到了离交易所的计算机很近的地方，以缩短交易指令通过光缆以光速旅行的距离。

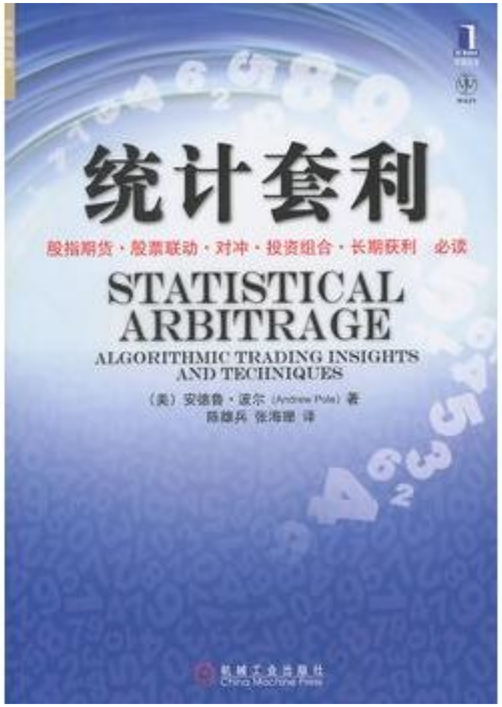
Quant

- Quant是谁？
- 有哪些Quant？
- 怎样才能成为Quant？

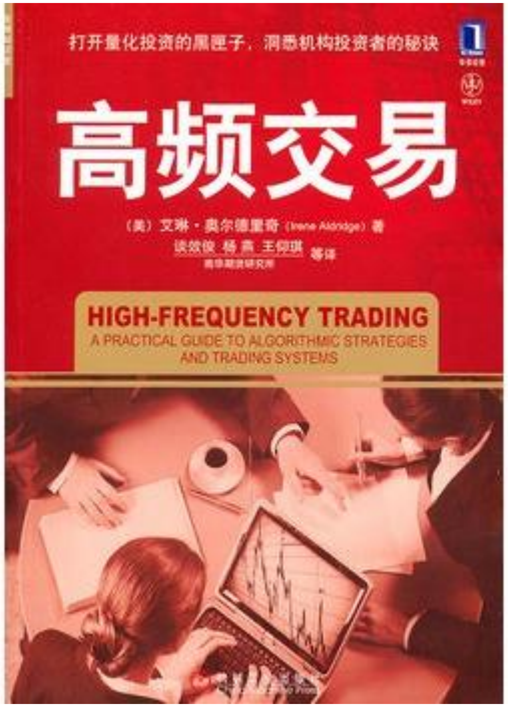

本书获选《商业周刊》10大好书
[美] 伊曼纽尔·德曼/著
(Emanuel Derman)
宽客人生
华尔街的数量金融大师
MY LIFE AS A QUANT
Reflections on Physics and Finance
2000年获国际金融工程师协会年度金融工程师称号
2002年入选《风险》杂志名人堂
中信出版社
CHINA CITIC PRESS

参考书


Quantitative
Investment
学习富可敌国的华尔街对冲基金的赚钱秘诀
深度解读金融大鳄的核心投资策略
量化投资
——策略与技术
(修订版)
丁鹏
编著
电子工业出版社


统计套利
股指期货·股票联动·对冲·投资组合·长期获利 必读
STATISTICAL
ARBITRAGE
ALGORITHMIC TRADING INSIGHTS
AND TECHNIQUES
(美) 安德鲁·波尔 (Andrew Pole) 著
陈雄兵 张海珊 译
机械工业出版社
China Machine Press

参考书



打开量化投资的黑匣子，洞悉机构投资者的秘诀

高频交易

（美）艾琳·奥尔德里奇（Irene Aldridge）著

谈效俊 杨燕 王仰琪 等译

南华期货研究所

HIGH-FREQUENCY TRADING

A PRACTICAL GUIDE TO ALGORITHMIC STRATEGIES AND TRADING SYSTEMS

南開大學 金融学本科教材系列

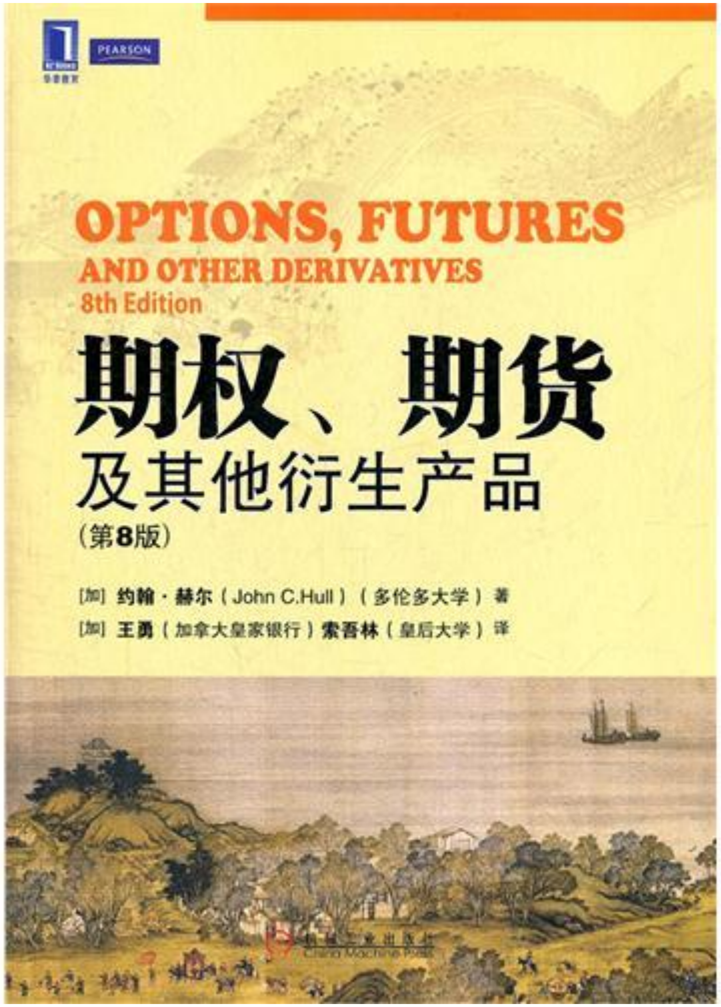
算法交易与套利交易

赵胜民等 编著

厦门大学出版社

XIAMEN UNIVERSITY PRESS

参考书



PEARSON

OPTIONS, FUTURES
AND OTHER DERIVATIVES
8th Edition

期权、期货
及其他衍生产品
（第8版）

[加] 约翰·赫尔（John C.Hull）（多伦多大学）著
[加] 王勇（加拿大皇家银行）索吾林（皇后大学）译

R

- R的源起

R是S语言的一种实现。S语言是由 AT&T 贝尔实验室开发的一种用来进行数据探索、统计分析、作图的解释型语言。最初S语言的实现版本主要是S-PLUS。S-PLUS是一个商业 软件，它基于S语言，并由MathSoft公司的统计科学部进一步完善。后来Auckland大学的Robert Gentleman 和 Ross Ihaka 及其他志愿人员开发了一个R系统。R的使用与S-PLUS有很多类似之处，两个软件有一定的兼容性。

R

- R is free

R是用于统计分析、绘图的语言和操作环境。R是属于GNU系统的一个自由、免费、源代码开放的软件，它是一个用于统计计算和统计制图的优秀工具。

R是一套完整的数据处理、计算和制图软件系统。其功能包括：数据存储和处理系统；数组运算工具（其向量、矩阵运算方面功能尤其强大）；完整连贯的统计分析工具；优秀的统计制图功能；简便而强大的编程语言：可操纵数据的输入和输入，可实现分支、循环，用户可自定义功能。

R是一个免费的自由软件，它有UNIX、LINUX、MacOS和WINDOWS版本，都是可以免费下载和使用的,在那儿可以下载到R的安装程序、各种外挂程序和文档。在R的安装程序中只包含了8个基础模块，其他外在模块可以通过CRAN获得。

R官方网站地址：http://www.r-project.org

R

- R的特点

1．有效的数据处理和保存机制。

2．拥有一整套数组和矩阵的操作运算符。

3．一系列连贯而又完整的数据分析中间工具。

4．图形统计可以对数据直接进行分析和显示，可用于多种图形设备。

5．一种相当完善、简洁和高效的程序设计语言。它包括条件语句、循环语句、用户自定义的递归函数以及输入输出接口。

6．R语言是彻底面向对象的统计编程语言。

7．R语言和其它编程语言、数据库之间有很好的接口。

8．R语言是自由软件，可以放心大胆地使用，但其功能却不比任何其它同类软件差。

9．R语言具有丰富的网上资源

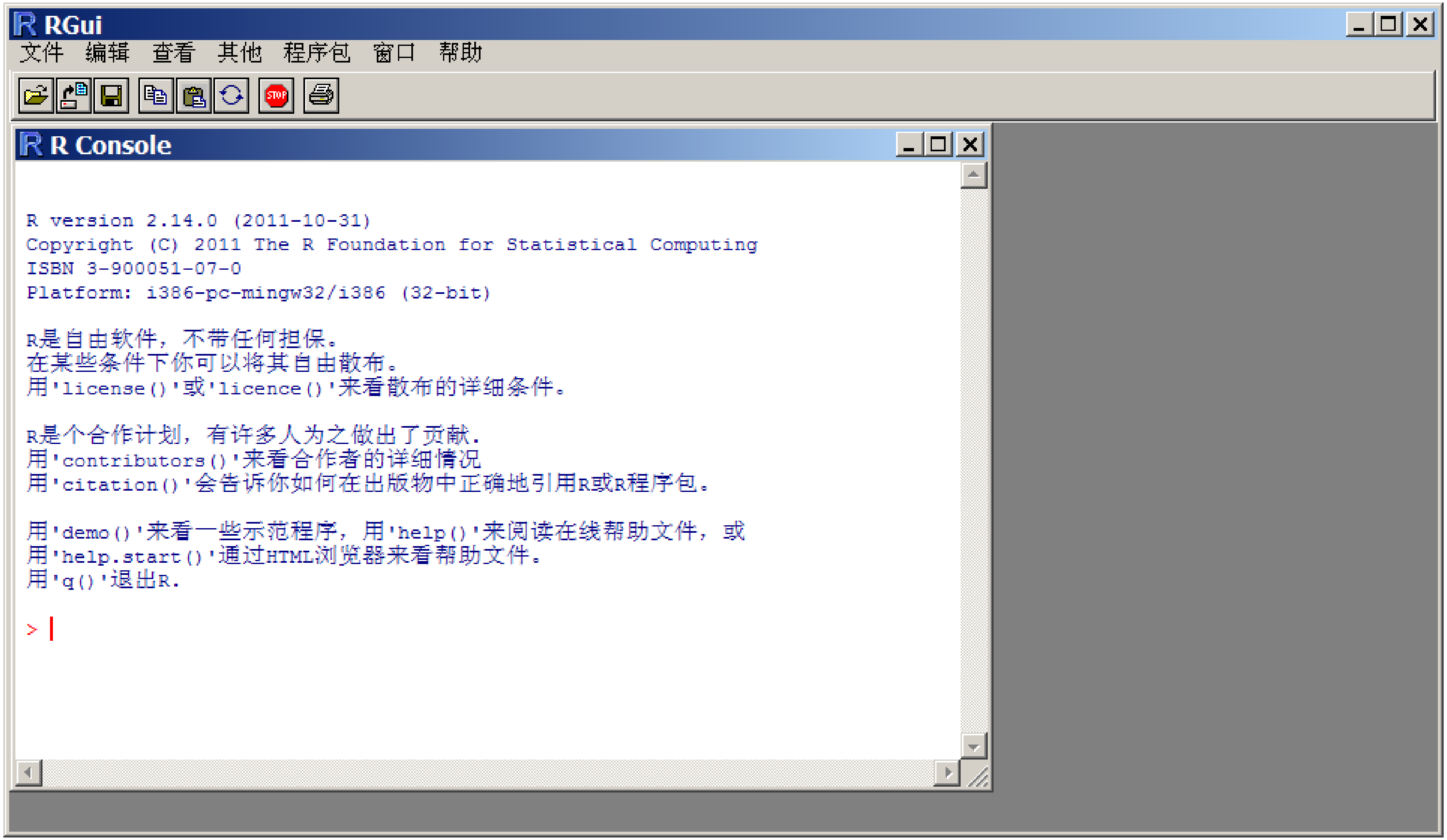
R

- 商业版本的R

Revolution R（官网：http://www.revolutionanalytics.com/）

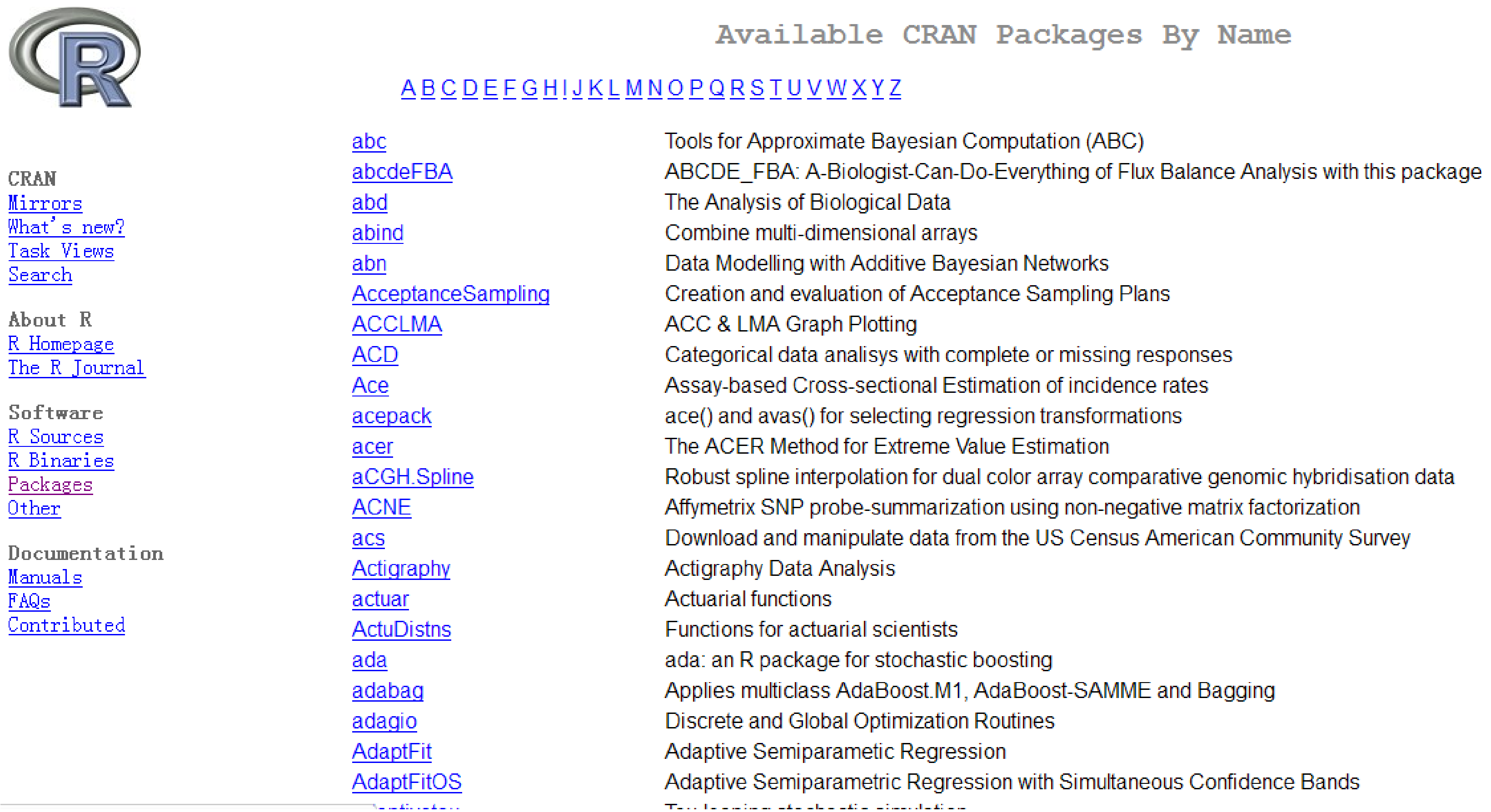
很多大型厂商也在开始推出自己的R或兼容R的产品，例如Oracle、IBM、Sybase

