



데이터 분석 기반의 의사결정을 위한

finance analytics

버전:

0.95

작성자:

유충현

2016년 4월 20일

차 례

제 1 장 패키지 설치하기	3
제 1 절 Utility 설치	3
1.1 QuantLib 설치	3
1.2 SYMPHONY 설치	5
제 2 절 R packages 설치	6
제 3 절 R packages 설치를 위한 사용자 정의 함수	9
제 4 절 Time series analysis packages 설치	9
제 5 절 Rmetrics suite of packages 설치	9
제 6 절 Optimization and mathematical programming packages 설치	10
제 7 절 기타 finance packages 설치	10
제 2 장 Quandl finance 데이터 수집하기	11
제 1 절 quandl.com의 소개	11
제 2 절 제공 데이터	12
제 3 절 시카고 상품 거래소 선물 데이터 가져오기	14
3.1 Code Formats	14
3.2 데이터 가져오기	15
3.3 메타 데이터 가져오기	18
제 4 절 런던 금과 시장 협회 데이터 가져오기	21
4.1 Code Formats	21
4.2 데이터 가져오기	21
4.3 메타 데이터 가져오기	23
제 5 절 미국 에너지 정보 관리국 데이터 가져오기	25
5.1 Code Formats	26
5.2 데이터 가져오기	29
5.3 메타 데이터 가져오기	31
제 6 절 상하이 선물 거래소 데이터 가져오기	35
6.1 Code Formats	35
6.2 데이터 가져오기	36
6.3 메타 데이터 가져오기	39
제 7 절 연방 준비 제도 이사회 (FRB) 경제 지표 가져오기	42
7.1 Code Formats	42
7.2 데이터 가져오기	44
7.3 메타 데이터 가져오기	45

제 3 장 주가 데이터 수집하기	49
제 1 절 국내 주가데이터 수집	49
1.1 종목코드 데이터 수집	49
1.2 종목코드 데이터 적재	49
1.3 종목별 거래정보 가져오기	51
1.4 관심종목 거래정보 가져오기	52
제 2 절 해외 주가데이터 수집	54
2.1 미국 종목코드 데이터 수집	54
2.2 관심종목 거래정보 가져오기	56
제 3 절 수익률 분석	57
3.1 한화생명 수익률 분석	57
3.2 최고가/최저가	57
3.3 단순 수익률/복리 수익률	58
3.4 Value-at-Risk	59
3.5 복합 연간 성장률	60
3.6 삼성생명 수익률 분석	60
3.7 최고가/최저가	60
3.8 단순 수익률/복리 수익률	60
3.9 Value-at-Risk	61
3.10 복합 연간 성장률	62

제 1 장

패키지 설치하기

제 1 절 Utility 설치

몇몇 R패키지는 OS의 Utility를 사용한다. 이러한 패키지는 연산의 대부분의 OS의 Utility를 이용하기 때문에, R과 OS의 Utility를 인터페이스하는 역할을 한다. 대표적인 패키지와 OS의 Utility는 다음과 같다.

- RQuantLib package
 - connects GNU R with QuantLib
 - QuantLib
 - * <http://quantlib.org>
 - * A free/open-source library for quantitative finance
- Rsymphony package
 - SYMPHONY
 - * <https://projects.coin-or.org/SYMPHONY>
 - * open-source solver for mixed-integer linear programs (MILPs) written in C.

1.1 QuantLib 설치

다음을 참고하여, mac console에서 QuantLib를 수행한다.