R语言



```
R version 2.14.0 (2011-10-31)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R是自由软件，不带任何担保。
在某些条件下你可以将其自由散布。
用'license()'或'licence()'来看散布的详细条件。

R是个合作计划，有许多人为之做出了贡献.
用'contributors()'来看合作者的详细情况
用'citation()'会告诉你如何在出版物中正确地引用R或R程序包。

用'demo()'来看一些示范程序，用'help()'来阅读在线帮助文件，或
用'help.start()'通过HTML浏览器来看帮助文件。
用'q()'退出R.

> 
```

R语言力量源泉——扩展包



CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

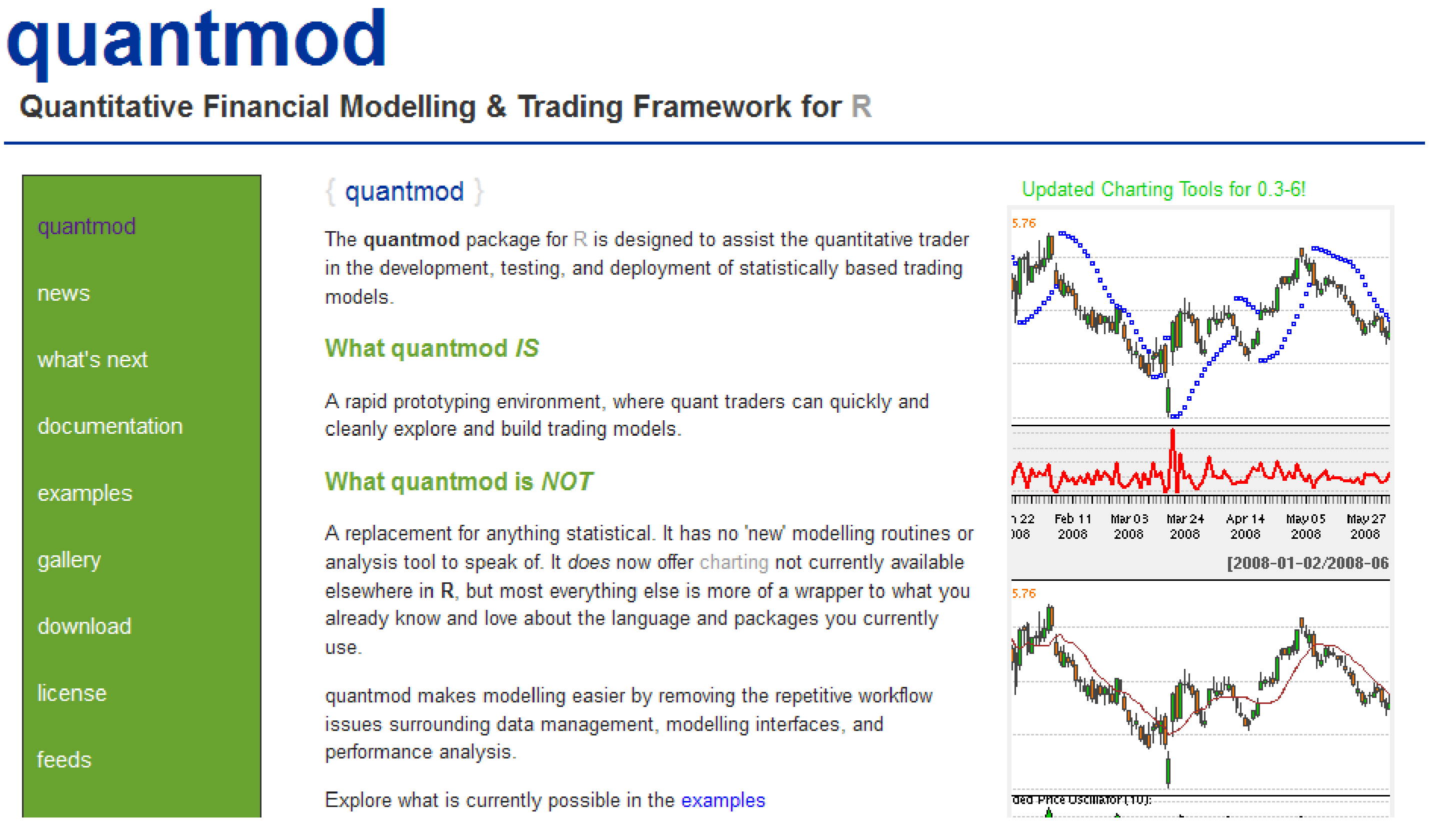
Available CRAN Packages By Name

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

abc	Tools for Approximate Bayesian Computation (ABC)
abcdeFBA	ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
abd	The Analysis of Biological Data
abind	Combine multi-dimensional arrays
abn	Data Modelling with Additive Bayesian Networks
AcceptanceSampling	Creation and evaluation of Acceptance Sampling Plans
ACCLMA	ACC & LMA Graph Plotting
ACD	Categorical data analisys with complete or missing responses
Ace	Assay-based Cross-sectional Estimation of incidence rates
acepack	ace() and avas() for selecting regression transformations
acer	The ACER Method for Extreme Value Estimation
aCGH.Spline	Robust spline interpolation for dual color array comparative genomic hybridisation data
ACNE	Affymetrix SNP probe-summarization using non-negative matrix factorization
acs	Download and manipulate data from the US Census American Community Survey
Actigraphy	Actigraphy Data Analysis
actuar	Actuarial functions
ActuDistns	Functions for actuarial scientists
ada	ada: an R package for stochastic boosting
adabag	Applies multiclass AdaBoost.M1, AdaBoost-SAMME and Bagging
adagio	Discrete and Global Optimization Routines
AdaptFit	Adaptive Semiparametric Regression
AdaptFitOS	Adaptive Semiparametric Regression with Simultaneous Confidence Bands
adaptivetau	Tau-leaping stochastic simulation

Quantmod扩展包

- http://www.quantmod.com/



quantmod

Quantitative Financial Modelling & Trading Framework for R

- quantmod
- news
- what's next
- documentation
- examples
- gallery
- download
- license
- feeds

{ quantmod }

The **quantmod** package for R is designed to assist the quantitative trader in the development, testing, and deployment of statistically based trading models.

What quantmod *IS*

A rapid prototyping environment, where quant traders can quickly and cleanly explore and build trading models.

What quantmod is *NOT*

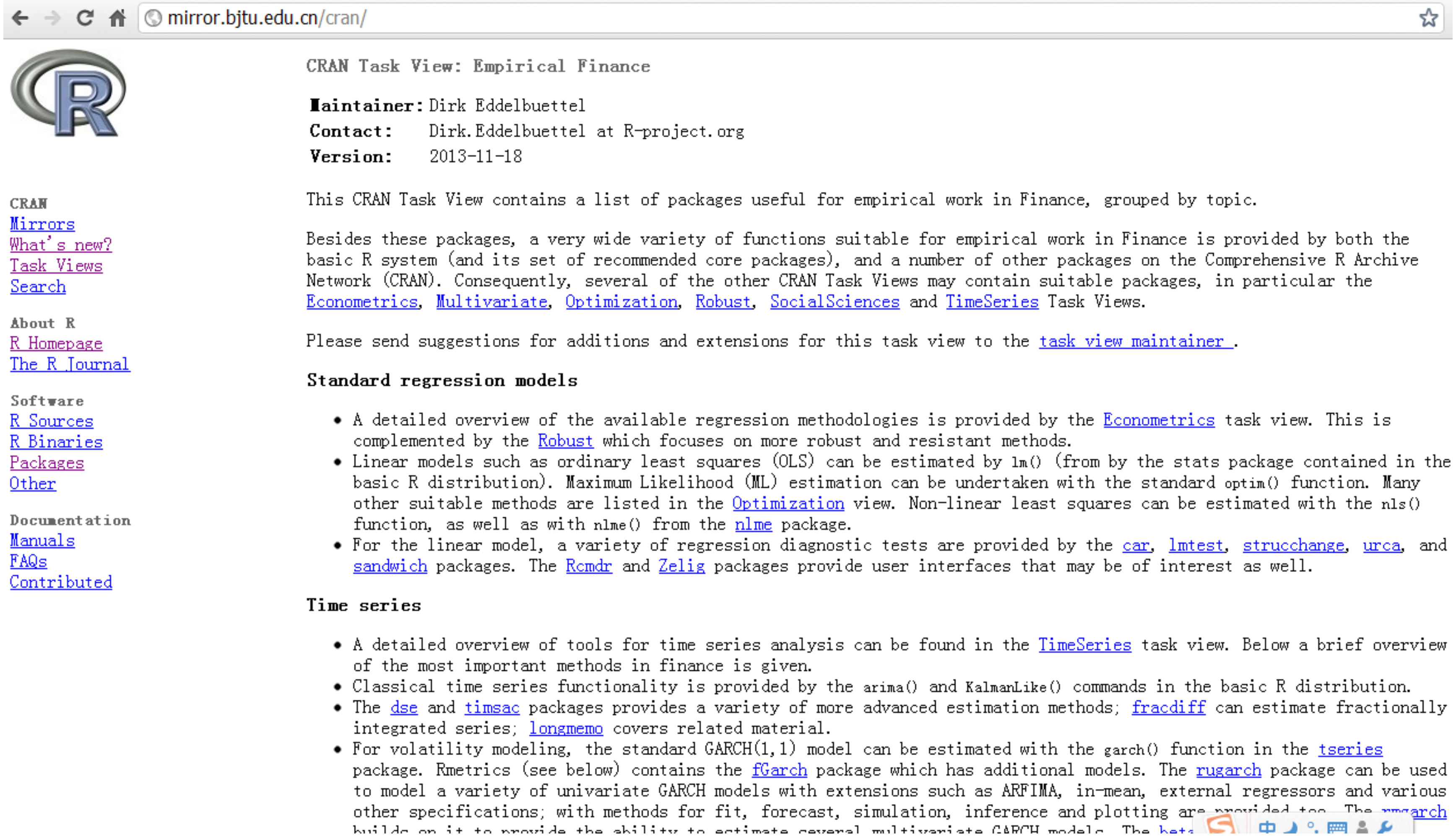
A replacement for anything statistical. It has no 'new' modelling routines or analysis tool to speak of. It *does* now offer charting not currently available elsewhere in **R**, but most everything else is more of a wrapper to what you already know and love about the language and packages you currently use.

quantmod makes modelling easier by removing the repetitive workflow issues surrounding data management, modelling interfaces, and performance analysis.

Explore what is currently possible in the examples

Updated Charting Tools for 0.3-6!

R语言中其它与金融数据分析有关的包



mirror.bjtu.edu.cn/cran/

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

CRAN Task View: Empirical Finance

Maintainer: Dirk Eddelbuettel
Contact: Dirk.Eddelbuettel at R-project.org
Version: 2013-11-18

This CRAN Task View contains a list of packages useful for empirical work in Finance, grouped by topic.

Besides these packages, a very wide variety of functions suitable for empirical work in Finance is provided by both the basic R system (and its set of recommended core packages), and a number of other packages on the Comprehensive R Archive Network (CRAN). Consequently, several of the other CRAN Task Views may contain suitable packages, in particular the Econometrics, Multivariate, Optimization, Robust, SocialSciences and TimeSeries Task Views.

Please send suggestions for additions and extensions for this task view to the task view maintainer .

Standard regression models

- A detailed overview of the available regression methodologies is provided by the Econometrics task view. This is complemented by the Robust which focuses on more robust and resistant methods.
- Linear models such as ordinary least squares (OLS) can be estimated by lm() (from by the stats package contained in the basic R distribution). Maximum Likelihood (ML) estimation can be undertaken with the standard optim() function. Many other suitable methods are listed in the Optimization view. Non-linear least squares can be estimated with the nls() function, as well as with nlme() from the nlme package.
- For the linear model, a variety of regression diagnostic tests are provided by the car, lmtest, strucchange, urca, and sandwich packages. The Rcmdr and Zelig packages provide user interfaces that may be of interest as well.

Time series

- A detailed overview of tools for time series analysis can be found in the TimeSeries task view. Below a brief overview of the most important methods in finance is given.
- Classical time series functionality is provided by the arima() and KalmanLike() commands in the basic R distribution.
- The dse and timsac packages provides a variety of more advanced estimation methods; fracdiff can estimate fractionally integrated series; longmemo covers related material.
- For volatility modeling, the standard GARCH(1,1) model can be estimated with the garch() function in the tseries package. Rmetrics (see below) contains the fGarch package which has additional models. The rugarch package can be used to model a variety of univariate GARCH models with extensions such as ARFIMA, in-mean, external regressors and various other specifications; with methods for fit, forecast, simulation, inference and plotting are provided too. The rmgarch builds on it to provide the ability to estimate several multivariate GARCH models. The beta

# 安装quantmod包

- 先安装并加载Defaults，xts,zoo,TTR包

install.packages("zoo")

require(“zoo”)

…

- 安装完上述的4个包后安装quantmod包

#安装quantmod包

install.packages(“quantmod")

require(“quantmod”)

Quantmod包中的函数

- ETL类函数
- 分析类函数
- 展现类函数

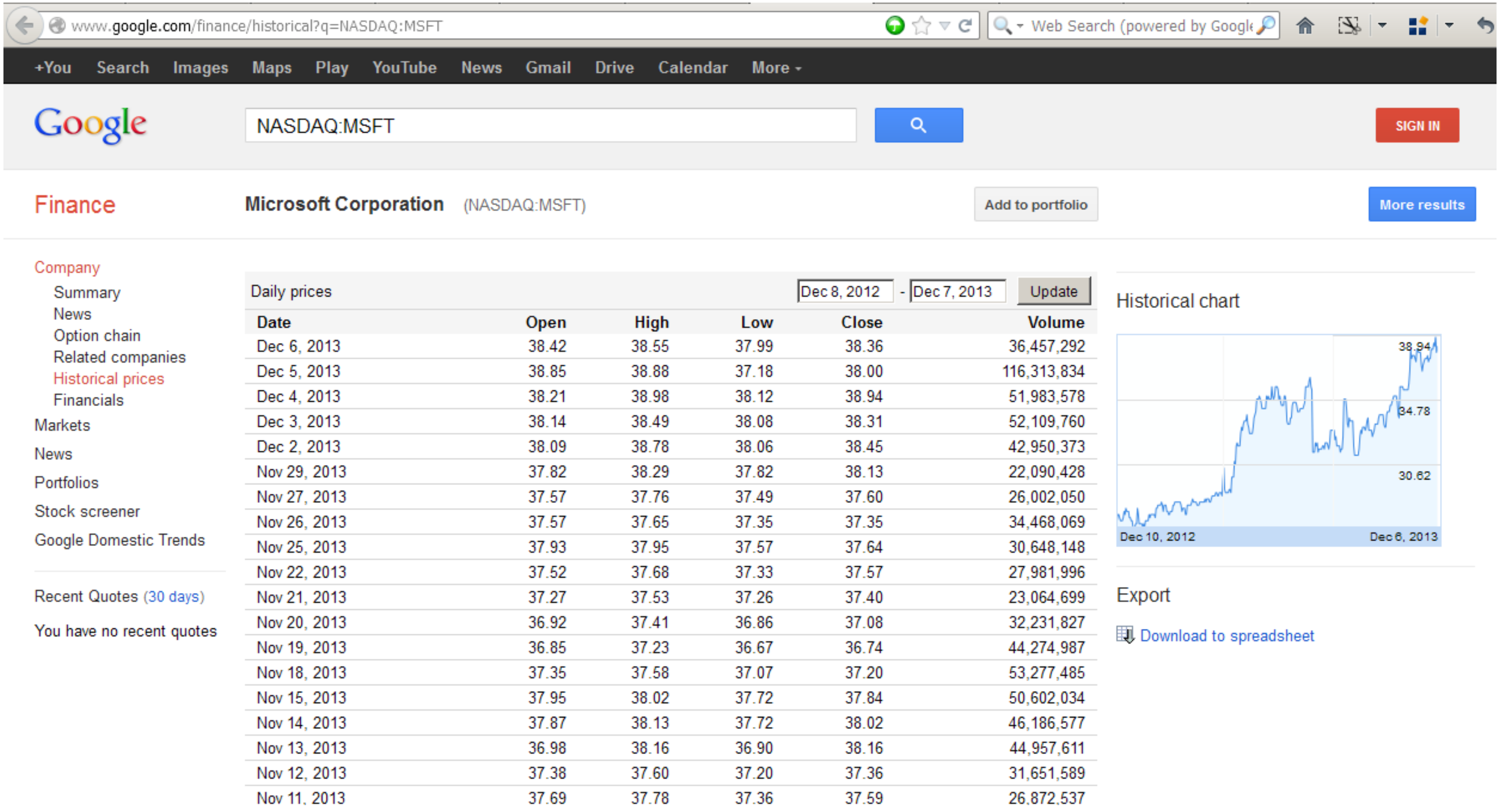
常用的数据类型

数据类型	全称	意义
Op	Open price	开盘价
Hi	High price	最高价
Lo	Low price	最低价
Cl	Close price	收盘价
Vo	Volume	交易量
Ad	Adjusted price	调整价格
HLC	High price，Low price，Close price	最高价、最低价和收盘价
OHLC	Open price，High price，Low price，Close price	开盘价、最高价、最低价和收盘价

ETL类函数

函数	作用	函数	作用
getSymbols()	从多种信息源里获得信息	getSymbols.csv()	从csv文件中读入数据
getDividends()	获取上市公司的股息数据	getSymbols.FRED()	从FRED中获取数据
getFinancials()	获取上市公司的财务报表	getSymbols.google()	从google中获取数据
getFX()	获取汇率数据	getSymbols.MySQL()	从MySQL中获取数据
getMetals()	获取重金属交易数据	getSymbols.oanda()	从oanda中获取数据
getSplits	获取上市公司的拆股数据	getSymbols.rda()	从R的二进制文件中获取数据
getOptionChain()	获取期权交易数据	getSymbols.SQLite()	从SQLite数据库中获取数据
getQuote	获取即时的网络报价	getSymbols.yahoo()	从雅虎网中获取数据

数据源：Google Finance historical data

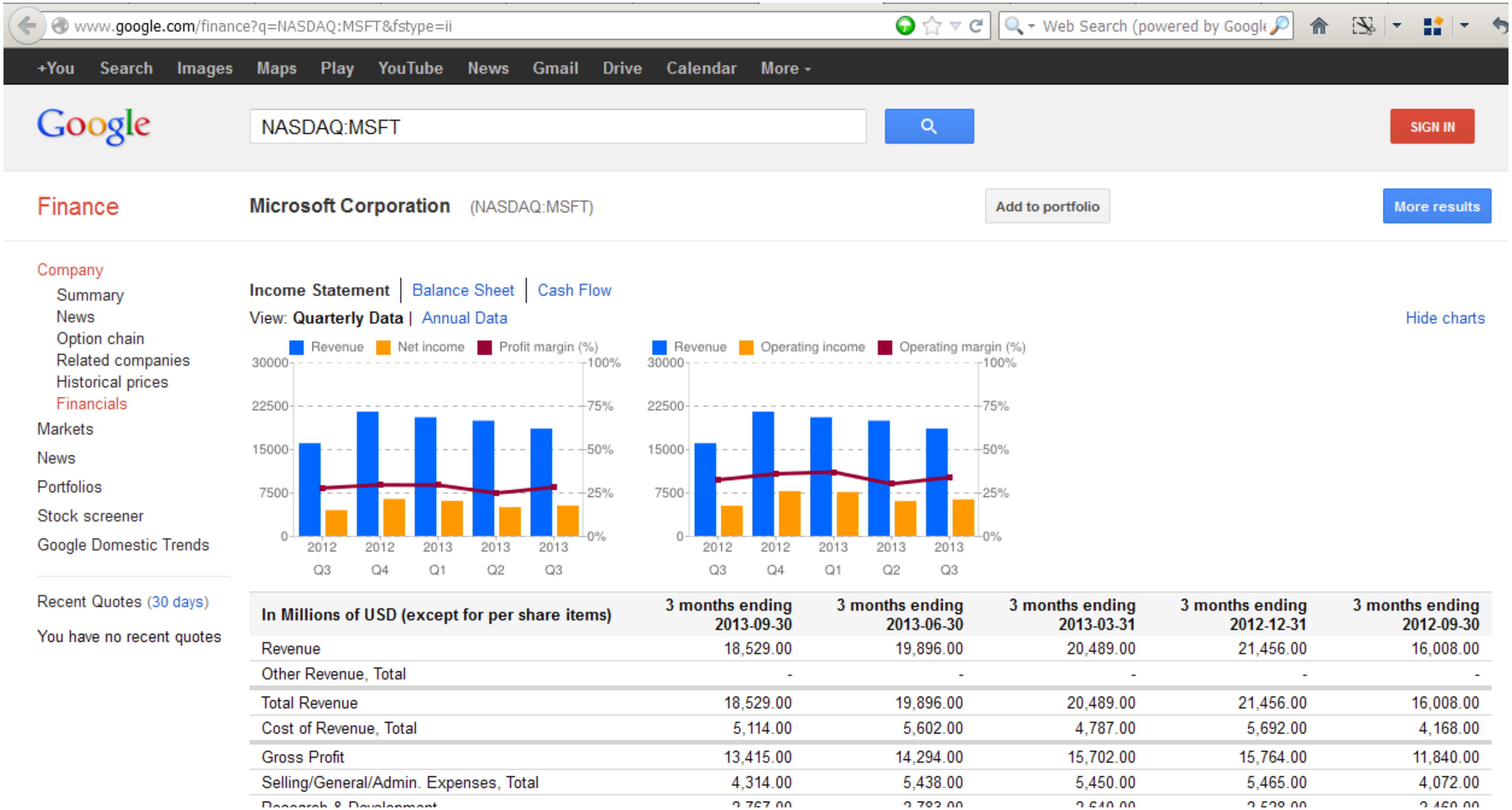


Microsoft Corporation (NASDAQ:MSFT)

Daily prices Dec 8, 2012 - Dec 7, 2013

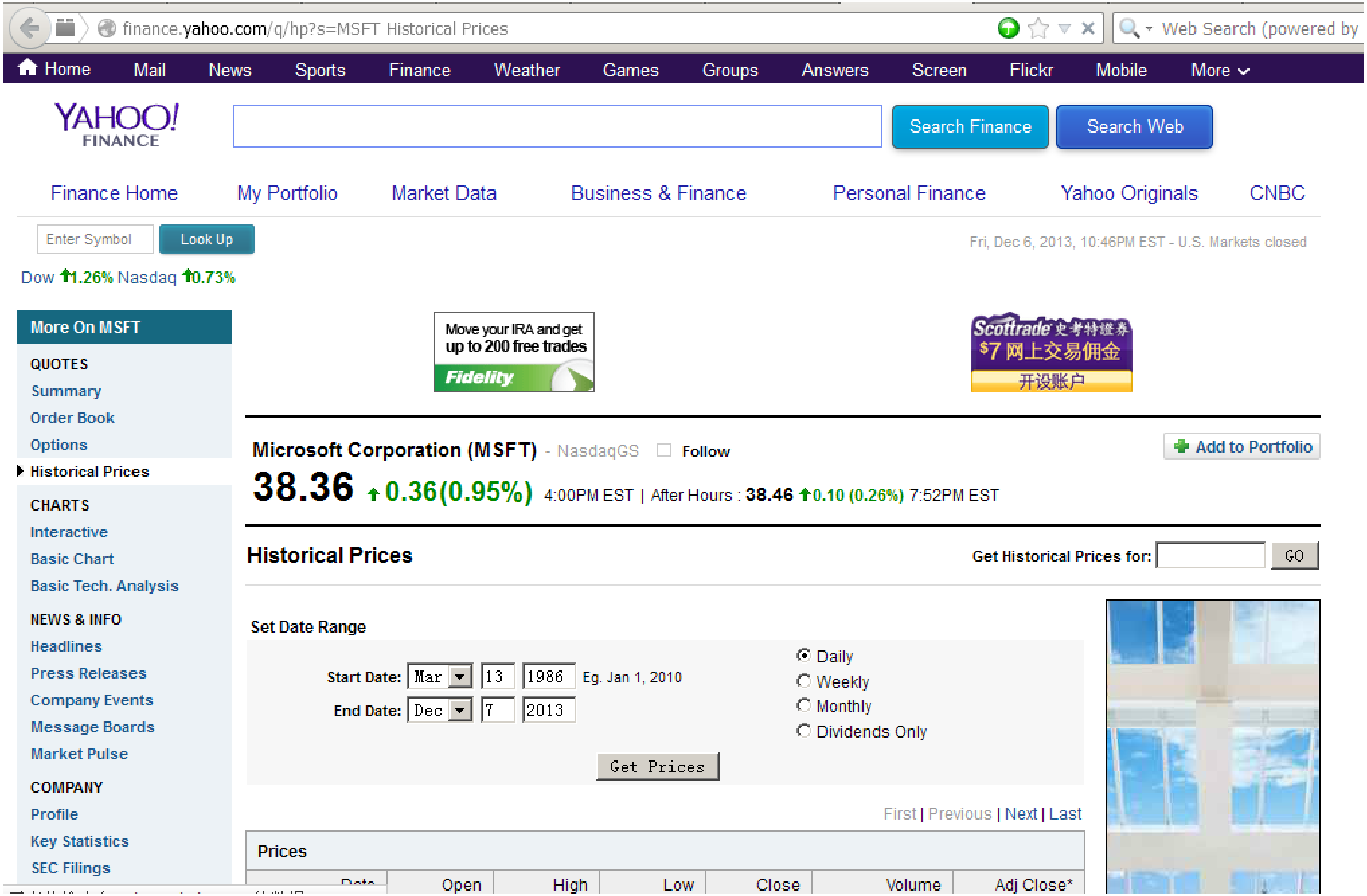
Date	Open	High	Low	Close	Volume
Dec 6, 2013	38.42	38.55	37.99	38.36	36,457,292
Dec 5, 2013	38.85	38.88	37.18	38.00	116,313,834
Dec 4, 2013	38.21	38.98	38.12	38.94	51,983,578
Dec 3, 2013	38.14	38.49	38.08	38.31	52,109,760
Dec 2, 2013	38.09	38.78	38.06	38.45	42,950,373
Nov 29, 2013	37.82	38.29	37.82	38.13	22,090,428
Nov 27, 2013	37.57	37.76	37.49	37.60	26,002,050
Nov 26, 2013	37.57	37.65	37.35	37.35	34,468,069
Nov 25, 2013	37.93	37.95	37.57	37.64	30,648,148
Nov 22, 2013	37.52	37.68	37.33	37.57	27,981,996
Nov 21, 2013	37.27	37.53	37.26	37.40	23,064,699
Nov 20, 2013	36.92	37.41	36.86	37.08	32,231,827
Nov 19, 2013	36.85	37.23	36.67	36.74	44,274,987
Nov 18, 2013	37.35	37.58	37.07	37.20	53,277,485
Nov 15, 2013	37.95	38.02	37.72	37.84	50,602,034
Nov 14, 2013	37.87	38.13	37.72	38.02	46,186,577
Nov 13, 2013	36.98	38.16	36.90	38.16	44,957,611
Nov 12, 2013	37.38	37.60	37.20	37.36	31,651,589
Nov 11, 2013	37.69	37.78	37.36	37.59	26,872,537

数据源：Google Finance balance sheets



In Millions of USD (except for per share items)	3 months ending 2013-09-30	3 months ending 2013-06-30	3 months ending 2013-03-31	3 months ending 2012-12-31	3 months ending 2012-09-30
Revenue	18,529.00	19,896.00	20,489.00	21,456.00	16,008.00
Other Revenue, Total	-	-	-	-	-
Total Revenue	18,529.00	19,896.00	20,489.00	21,456.00	16,008.00
Cost of Revenue, Total	5,114.00	5,602.00	4,787.00	5,692.00	4,168.00
Gross Profit	13,415.00	14,294.00	15,702.00	15,764.00	11,840.00
Selling/General/Admin. Expenses, Total	4,314.00	5,438.00	5,450.00	5,465.00	4,072.00
Research & Development	2,767.00	2,783.00	2,640.00	2,528.00	2,460.00

数据源：Yahoo Finance historical data



finance.yahoo.com/q/hp?s=MSFT Historical Prices

Fri, Dec 6, 2013, 10:46PM EST - U.S. Markets closed

Dow ↑1.26% Nasdaq ↑0.73%

Microsoft Corporation (MSFT) - NasdaqGS □ Follow

38.36 ↑0.36(0.95%) 4:00PM EST | After Hours : 38.46 ↑0.10 (0.26%) 7:52PM EST

Historical Prices

Get Historical Prices for: GO

Set Date Range

Start Date: Mar 13 1986 Eg. Jan 1, 2010

End Date: Dec 7 2013

● Daily
○ Weekly
○ Monthly
○ Dividends Only

Get Prices

First | Previous | Next | Last

Prices

Date	Open	High	Low	Close	Volume	Adj Close*

其它数据源

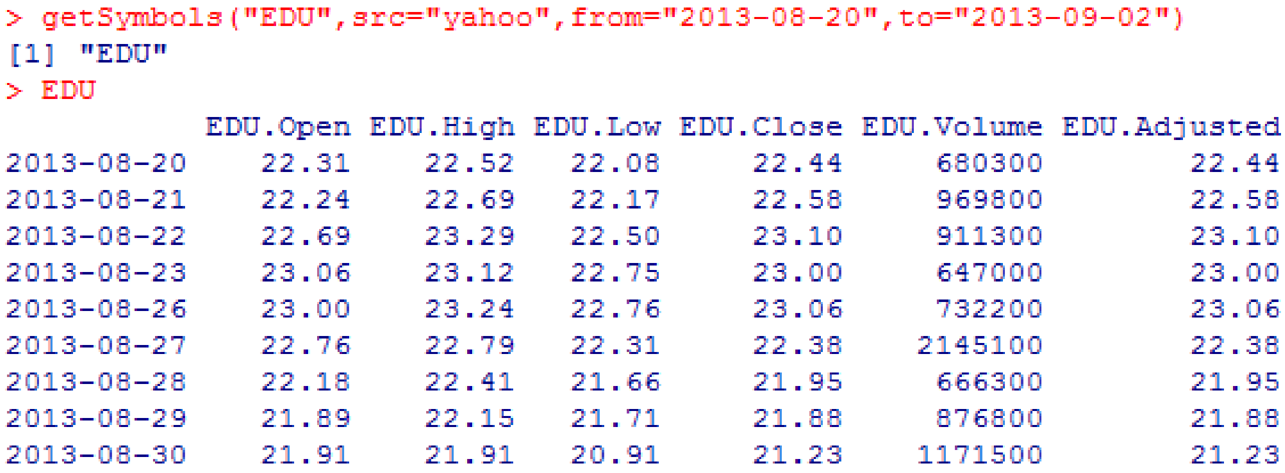
- http://www.oanda.com/
- http://research.stlouisfed.org/fred2/