- boost 설치
 - QuantLib를 설치하기 위해서 사전에 설치해야 함
 - Boost provides free peer-reviewed portable C++ source libraries.
 - <http://www.pyimagesearch.com/2015/04/27/installing-boost-and-boost-python-on-osx-with-homebrew/>
- QuantLib 설치
 - <http://quantlib.org/install/macosx.shtml>

1.1.1 Install Homebrew

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

1.1.2 Update Homebrew

```
brew update
```

1.1.3 Install Python

```
brew install python
```

1.1.4 Installing boost

```
brew install boost --with-python
```

1.1.5 Installing boost-python

```
brew install boost-python
```

1.1.6 Confirm boost and boost-python is installed

설치된 프로그램 확인함

```
brew list | grep 'boost'
```

1.1.7 QuantLib download

<http://sourceforge.net/projects/quantlib/files/>에서 프로그램을 다운로드함. 여기서는 1.7.1 버전인 QuantLib-1.7.1.tar.gz을 다운로드함.

1.1.8 QuantLib 압축 풀기

```
tar xzvf QuantLib-1.7.1.tar.gz  
cd QuantLib-1.7.1
```

1.1.9 QuantLib 환경설정

```
./configure --enable-static --with-boost-include=/opt/local/include/ \
--with-boost-lib=/opt/local/lib/ --prefix=/opt/local/
```

1.1.10 QuantLib 컴파일 및 설치

```
make && sudo make install
```

1.1.11 QuantLib 설치 확인

예제인 BermudanSwaption을 컴파일 한 후 실행하여 정상 동작하면, QuantLib가 정상적으로 설치된 것이다.

```
cd Examples/BermudanSwaption
g++ -I/opt/local/include/ -I/opt/local/include/boost BermudanSwaption.cpp \
-o bermudanswaption -L/opt/local/lib/ -lQuantLib
./bermudanswaption
```

1.1.12 Boost Configuration

If you want to make the Boost headers and libraries available to all C++ projects, edit `~/.bash_profile` and add the following lines into the file

```
echo 'export CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/opt/local/include' >> ~/.bash_profile
echo 'export C_INCLUDE_PATH=$C_INCLUDE_PATH:/opt/local/include' >> ~/.bash_profile
echo 'export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/opt/local/lib' >> ~/.bash_profile
```

1.2 SYMPHONY 설치

1.2.1 SYMPHONY 다운로드 및 압축 풀기

<http://www.coin-or.org/download/source/SYMPHONY/>에서 소스 파일을 다운로드한다. 여기서는 5.6.14 버전인 SYMPHONY-5.6.14.tgz을 다운로드하였다.

```
tar -xzf SYMPHONY-5.6.14.tgz
```

1.2.2 SYMPHONY 환경설정

SYMPHONY를 /usr/local 디렉토리에 설치하기 위해서 prefix를 설정하였다.

```
cd SYMPHONY-5.6.14
./configure --prefix=/usr/local
```

1.2.3 SYMPHONY 컴파일 및 설치

```
make
make install
```

1.2.4 SYMPHONY 소스코드 설치

```
cd ..
rm -rf SYMPHONY-5.6.14
```

제 2 절 R packages 설치

finance analytics에 특화된 다음 R packages를 설치한다.

- Time series analysis
 - forecast
 - * Forecasting functions for time series and linear
 - SDD
 - * Serial dependence diagrams
 - Wats
 - * Wrap around time series graphics
 - dygraphs
 - * Interface to ‘Dygraphs’ interactive time series charting library
 - ZRA
 - * Dynamic plots for time series forecasting
 - vars
 - * VAR modelling
 - fanplot
 - * Visualisation of sequential probability distributions using fan charts
 - lubridate
 - * Make dealing with dates a little easier
 - tis
 - * Time indexes and time indexed series
 - mondate
 - * Keep track of dates in terms of months
 - tseries
 - * Time series analysis and computational finance
 - fts