其它工具

- RCurl
- RJSON
- RJSONIO
- XML
- Scraper
- WDI
- Tseries
- 参考： http://www.dataguru.cn/article-1466-1.html
- http://f.dataguru.cn/forum.php?mod=viewthread&tid=147799

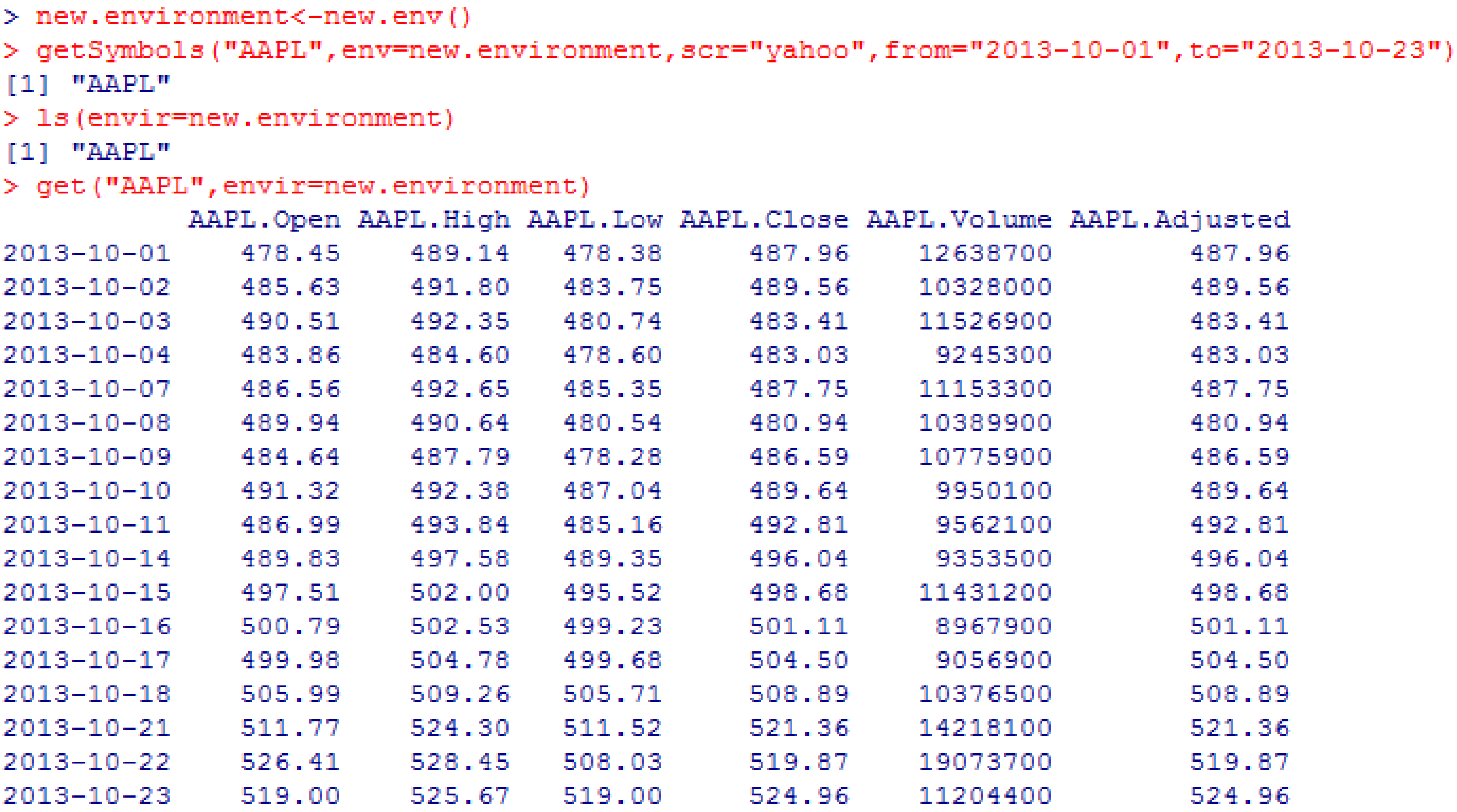
getSymbols()函数

- 获取上市公司股票的日交易数据



```
> getSymbols("EDU",src="yahoo",from="2013-08-20",to="2013-09-02")
[1] "EDU"
> EDU
           EDU.Open EDU.High EDU.Low EDU.Close EDU.Volume EDU.Adjusted
2013-08-20    22.31    22.52   22.08     22.44     680300        22.44
2013-08-21    22.24    22.69   22.17     22.58     969800        22.58
2013-08-22    22.69    23.29   22.50     23.10     911300        23.10
2013-08-23    23.06    23.12   22.75     23.00     647000        23.00
2013-08-26    23.00    23.24   22.76     23.06     732200        23.06
2013-08-27    22.76    22.79   22.31     22.38    2145100        22.38
2013-08-28    22.18    22.41   21.66     21.95     666300        21.95
2013-08-29    21.89    22.15   21.71     21.88     876800        21.88
2013-08-30    21.91    21.91   20.91     21.23    1171500        21.23
```

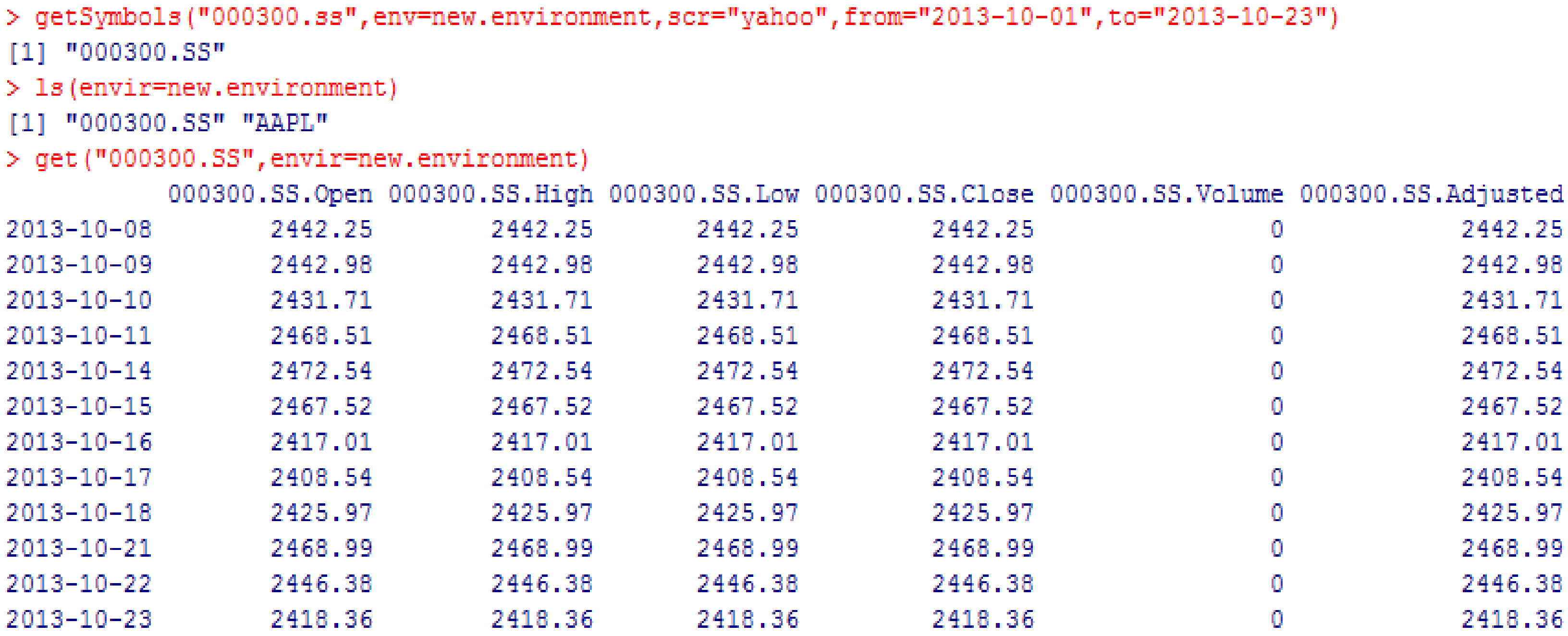
getSymbols()函数



```
> new.environment<-new.env()
> getSymbols("AAPL",env=new.environment,scr="yahoo",from="2013-10-01",to="2013-10-23")
[1] "AAPL"
> ls(envir=new.environment)
[1] "AAPL"
> get("AAPL",envir=new.environment)
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-10-01    478.45    489.14   478.38     487.96    12638700        487.96
2013-10-02    485.63    491.80   483.75     489.56    10328000        489.56
2013-10-03    490.51    492.35   480.74     483.41    11526900        483.41
2013-10-04    483.86    484.60   478.60     483.03     9245300        483.03
2013-10-07    486.56    492.65   485.35     487.75    11153300        487.75
2013-10-08    489.94    490.64   480.54     480.94    10389900        480.94
2013-10-09    484.64    487.79   478.28     486.59    10775900        486.59
2013-10-10    491.32    492.38   487.04     489.64     9950100        489.64
2013-10-11    486.99    493.84   485.16     492.81     9562100        492.81
2013-10-14    489.83    497.58   489.35     496.04     9353500        496.04
2013-10-15    497.51    502.00   495.52     498.68    11431200        498.68
2013-10-16    500.79    502.53   499.23     501.11     8967900        501.11
2013-10-17    499.98    504.78   499.68     504.50     9056900        504.50
2013-10-18    505.99    509.26   505.71     508.89    10376500        508.89
2013-10-21    511.77    524.30   511.52     521.36    14218100        521.36
2013-10-22    526.41    528.45   508.03     519.87    19073700        519.87
2013-10-23    519.00    525.67   519.00     524.96    11204400        524.96
```

getSymbols()函数

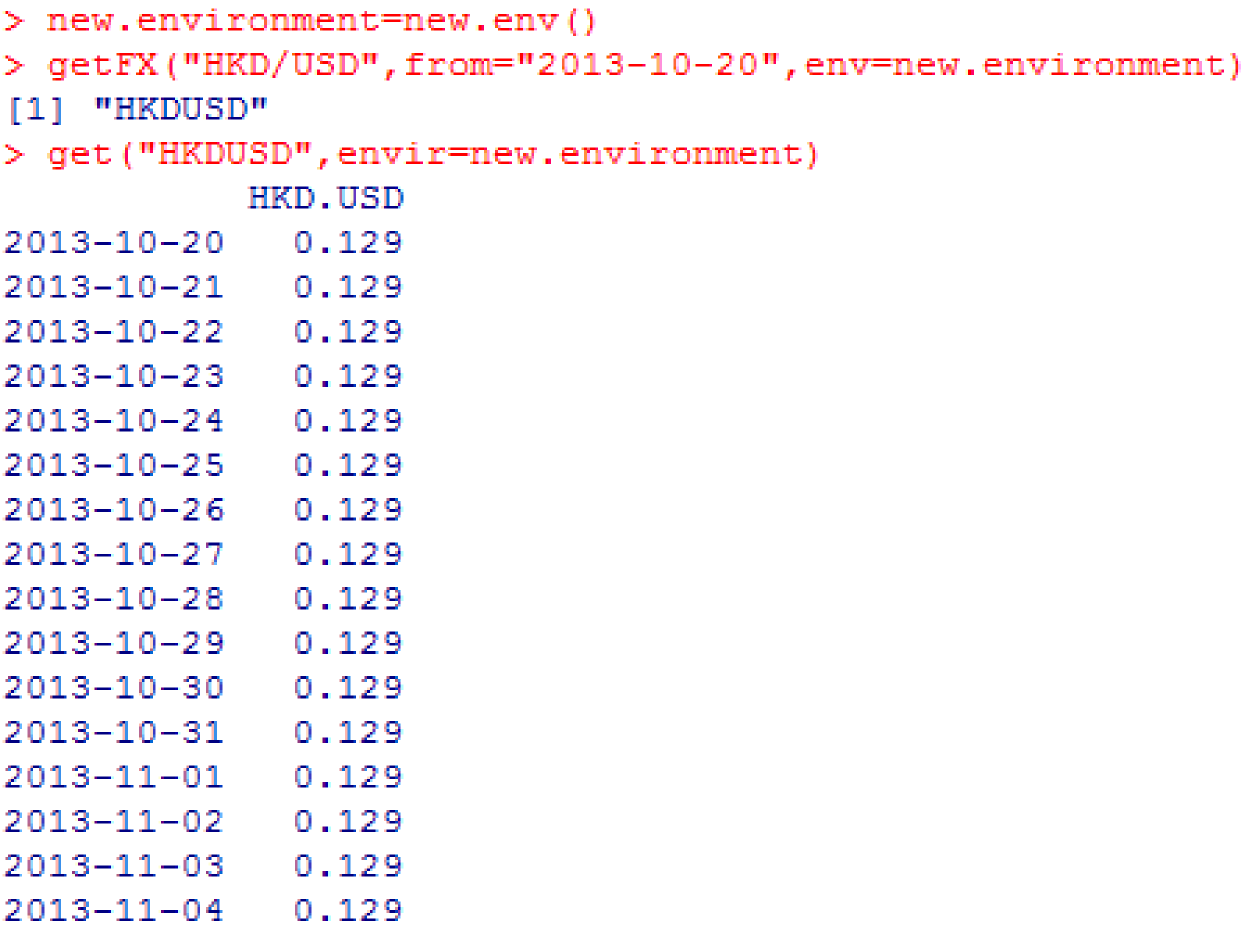
- 获取各种指数的日数据
- 例如我们想获取沪深300指数的日数据，它的代码为000300.ss，后面的字母表示该指数从属哪个交易所



```
> getSymbols("000300.ss",env=new.environment,scr="yahoo",from="2013-10-01",to="2013-10-23")
[1] "000300.SS"
> ls(envir=new.environment)
[1] "000300.SS" "AAPL"
> get("000300.SS",envir=new.environment)
           000300.SS.Open 000300.SS.High 000300.SS.Low 000300.SS.Close 000300.SS.Volume 000300.SS.Adjusted
2013-10-08        2442.25        2442.25       2442.25         2442.25                0            2442.25
2013-10-09        2442.98        2442.98       2442.98         2442.98                0            2442.98
2013-10-10        2431.71        2431.71       2431.71         2431.71                0            2431.71
2013-10-11        2468.51        2468.51       2468.51         2468.51                0            2468.51
2013-10-14        2472.54        2472.54       2472.54         2472.54                0            2472.54
2013-10-15        2467.52        2467.52       2467.52         2467.52                0            2467.52
2013-10-16        2417.01        2417.01       2417.01         2417.01                0            2417.01
2013-10-17        2408.54        2408.54       2408.54         2408.54                0            2408.54
2013-10-18        2425.97        2425.97       2425.97         2425.97                0            2425.97
2013-10-21        2468.99        2468.99       2468.99         2468.99                0            2468.99
2013-10-22        2446.38        2446.38       2446.38         2446.38                0            2446.38
2013-10-23        2418.36        2418.36       2418.36         2418.36                0            2418.36
```

getFX()函数

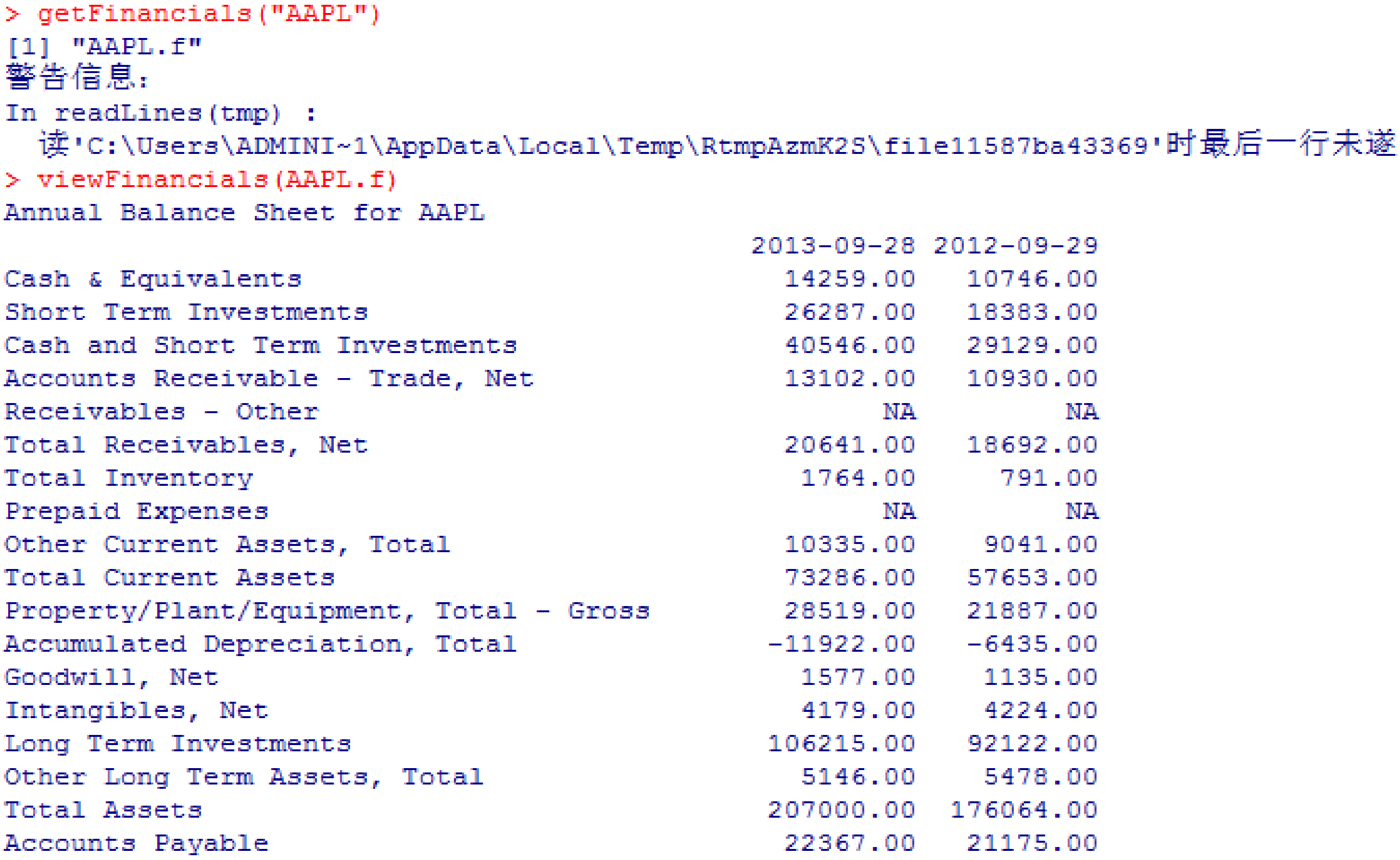
- 从oanda上获取汇率



```
> new.environment=new.env()
> getFX("HKD/USD",from="2013-10-20",env=new.environment)
[1] "HKDUSD"
> get("HKDUSD",envir=new.environment)
           HKD.USD
2013-10-20   0.129
2013-10-21   0.129
2013-10-22   0.129
2013-10-23   0.129
2013-10-24   0.129
2013-10-25   0.129
2013-10-26   0.129
2013-10-27   0.129
2013-10-28   0.129
2013-10-29   0.129
2013-10-30   0.129
2013-10-31   0.129
2013-11-01   0.129
2013-11-02   0.129
2013-11-03   0.129
2013-11-04   0.129
```

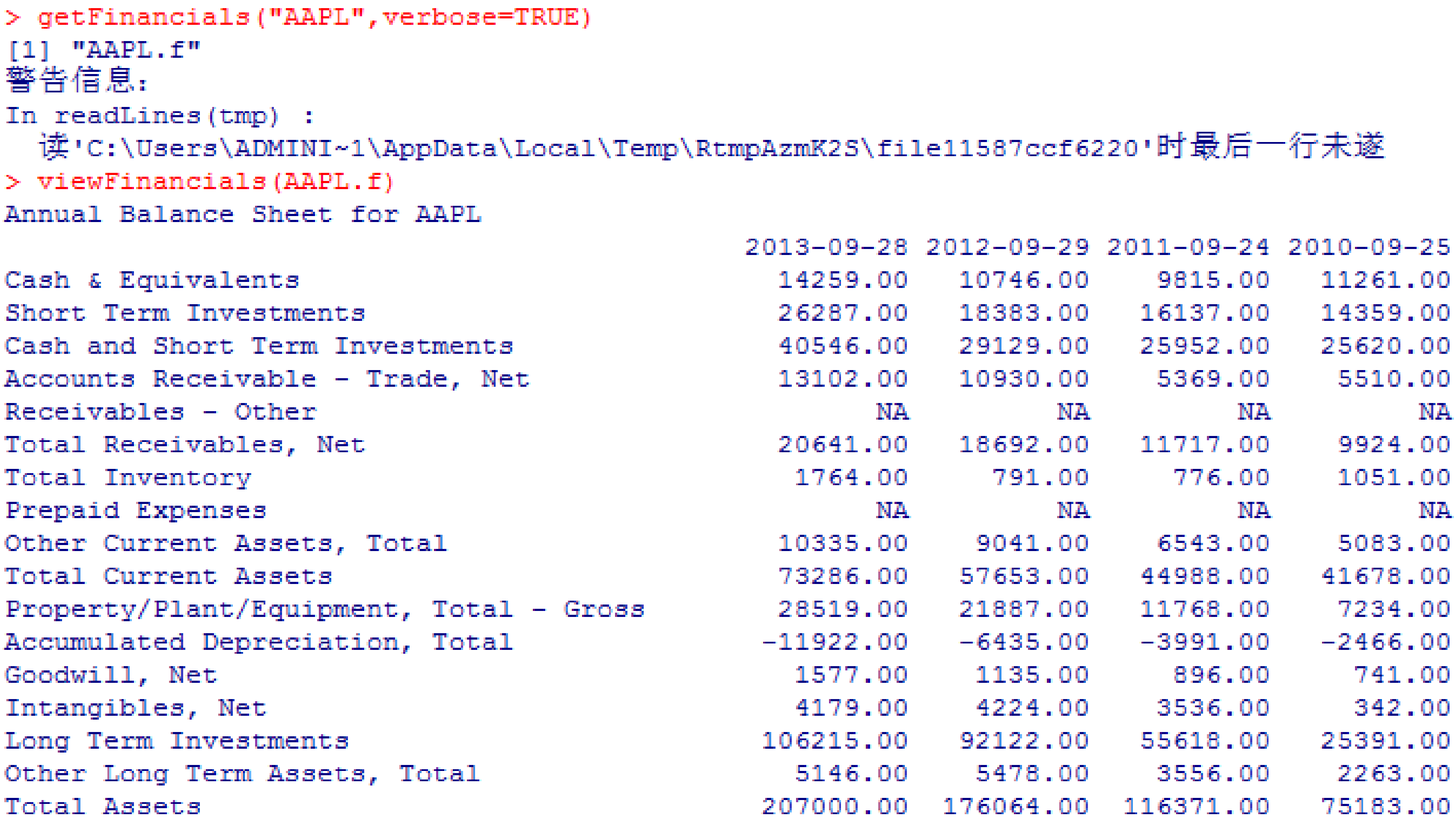
getFianacials()函数

- 从Google Finance上下载财务报表，其中包括income statement(IS), Balance Sheet(BS)和Cash Flow Statement(CF)



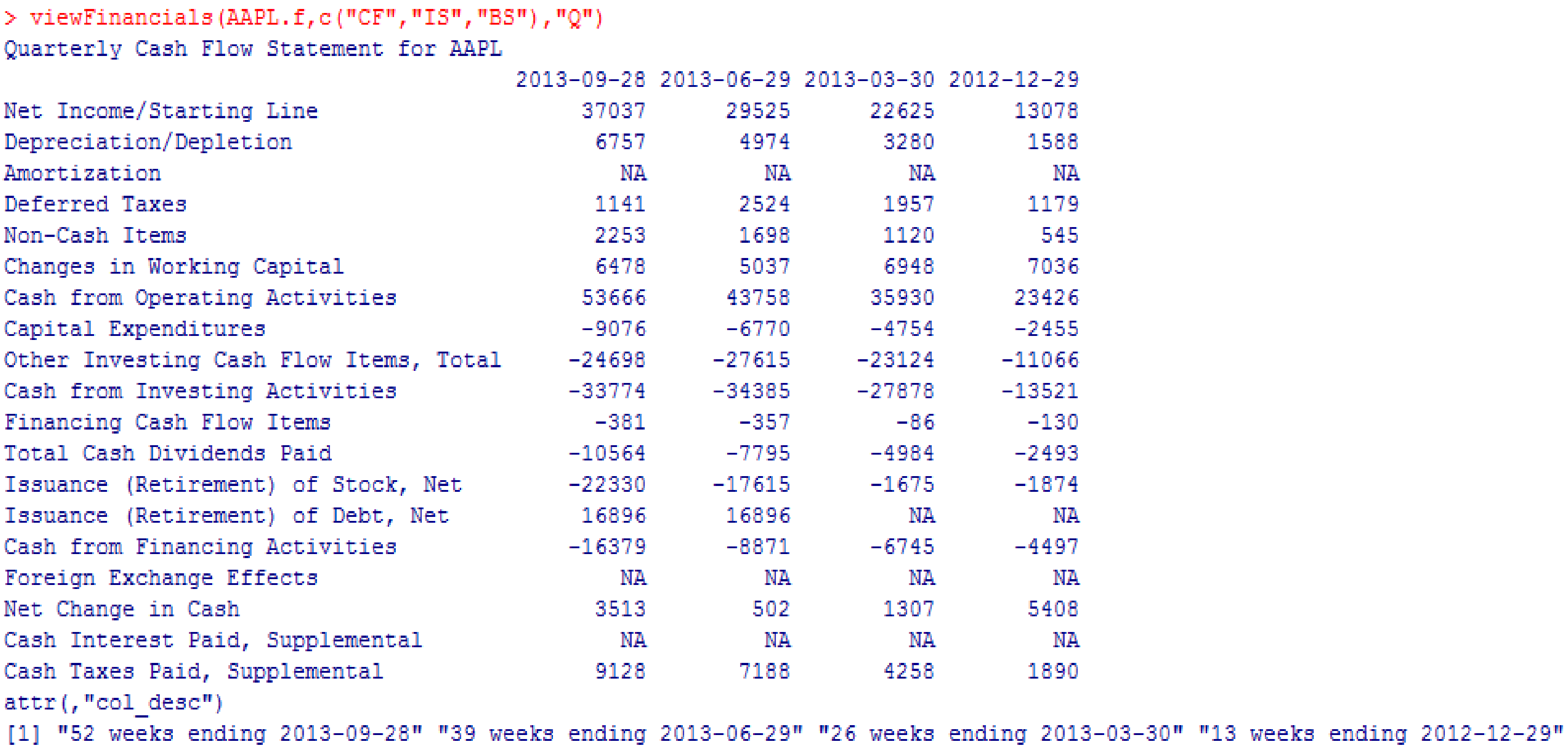
```
> getFinancials("AAPL")
[1] "AAPL.f"
警告信息:
In readLines(tmp) :
  读'C:\Users\ADMINI~1\AppData\Local\Temp\RtmpAzmK2S\file11587ba43369'时最后一行未遂
> viewFinancials(AAPL.f)
Annual Balance Sheet for AAPL
                                             2013-09-28 2012-09-29
Cash & Equivalents                             14259.00   10746.00
Short Term Investments                         26287.00   18383.00
Cash and Short Term Investments                40546.00   29129.00
Accounts Receivable - Trade, Net               13102.00   10930.00
Receivables - Other                                  NA         NA
Total Receivables, Net                         20641.00   18692.00
Total Inventory                                 1764.00     791.00
Prepaid Expenses                                     NA         NA
Other Current Assets, Total                    10335.00    9041.00
Total Current Assets                           73286.00   57653.00
Property/Plant/Equipment, Total - Gross        28519.00   21887.00
Accumulated Depreciation, Total               -11922.00   -6435.00
Goodwill, Net                                   1577.00    1135.00
Intangibles, Net                                4179.00    4224.00
Long Term Investments                         106215.00   92122.00
Other Long Term Assets, Total                   5146.00    5478.00
Total Assets                                  207000.00  176064.00
Accounts Payable                               22367.00   21175.00
```

getFianacials()函数



```
> getFinancials("AAPL",verbose=TRUE)
[1] "AAPL.f"
警告信息:
In readLines(tmp) :
  读'C:\Users\ADMINI~1\AppData\Local\Temp\RtmpAzmK2S\file11587ccf6220'时最后一行未遂
> viewFinancials(AAPL.f)
Annual Balance Sheet for AAPL
                                             2013-09-28 2012-09-29 2011-09-24 2010-09-25
Cash & Equivalents                             14259.00   10746.00    9815.00   11261.00
Short Term Investments                         26287.00   18383.00   16137.00   14359.00
Cash and Short Term Investments                40546.00   29129.00   25952.00   25620.00
Accounts Receivable - Trade, Net               13102.00   10930.00    5369.00    5510.00
Receivables - Other                                  NA         NA         NA         NA
Total Receivables, Net                         20641.00   18692.00   11717.00    9924.00
Total Inventory                                 1764.00     791.00     776.00    1051.00
Prepaid Expenses                                     NA         NA         NA         NA
Other Current Assets, Total                    10335.00    9041.00    6543.00    5083.00
Total Current Assets                           73286.00   57653.00   44988.00   41678.00
Property/Plant/Equipment, Total - Gross        28519.00   21887.00   11768.00    7234.00
Accumulated Depreciation, Total               -11922.00   -6435.00   -3991.00   -2466.00
Goodwill, Net                                   1577.00    1135.00     896.00     741.00
Intangibles, Net                                4179.00    4224.00    3536.00     342.00
Long Term Investments                         106215.00   92122.00   55618.00   25391.00
Other Long Term Assets, Total                   5146.00    5478.00    3556.00    2263.00
Total Assets                                  207000.00  176064.00  116371.00   75183.00
```