- * R interface to tslib (a time series library in c++)
- MAPA
 - * Multiple aggregation prediction algorithm
- bayesGARCH
 - * Bayesian estimation of the GARCH(1,1) model with student-t innovations
- Rmetrics suite of packages
 - fArma
 - * ARMA time series modelling
 - fAsianOptions
 - * EBM and asian option valuation
 - fAssets
 - * Analysing and modelling financial assets
 - fBasics
 - * Markets and basic statistics
 - fBonds
 - * Bonds and interest rate models
 - timeDate
 - * Chronological and calendar objects
 - fCopulae
 - * Bivariate dependence structures with copulae
 - fExoticOptions
 - * Exotic option valuation
 - fExtremes
 - * Extreme financial market data
 - fGarch
 - * Autoregressive conditional heteroskedastic modelling
 - fImport
 - * Economic and financial data import
 - fNonlinear
 - * Nonlinear and chaotic time series modelling
 - fOptions
 - * Pricing and evaluating basic options
 - fPortfolio
 - * Portfolio selection and optimization

- fRegression
 - * Regression based decision and prediction
- timeSeries
 - * Financial time series objects
- fTrading
 - * Technical trading analysis
- fUnitRoots
 - * Trends and unit roots
- Optimization and mathematical programming
 - optimx
 - * A replacement and extension of the optim() function
 - ROI
 - * R optimization infrastructure
 - alabama
 - * Constrained nonlinear optimization
 - NlcOptim
 - * Solve nonlinear optimization with nonlinear constraints
- 기타 라이브러리
 - RQuantLib
 - * R interface to the 'QuantLib' library
 - portfolio
 - * Analysing equity portfolios
 - portfolioSim
 - * Framework for simulating equity portfolio strategies
 - backtest
 - * Exploring portfolio-based conjectures about financial instruments
 - stockPortfolio
 - * Build stock models and analyze stock portfolios
 - PerformanceAnalytics
 - * Econometric tools for performance and risk analysis
 - TTR
 - * Technical trading rules
 - quantmod
 - * Quantitative financial modelling framework

- opefimor
 - * Option pricing and estimation of financial models in R
- Quandl
 - * API Wrapper for Quandl.com
- GUIDE
 - * GUI for DErivatives in R
- copula
 - * Multivariate dependence with copulas
- CreditMetrics
 - * Functions for calculating the creditMetrics risk model
- evir
 - * Extreme values in R

제 3 절 R packages 설치를 위한 사용자 정의 함수

```
getPackages <- function(x) {
  x <- x[!x %in% installed.packages()[, 1]]

  if (length(x) == 0)
    return()

  install.packages(x)

  x <- x[!x %in% installed.packages()[, 1]]

  if (length(x) > 0)
    sprintf("%s is not install!!", paste(x, collapse = ", "))
}
```

제 4 절 Time series analysis packages 설치

```
pkgs <- c("forecast", "SDD", "Wats", "dygraphs", "ZRA", "vars", "fanplot",
  "lubridate", "tis", "mondate", "tseries", "fts", "MAPA", "bayesGARCH")
getPackages(pkgs)
```

제 5 절 Rmetrics suite of packages 설치

```
pkgs <- c("fArma", "fAsianOptions", "fAssets", "fBasics", "fBonds",
  "timeDate", "fCopulae", "fExoticOptions", "fExtremes", "fGarch",
  "fImport", "fNonlinear", "fOptions", "fPortfolio", "fRegression",
  "timeSeries", "fTrading", "fUnitRoots")
getPackages(pkgs)
```

제 6 절 Optimization and mathematical programming packages 설치

```
pkgs <- c("optimx", "ROI", "alabama", "NlcOptim")
getPackages(pkgs)
```

제 7 절 기타 finance packages 설치

```
pkgs <- c("RQuantLib", "portfolio", "portfolioSim", "backtest", "stockPortfolio",
  "PerformanceAnalytics", "TTR", "quantmod", "openfimor", "Quandl",
  "GUIDE", "copula", "CreditMetrics")
getPackages(pkgs)
```

제 2 장

Quandl finance 데이터 수집하기

제 1 절 quandl.com의 소개

<https://www.quandl.com/> site는 많은 종류의 finance 데이터를 제공하는 site다. 이 사이트의 개괄적인 컨셉은 다음 그림 2.1과 같다.



그림 2.1: quandl.com의 소개

그림 2.1처럼 quandl.com의 장점을 정리하면 다음과 같다.