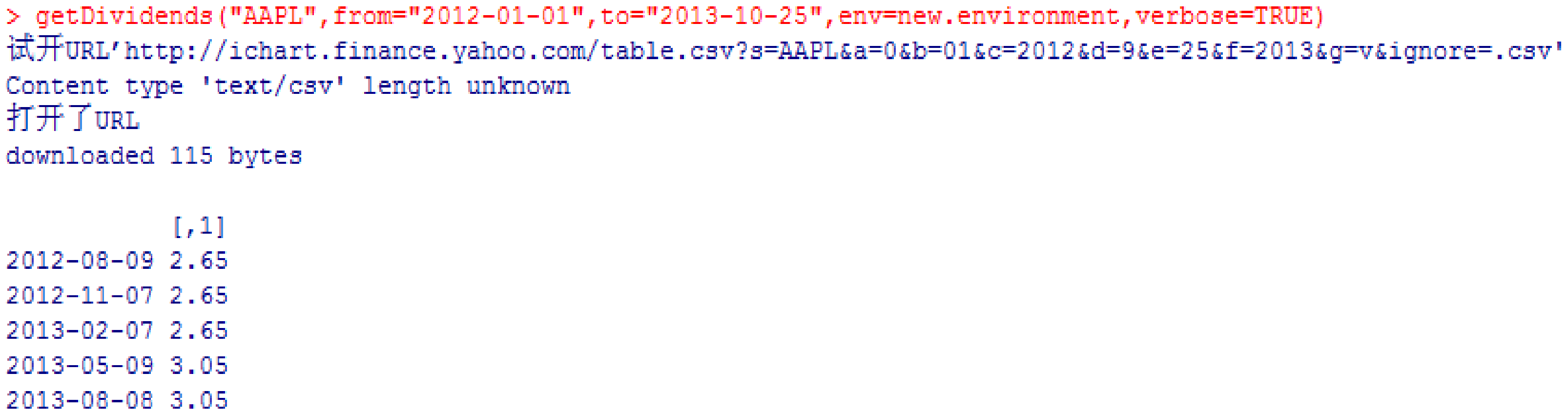
getFianacials()函数



```
> viewFinancials(AAPL.f,c("CF","IS","BS"),"Q")
Quarterly Cash Flow Statement for AAPL
                                       2013-09-28 2013-06-29 2013-03-30 2012-12-29
Net Income/Starting Line                    37037      29525      22625      13078
Depreciation/Depletion                       6757       4974       3280       1588
Amortization                                   NA         NA         NA         NA
Deferred Taxes                               1141       2524       1957       1179
Non-Cash Items                               2253       1698       1120        545
Changes in Working Capital                   6478       5037       6948       7036
Cash from Operating Activities              53666      43758      35930      23426
Capital Expenditures                        -9076      -6770      -4754      -2455
Other Investing Cash Flow Items, Total     -24698     -27615     -23124     -11066
Cash from Investing Activities             -33774     -34385     -27878     -13521
Financing Cash Flow Items                    -381       -357        -86       -130
Total Cash Dividends Paid                  -10564      -7795      -4984      -2493
Issuance (Retirement) of Stock, Net        -22330     -17615      -1675      -1874
Issuance (Retirement) of Debt, Net          16896      16896         NA         NA
Cash from Financing Activities             -16379      -8871      -6745      -4497
Foreign Exchange Effects                       NA         NA         NA         NA
Net Change in Cash                           3513        502       1307       5408
Cash Interest Paid, Supplemental               NA         NA         NA         NA
Cash Taxes Paid, Supplemental                9128       7188       4258       1890
attr(,"col_desc")
[1] "52 weeks ending 2013-09-28" "39 weeks ending 2013-06-29" "26 weeks ending 2013-03-30" "13 weeks ending 2012-12-29"
```

getDividends()函数

- 从Yahoo！ Finance上下载股息数据



```
> getDividends("AAPL",from="2012-01-01",to="2013-10-25",env=new.environment,verbose=TRUE)
试开URL'http://ichart.finance.yahoo.com/table.csv?s=AAPL&a=0&b=01&c=2012&d=9&e=25&f=2013&g=v&ignore=.csv'
Content type 'text/csv' length unknown
打开了URL
downloaded 115 bytes

           [,1]
2012-08-09 2.65
2012-11-07 2.65
2013-02-07 2.65
2013-05-09 3.05
2013-08-08 3.05
```

分析类函数

- is族函数
- has族函数
- 列名函数
- 计算函数

is 族函数

- 判断某数据是否是某类型的数据

is.OHLC()

is.OHLCV()

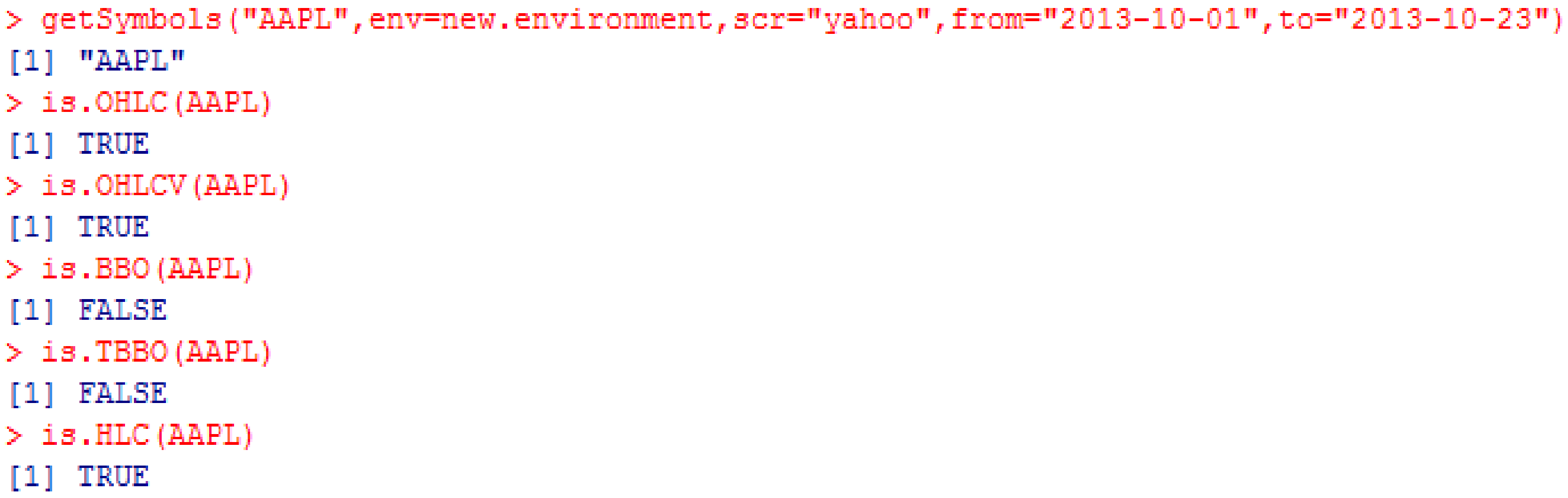
is.BBO()

is.TBBO()

is.HLC()

is 族函数

- Example



```
> getSymbols("AAPL",env=new.environment,scr="yahoo",from="2013-10-01",to="2013-10-23")
[1] "AAPL"
> is.OHLC(AAPL)
[1] TRUE
> is.OHLCV(AAPL)
[1] TRUE
> is.BBO(AAPL)
[1] FALSE
> is.TBBO(AAPL)
[1] FALSE
> is.HLC(AAPL)
[1] TRUE
```

has族函数

- 检查数据里面是否包含某类型的数据

has.OHLC()

has.HLC()

has.OHLCV()

has.Op()

has.Hi()

has.Lo()

has.Cl()

has.Vo()

has.Ad()

has.Ask()

has.Bid()

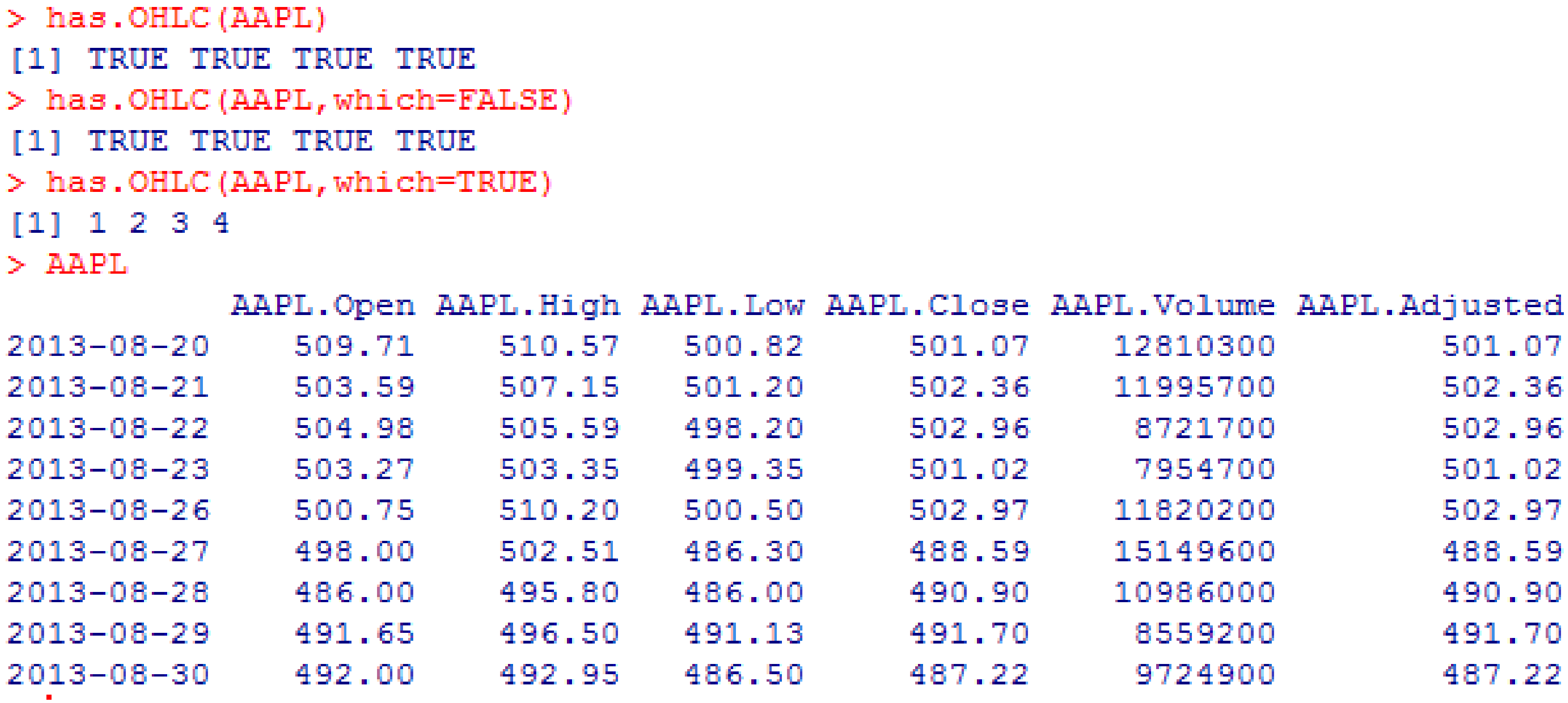
has.Price()

has.Qty()

has.Trade()

has族函数

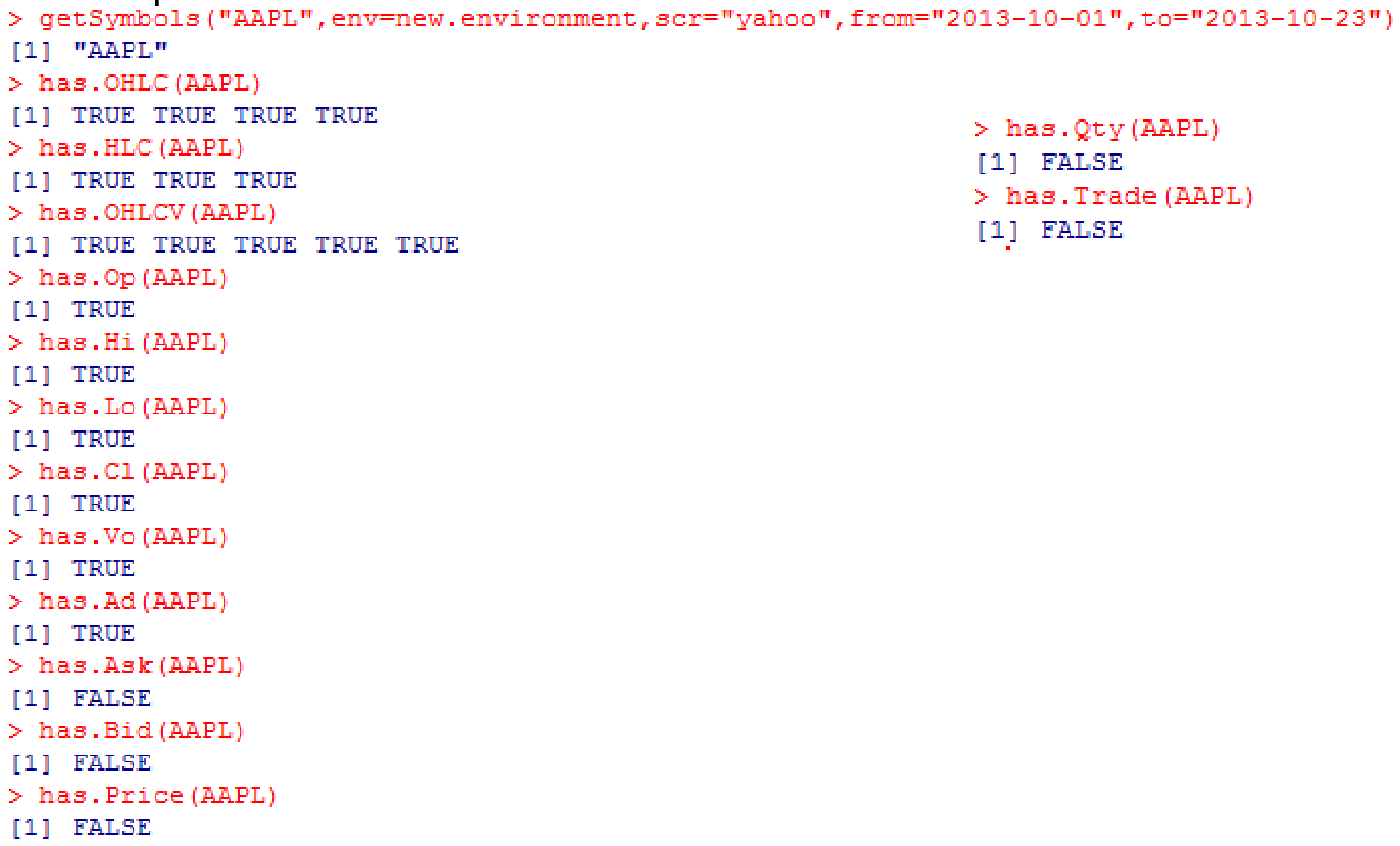
- Example



```
> has.OHLC(AAPL)
[1] TRUE TRUE TRUE TRUE
> has.OHLC(AAPL,which=FALSE)
[1] TRUE TRUE TRUE TRUE
> has.OHLC(AAPL,which=TRUE)
[1] 1 2 3 4
> AAPL
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-20    509.71    510.57   500.82     501.07    12810300        501.07
2013-08-21    503.59    507.15   501.20     502.36    11995700        502.36
2013-08-22    504.98    505.59   498.20     502.96     8721700        502.96
2013-08-23    503.27    503.35   499.35     501.02     7954700        501.02
2013-08-26    500.75    510.20   500.50     502.97    11820200        502.97
2013-08-27    498.00    502.51   486.30     488.59    15149600        488.59
2013-08-28    486.00    495.80   486.00     490.90    10986000        490.90
2013-08-29    491.65    496.50   491.13     491.70     8559200        491.70
2013-08-30    492.00    492.95   486.50     487.22     9724900        487.22
```

has族函数

- Example



```
> getSymbols("AAPL",env=new.environment,scr="yahoo",from="2013-10-01",to="2013-10-23")
[1] "AAPL"
> has.OHLC(AAPL)
[1] TRUE TRUE TRUE TRUE
> has.HLC(AAPL)
[1] TRUE TRUE TRUE
> has.OHLCV(AAPL)
[1] TRUE TRUE TRUE TRUE TRUE
> has.Op(AAPL)
[1] TRUE
> has.Hi(AAPL)
[1] TRUE
> has.Lo(AAPL)
[1] TRUE
> has.Cl(AAPL)
[1] TRUE
> has.Vo(AAPL)
[1] TRUE
> has.Ad(AAPL)
[1] TRUE
> has.Ask(AAPL)
[1] FALSE
> has.Bid(AAPL)
[1] FALSE
> has.Price(AAPL)
[1] FALSE
```

```
> has.Qty(AAPL)
[1] FALSE
> has.Trade(AAPL)
[1] FALSE
```