- 단일 플랫폼에서 금융 및 경제에 대한 수백개의 데이터베이스에 액세스할 수 있음
- 데이터를 가져오는 방법의 다양성
 - 웹사이트 메뉴의 이용
 - API를 이용한 데이터 가져오기
 - Quandl 패키지로 R에서 직접 가져오기
 - Quandl 패키지로 Python에서 직접 가져오기
 - Excel 데이터 다운로드
- 무료로 수백만 데이터 세트를 사용하거나 프리미엄 데이터의 단품 구입 가능

quandl.com의 데이터 제공에 대한 흐름 그림 2.2와 같다.

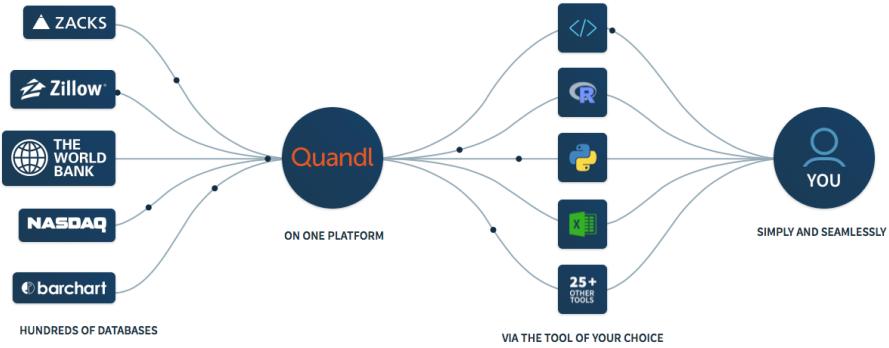


그림 2.2: quandl.com의 데이터 제공 흐름

그림 2.2처럼 사용자는 자신이 익숙한 툴을 이용해서 데이터를 가져오면 된다. 그리고, 이 글에서는 R의 API인 Quandl 패키지를 이용한다. API에 대한 간단한 사용 방법은 그림 2.3을 통해 이해할 수 있을 것이다.

```

<- library (Quandl)
> data <- Quandl("FRED/GDP")
> head (data)

      DATE      VALUE
1 2015-04-01 17902.0
2 2015-01-01 17641.3
3 2014-09-01 17615.9
4 2014-07-01 17622.1
5 2014-04-01 17270.0
6 2014-01-01 16984.3
7 2013-10-01 16957.6
8 2013-07-01 16727.5
9 2013-04-01 16526.8
10 2013-01-01 16440.7
11 2012-10-01 16297.3
12 2012-07-01 16227.9
13 2012-04-01 16221.9
14 2012-01-01 15973.9
15 2011-10-01 15785.3
16 2011-07-01 15587.1
  
```

그림 2.3: quandl.com의 R API 사용 방법

quandl.com는 다양한 방법으로 데이터를 제공하지만, 그 데이터의 질적 수준도 상당히 높다. 다음 그림 2.4를 보면 Haver, TR, Bloomberg의 데이터보다 더 많은 종류의 데이터와 무료 데이터를 제공한다는 점을 알 수 있다.

제 2 절 제공 데이터

quandl.com에서 제공하는 데이터는 크게 다음과 같은 6개의 섹션으로 나눌 수 있다.

- STOCKS
- OPTIONS
- FUTURES
- FORECASTS
- COMMODITIES
- ECONOMICS

	Haver	TR	Bloomberg	Quandl
COVERAGE				
Core Financial Data	✗	✓	✓	✓
Core Economic Data	✓	✓	✓	✓
Long Tail Financial & Economic Data	✗	✓	✓	✓
Social and Demographic Data	✓	✓	✗	✓
Free Data	✗	✗	✗	✓
Open Data	✗	✗	✗	✓
DELIVERY				
Download in Any Format	✗	✗	✗	✓
Unlimited API Use	✗	✗	✗	✓
Structure and Documentation	✓	✓	✓	✓
Integrations with All Major Tools	✗	✗	✗	✓
Upload and Sell Your Data	✗	✗	✗	✓
PRICING				
Unlimited Free Data Usage	✗	✗	✗	✓

그림 2.4: quandl.com의 데이터 비교

이 6개의 섹션에 대한 것을 표로 정리하면 2016-03-20 09:40:00 현재 기준으로 표 2.1과 같다.