列名函数

- 提取某种数据类型

Op()

Hi()

Lo()

Cl()

Vo()

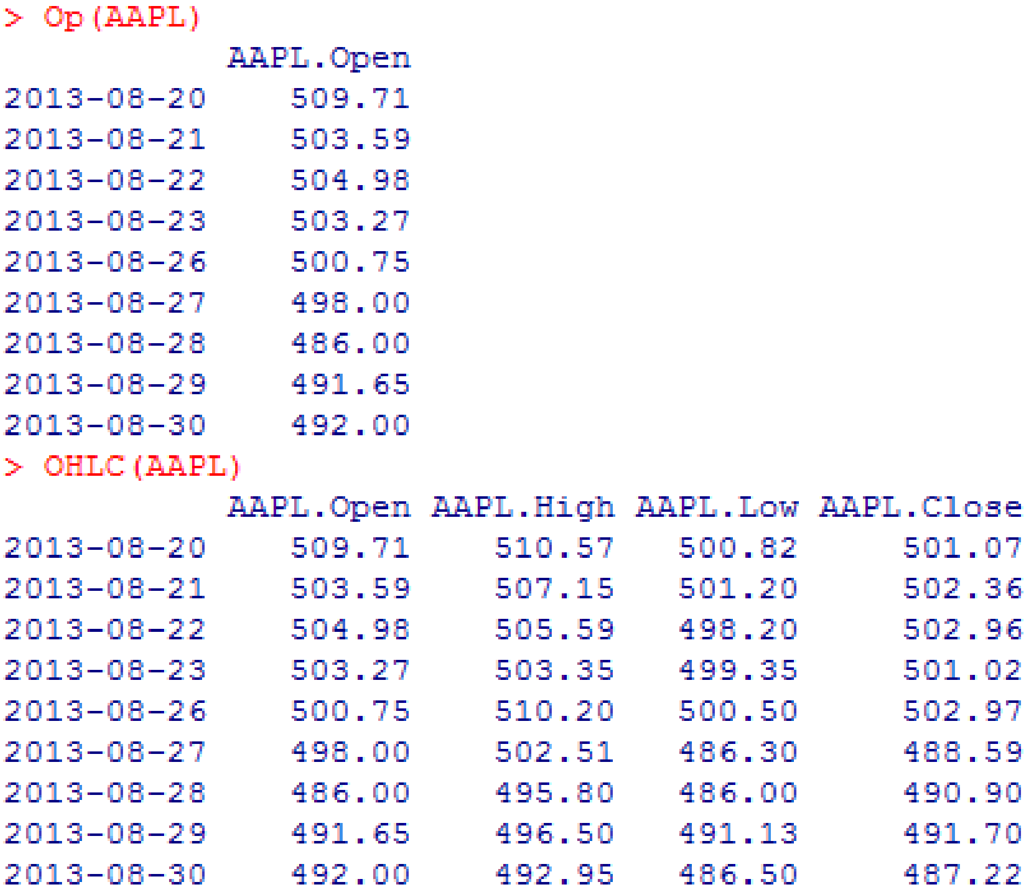
Ad()

HLC()

OHLC()

列名函数

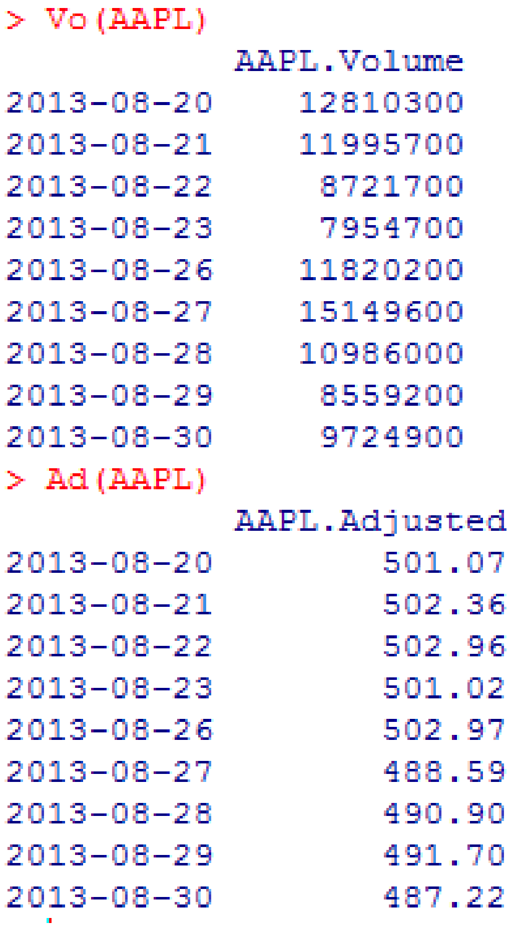
- Example



```
> Op(AAPL)
           AAPL.Open
2013-08-20    509.71
2013-08-21    503.59
2013-08-22    504.98
2013-08-23    503.27
2013-08-26    500.75
2013-08-27    498.00
2013-08-28    486.00
2013-08-29    491.65
2013-08-30    492.00
> OHLC(AAPL)
           AAPL.Open AAPL.High AAPL.Low AAPL.Close
2013-08-20    509.71    510.57   500.82     501.07
2013-08-21    503.59    507.15   501.20     502.36
2013-08-22    504.98    505.59   498.20     502.96
2013-08-23    503.27    503.35   499.35     501.02
2013-08-26    500.75    510.20   500.50     502.97
2013-08-27    498.00    502.51   486.30     488.59
2013-08-28    486.00    495.80   486.00     490.90
2013-08-29    491.65    496.50   491.13     491.70
2013-08-30    492.00    492.95   486.50     487.22
```

列名函数

- Example



```
> Vo(AAPL)
           AAPL.Volume
2013-08-20    12810300
2013-08-21    11995700
2013-08-22     8721700
2013-08-23     7954700
2013-08-26    11820200
2013-08-27    15149600
2013-08-28    10986000
2013-08-29     8559200
2013-08-30     9724900
> Ad(AAPL)
           AAPL.Adjusted
2013-08-20        501.07
2013-08-21        502.36
2013-08-22        502.96
2013-08-23        501.02
2013-08-26        502.97
2013-08-27        488.59
2013-08-28        490.90
2013-08-29        491.70
2013-08-30        487.22
```

计算函数

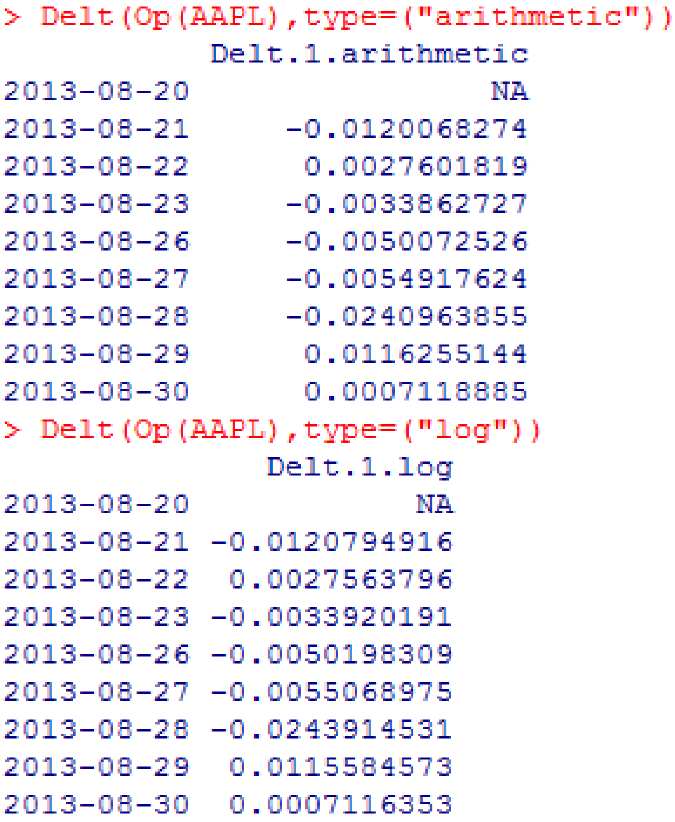
Delt()	计算变化率
Lag()	求滞后k期
Next()	求k个后
first()	求前k个
last()	求后k个
findPeaks()	找出峰值
findValleys()	找出谷值
seriesIncr()	差分后大于限值的点
seriesDecr()	差分后小于限值的点
endpoints()	寻找节点
to.weekly()	将OHLC数据转化为周数据
to.monthly()	将PHLC数据转化为月数据
periodicity()	返回数据的日期范围

Delt()函数

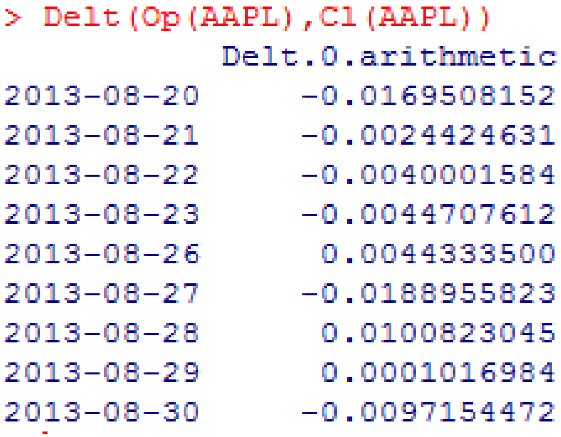
- 主要是用来计算一个序列的一个阶段到另一个阶段的变化率或者计算两个序列之间的变化率

Delt()函数

- Example



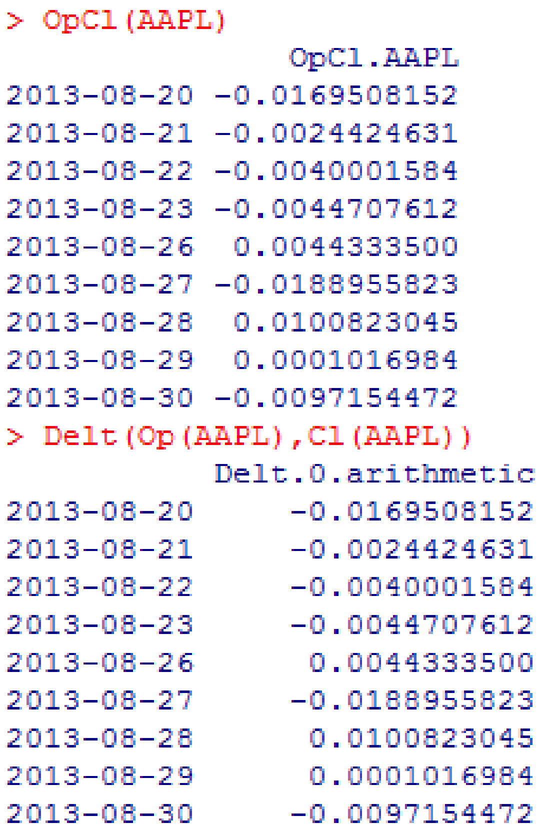
```
> Delt(Op(AAPL),type=("arithmetic"))
           Delt.1.arithmetic
2013-08-20                NA
2013-08-21     -0.0120068274
2013-08-22      0.0027601819
2013-08-23     -0.0033862727
2013-08-26     -0.0050072526
2013-08-27     -0.0054917624
2013-08-28     -0.0240963855
2013-08-29      0.0116255144
2013-08-30      0.0007118885
> Delt(Op(AAPL),type=("log"))
              Delt.1.log
2013-08-20            NA
2013-08-21 -0.0120794916
2013-08-22  0.0027563796
2013-08-23 -0.0033920191
2013-08-26 -0.0050198309
2013-08-27 -0.0055068975
2013-08-28 -0.0243914531
2013-08-29  0.0115584573
2013-08-30  0.0007116353
```



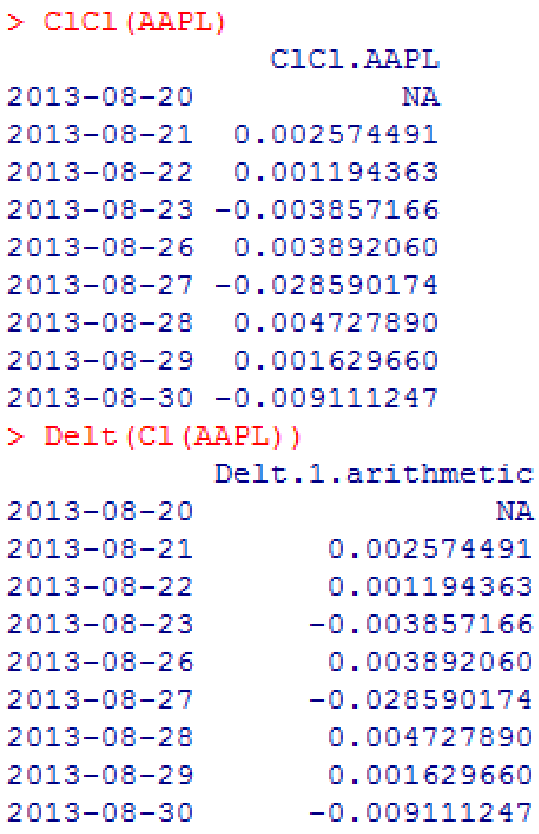
```
> Delt(Op(AAPL),Cl(AAPL))
           Delt.0.arithmetic
2013-08-20     -0.0169508152
2013-08-21     -0.0024424631
2013-08-22     -0.0040001584
2013-08-23     -0.0044707612
2013-08-26      0.0044333500
2013-08-27     -0.0188955823
2013-08-28      0.0100823045
2013-08-29      0.0001016984
2013-08-30     -0.0097154472
```