어떤 종류의 데이터를 제공하는지 파악하기 위해서 두 가지 방법을 사용할 수 있다. 첫 번째는 그림 2.5처럼 섹션별 제공 데이터를 파악하는 방법이다. 두 번째 방법은 그림 2.6처럼 모든 데이터를 검색하는 방법이다.

Browse hundreds of free and premium financial and economic databases.

STOCKS	OPTIONS	FUTURES	FORECASTS	COMMODITIES	ECONOMICS
 DATABASE Federal Reserve Economic Data	DATASETS 194,819	DOWNLOADS 106,904,625			FREE
Growth, employment, inflation, labor, manufacturing and other US economic statistics from the research department of the Federal Reserve Bank of St. Louis.					
 DATABASE Trading Economics	DATASETS 7,382	DOWNLOADS 14,578,408			PREMIUM
Comprehensive, harmonized macroeconomic statistics for over 200 countries. 7000+ indicators from 1000+ sources, updated hourly.					
 DATABASE Global Yield Curves	DATASETS 330	DOWNLOADS 1,454,265			PREMIUM
Government bond yield curves from 35 countries, representing the entire term structure of interest rates. Updated daily.					

그림 2.5: 섹션별 제공하는 데이터 파악하는 화면

섹션	데이터베이스 이름	데이터 세트 개수	다운로드 수	서비스구분
STOCKS	Core US Fundamentals Data	2,548,946	13,695,037,498	Premium
	End of Day US Stock Prices	9,404	797,815,158	Premium
	Chinese Stock Prices	2,977	2,483,960	Premium
OPTIONS	ORATS Option Volatility Surfaces	5,632	4,984,042	Premium
	OptionWorks Futures Options	16,548	4,290,858	Premium
	US Equity Historical & Option Implied Volatilities	4,899	4,423,972	Premium
FUTURES	Chicago Mercantile Exchange Futures Data	31,942	79,113,311	Free
	Stevens Continuous Futures	2,562	6,567,796	Premium
	Stevens Reference Futures	11,617	9,702,440	Premium
FORECASTS	Zacks Earnings Estimates	12,415	840,787,477	Premium
	Zacks Sales Estimates	8,624	578,008,069	Premium
	PsychSignal Social Media Sentiment Data	53,455	49,984,599	Premium
COMMODITIES	London Bullion Market Association	4	2,336,531	Free
	U.S. Energy Information Administration Data	1,424,437	166,766,578	Free
	Shanghai Futures Exchange	1,302	1,489,618	Free
ECONOMICS	Federal Reserve Economic Data	194,819	106,904,625	Free
	Trading Economics	7,382	14,577,650	Premium
	Global Yield Curves	330	1,454,265	Premium

표 2.1: quandl.com에서 제공하는 데이터 종류

The screenshot shows the Quandl Database Browser interface. At the top, there's a search bar and a navigation menu with links to 'TOOLS', 'SOLUTIONS', 'RESOURCES', 'API DOCS', 'SELL', 'LOG IN', and 'SIGN UP'. Below the header, the URL is 'Database Browser / Stock Data / Rest of World / Benchmark Stock Indexes'. The main content area is divided into several sections: 'Stock Data' (Prices, Fundamentals, Forecasts, Sentiment, Ratings, Options and Indexes), 'Futures Data' (Prices, Options, Commitment of Traders, Continuous Contracts), 'Commodity Data' (Prices, Production, Consumption, Futures, Options), 'Currency Data' (Exchange Rates, Futures, Options, Bitcoin), 'Interest Rate Data' (Government Bond Yields, Corporate Bonds and Mortgage Rates), 'Option Data' (Option Prices, Volatility Surfaces and Distribution Data), 'Asset Management and Fund Data' (Prices and Investment Flows), and 'Index Data' (Asset Price Indexes, Economic Indexes and Volatility Indexes). To the right, there are three detailed views of specific databases: 'Barchart Global Index Prices' (Premium), 'NASDAQ OMX Global Index Data' (Free), and 'Nikkei' (Free). Each view shows the number of datasets and downloads.

그림 2.6: 모든 제공 데이터를 탐색하며 파악하는 화면

제 3 절 시카고 상품 거래소 선물 데이터 가져오기

시카고 상품 거래소 선물 데이터(Chicago Mercantile Exchange Futures Data)는 code name이 CME로 31,942개의 데이터셋을 제공한다. 이것은 상품 (금속, 곡물, 에너지) 및 금융 상품 (주식, 통화, 금리)의 광범위한 선물 데이터다.

3.1 Code Formats

선물 계약정보를 가져오는 Quandl 코드 포맷은 CME/{CODE}{MONTH}{YEAR}로 각 파트의 의미는 다음과 같다.

- CODE : 선물 시세 코드
- MONTH : 단일 문자 개월 코드

- YEAR : 4 자리 연도

- F: January
- G: February
- H: March
- J: April
- K: May
- M: June
- N: July
- Q: August
- U: September
- V: October
- X: November
- Z: December

시카고 상품 거래소(CME)에서 사용 가능한 모든 선물의 목록에 대한 코드는 “<https://s3.amazonaws.com/quandl-static-content/Ticker+CSV%27s/Futures/CMEGroup.csv>”에서 CSV 파일로 다운로드 받을 수 있다. 그리고 이 파일의 레이아웃은 Symbol, Exchange, Name, Contract Months, Quandl Code이다. 그러므로 다음과 같은 코드로 얻을 수 있다.

```
> ## get code group
> url_name <- "https://s3.amazonaws.com/quandl-static-content/Ticker+CSV%27s/Futures/CMEGroup.csv"
> cme_codes <- read.csv(url_name, header = FALSE)
> names(cme_codes) <- c("symbol", "exchange", "name", "contract_months", "quandl_code")
>
> dim(cme_codes)

[1] 308   5

> head(cme_codes)

  symbol exchange           name contract_months quandl_code
1     IBV      CME    Ibovespa Futures        GJMQVZ    CME/IBV
2     EM       CME 1-month Eurodollar Futures    FGHJKMNQUVXZ    CME/EM
3     EY       CME    EuroYen Futures         HMUZ        CME/EY
4     ED       CME    Eurodollar Futures        HJKMNQUZ    CME/ED
5    CPO      CME   Crude Palm Oil Futures    FHKMNQUX    CME/CPO
6     ES       CME   E-mini S&P 500 Futures         HMUZ    CME/ES
```

현재 사용 가능한 코드의 개수는 308개 이다.

3.2 데이터 가져오기

시카고 상품 거래소의 선물 데이터를 가져오는 방법은 “<https://www.quandl.com/api/v1/datasets/Quandl코드.csv>”의 URL을 호출하면 된다. R에서 CSV 파일을 읽는 read.csv() 함수가 인터넷 URL로 정의한 파일도 읽을 수 있으므로 간단하게 작업할 수 있다.

3.2.1 가능한 계약월 가져오기

가능한 계약월 가져오는 사용자 정의함수 getContractMonths()를 다음과 같이 작성하였다. 사용 가능한 코드 중에서 첫번째인 “bovespa Futures”에 대한 가능한 계약월은 2, 4, 6, 8, 10, 12월임을 알 수 있다.

```
> ## get data
> getContractMonths <- function(symbol = "IBV") {
+   months <- c("F", "G", "H", "J", "K", "M", "N", "Q", "U", "V", "X", "Z")
+
+   cm <- cme_codes[cme_codes == symbol, "contract_months"]
+   cm <- as.character(cm)
+   cm <- rawToChar(charToRaw(cm), multiple = TRUE)
+
+   which(months %in% cm)
+ }
> getContractMonths()

[1] 2 4 6 8 10 12
```