Delt()函数

- 另外，Delt()函数还有一些比较简便的表达方式,如下面



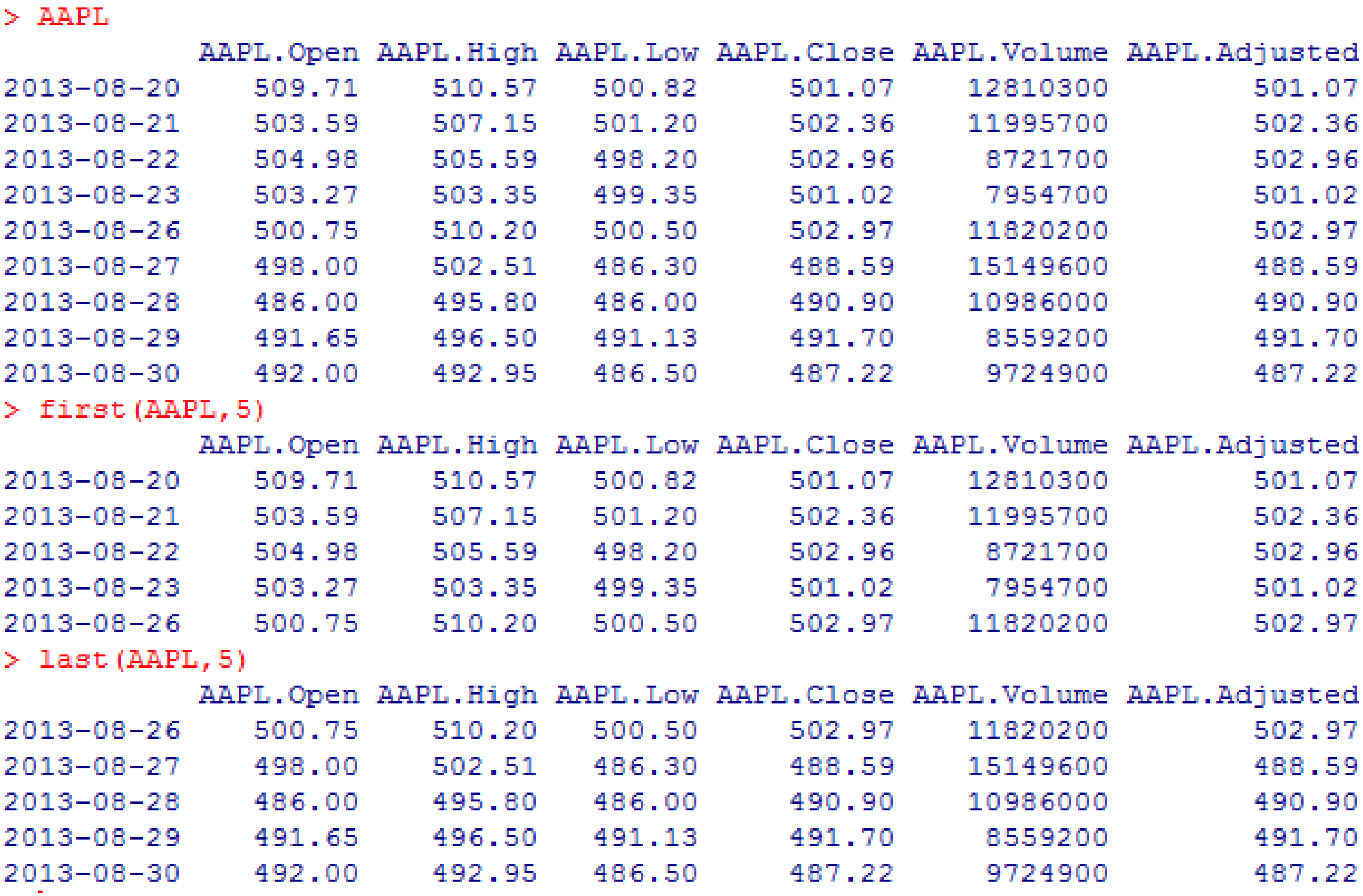
```
> OpCl(AAPL)
                  OpCl.AAPL
2013-08-20 -0.0169508152
2013-08-21 -0.0024424631
2013-08-22 -0.0040001584
2013-08-23 -0.0044707612
2013-08-26  0.0044333500
2013-08-27 -0.0188955823
2013-08-28  0.0100823045
2013-08-29  0.0001016984
2013-08-30 -0.0097154472
> Delt(Op(AAPL),Cl(AAPL))
           Delt.0.arithmetic
2013-08-20     -0.0169508152
2013-08-21     -0.0024424631
2013-08-22     -0.0040001584
2013-08-23     -0.0044707612
2013-08-26      0.0044333500
2013-08-27     -0.0188955823
2013-08-28      0.0100823045
2013-08-29      0.0001016984
2013-08-30     -0.0097154472
```



```
> ClCl(AAPL)
               ClCl.AAPL
2013-08-20            NA
2013-08-21  0.002574491
2013-08-22  0.001194363
2013-08-23 -0.003857166
2013-08-26  0.003892060
2013-08-27 -0.028590174
2013-08-28  0.004727890
2013-08-29  0.001629660
2013-08-30 -0.009111247
> Delt(Cl(AAPL))
           Delt.1.arithmetic
2013-08-20                NA
2013-08-21       0.002574491
2013-08-22       0.001194363
2013-08-23      -0.003857166
2013-08-26       0.003892060
2013-08-27      -0.028590174
2013-08-28       0.004727890
2013-08-29       0.001629660
2013-08-30      -0.009111247
```

first()函数和last()函数

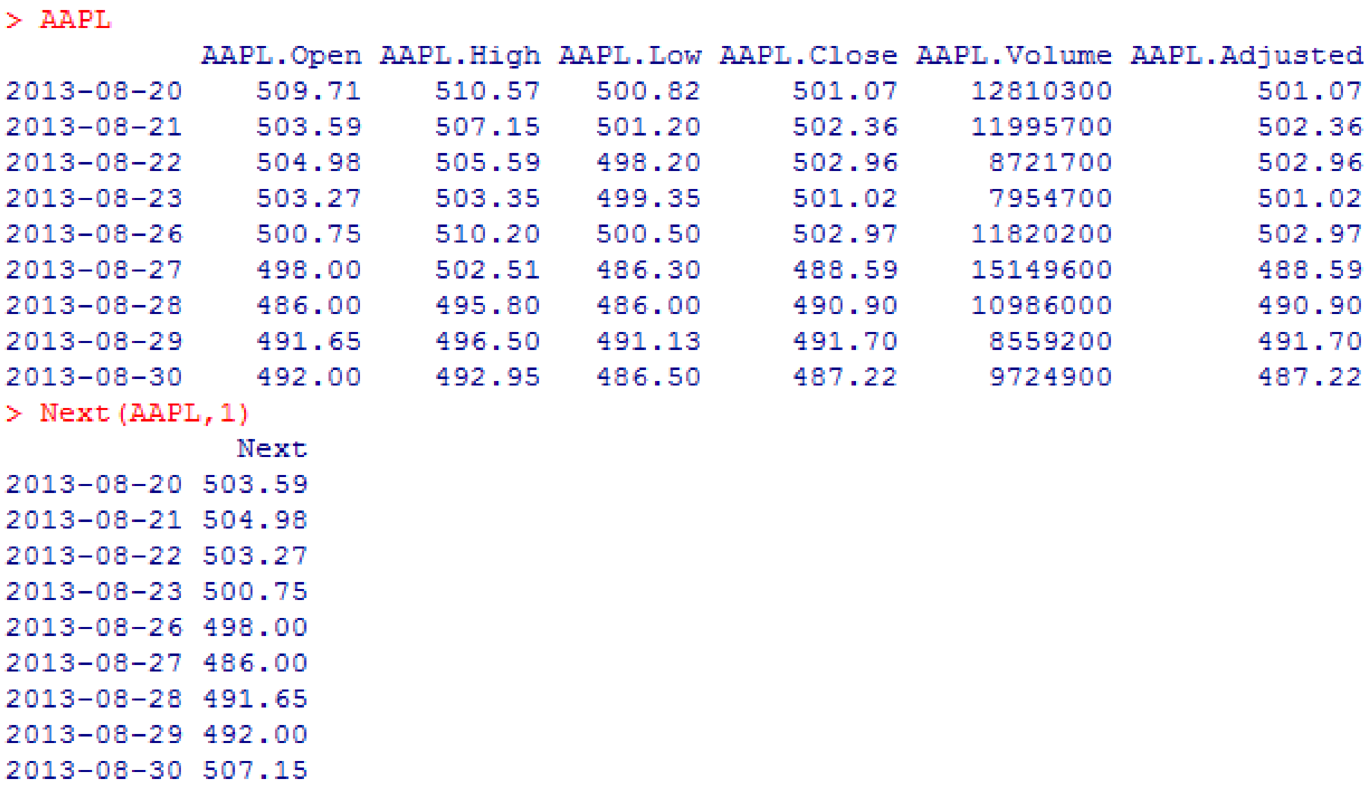
- 求出对象的前k个元素或者后k个元素



```
> AAPL
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-20    509.71    510.57   500.82     501.07    12810300        501.07
2013-08-21    503.59    507.15   501.20     502.36    11995700        502.36
2013-08-22    504.98    505.59   498.20     502.96     8721700        502.96
2013-08-23    503.27    503.35   499.35     501.02     7954700        501.02
2013-08-26    500.75    510.20   500.50     502.97    11820200        502.97
2013-08-27    498.00    502.51   486.30     488.59    15149600        488.59
2013-08-28    486.00    495.80   486.00     490.90    10986000        490.90
2013-08-29    491.65    496.50   491.13     491.70     8559200        491.70
2013-08-30    492.00    492.95   486.50     487.22     9724900        487.22
> first(AAPL,5)
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-20    509.71    510.57   500.82     501.07    12810300        501.07
2013-08-21    503.59    507.15   501.20     502.36    11995700        502.36
2013-08-22    504.98    505.59   498.20     502.96     8721700        502.96
2013-08-23    503.27    503.35   499.35     501.02     7954700        501.02
2013-08-26    500.75    510.20   500.50     502.97    11820200        502.97
> last(AAPL,5)
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-26    500.75    510.20   500.50     502.97    11820200        502.97
2013-08-27    498.00    502.51   486.30     488.59    15149600        488.59
2013-08-28    486.00    495.80   486.00     490.90    10986000        490.90
2013-08-29    491.65    496.50   491.13     491.70     8559200        491.70
2013-08-30    492.00    492.95   486.50     487.22     9724900        487.22
```

Next()函数

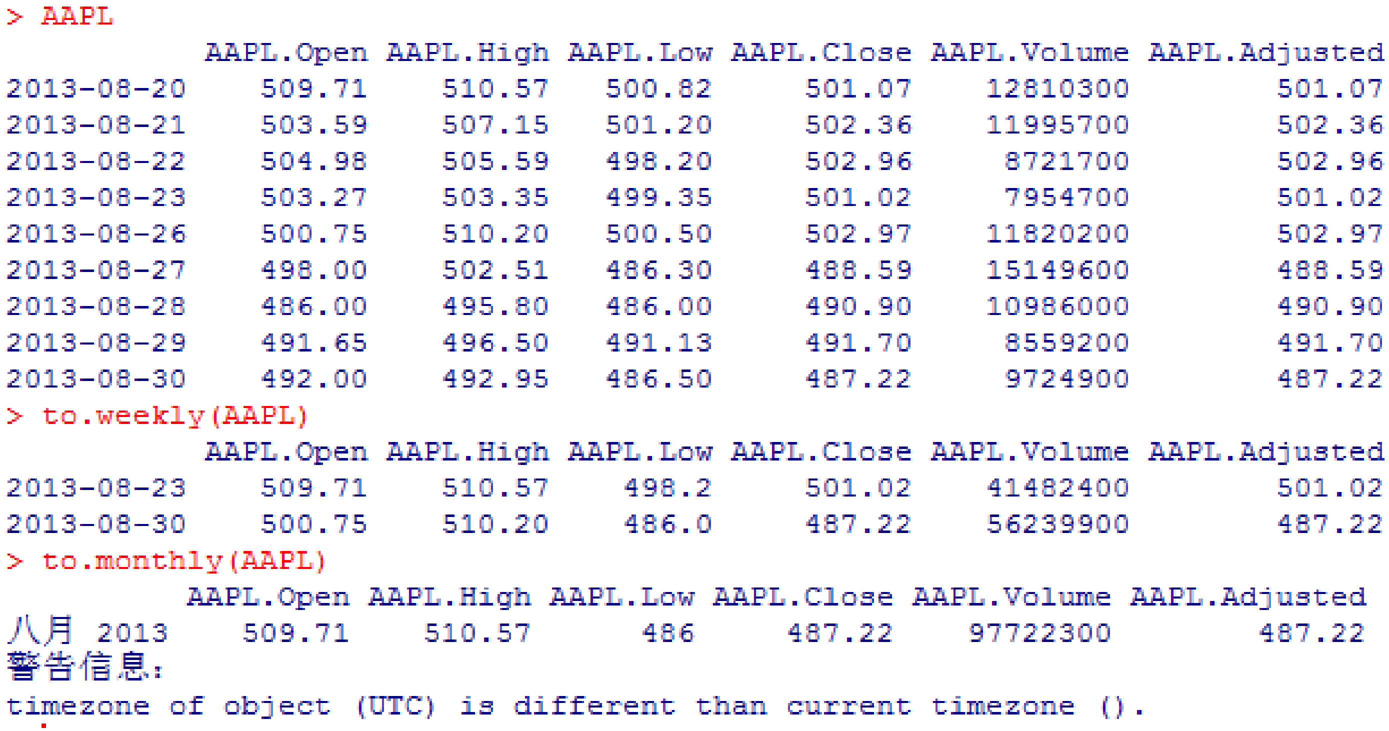
- 所有观测值的值前进k个单位



```
> AAPL
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-20    509.71    510.57   500.82     501.07    12810300        501.07
2013-08-21    503.59    507.15   501.20     502.36    11995700        502.36
2013-08-22    504.98    505.59   498.20     502.96     8721700        502.96
2013-08-23    503.27    503.35   499.35     501.02     7954700        501.02
2013-08-26    500.75    510.20   500.50     502.97    11820200        502.97
2013-08-27    498.00    502.51   486.30     488.59    15149600        488.59
2013-08-28    486.00    495.80   486.00     490.90    10986000        490.90
2013-08-29    491.65    496.50   491.13     491.70     8559200        491.70
2013-08-30    492.00    492.95   486.50     487.22     9724900        487.22
> Next(AAPL,1)
             Next
2013-08-20 503.59
2013-08-21 504.98
2013-08-22 503.27
2013-08-23 500.75
2013-08-26 498.00
2013-08-27 486.00
2013-08-28 491.65
2013-08-29 492.00
2013-08-30 507.15
```

to.weekly()和to.monthly()函数

- 把OHLC数据转化为周数据或者月数据



```
> AAPL
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-20    509.71    510.57   500.82     501.07    12810300        501.07
2013-08-21    503.59    507.15   501.20     502.36    11995700        502.36
2013-08-22    504.98    505.59   498.20     502.96     8721700        502.96
2013-08-23    503.27    503.35   499.35     501.02     7954700        501.02
2013-08-26    500.75    510.20   500.50     502.97    11820200        502.97
2013-08-27    498.00    502.51   486.30     488.59    15149600        488.59
2013-08-28    486.00    495.80   486.00     490.90    10986000        490.90
2013-08-29    491.65    496.50   491.13     491.70     8559200        491.70
2013-08-30    492.00    492.95   486.50     487.22     9724900        487.22
> to.weekly(AAPL)
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2013-08-23    509.71    510.57    498.2     501.02    41482400        501.02
2013-08-30    500.75    510.20    486.0     487.22    56239900        487.22
> to.monthly(AAPL)
          AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
八月 2013    509.71    510.57      486     487.22    97722300        487.22
警告信息:
timezone of object (UTC) is different than current timezone ().
```

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间