3.2.2 선물 데이터 가져오기

선물 데이터를 가져오는 사용자 정의함수 getCMEData()를 다음과 같이 작성하였다. 사용 가능한 코드 중에서 첫번째인 “bovespa Futures”에 대한 2015년 2월의 데이터와 옥수수 선물 계약인 “Corn Futures”의 2013년 3월 데이터를 가져왔다.

```
> ## use defined function
> getCMEData <- function(symbol = "IBV", month = getContractMonths(symbol)[1], year = "2015") {
+   url_prefix <- "https://www.quandl.com/api/v1/datasets/CME/"

+   months <- c("F", "G", "H", "J", "K", "M", "N", "Q", "U", "V", "X", "Z")

+   code <- paste0(symbol, months[month], year)

+   url_name <- paste0(url_prefix, code, ".csv")

+   read.csv(url_name, header = TRUE)
+
+ }
>
> IVB201502 <- getCMEData()
>
> cme_codes[12, ]

  symbol exchange      name contract_months quandl_code
12       C      CBT Corn Futures           HKNUZ      CME/C

> C201303 <- getCMEData("C", 3, "2013")
```

3.2.3 선물 데이터 구조보기

```
> ## dimension
> dim(C201303)

[1] 535   9

> ## variable names
> names(C201303)

[1] "Date"           "Open"          "High"          "Low"           "Last"          "Change"
[4] "Low"            "Last"          "Change"        "Settle"        "Volume"        "Prev..Day.Open.Interest"

> ## few data
> head(C201303)

      Date Open  High  Low Last Change Settle Volume Prev..Day.Open.Interest
1 2013-03-14 742.00 749.00 730.0 NA    NA 732.75    890                 737
2 2013-03-13 741.00 744.00 730.0 NA    NA 741.25   1891                1890
3 2013-03-12 733.25 743.25 730.0 NA    NA 741.00   1814               3189
4 2013-03-11 726.50 736.75 725.5 NA    NA 734.50   1631               4281
5 2013-03-08 712.75 725.75 711.5 NA    NA 725.25   1741               5464
6 2013-03-07 707.75 713.50 704.0 NA    NA 711.50   2109               6414

> ## summary data
> summary(C201303)

      Date           Open           High           Low           Last
2010-12-14: 1  Min. : 0.0  Min. : 0.0  Min. : 0.0  Mode:logical
2010-12-15: 1  1st Qu.: 0.0  1st Qu.:566.6  1st Qu.:558.4  NA's:535
2010-12-16: 1  Median :575.0  Median :606.8  Median :599.5
2010-12-17: 1  Mean   :461.1  Mean   :637.2  Mean   :625.8
2010-12-20: 1  3rd Qu.:721.2  3rd Qu.:728.8  3rd Qu.:714.0
2010-12-21: 1  Max.  :836.5  Max.  :846.2  Max.  :827.5
(Other)   :529

      Change          Settle          Volume          Prev..Day.Open.Interest
Mode:logical  Min. :501.8  Min. : 0.0  Min. :     0
NA's:535      1st Qu.:562.1  1st Qu.: 68.5  1st Qu.: 12312
                  Median :604.5  Median : 1590.0  Median : 42936
                  Mean   :632.9  Mean   : 25196.9  Mean   :123035
                  3rd Qu.:721.5  3rd Qu.: 25057.5  3rd Qu.:155589
                  Max.  :837.8  Max.  :308268.0  Max.  :649137
```

3.3 메타 데이터 가져오기

3.3.1 상세 메타 데이터 가져오기

선물 데이터의 메타 데이터는 데이터셋 아이디, 데이터셋 이름, 시작일자, 종료일자, 마지막으로 업데이트된 일자, 사용 가능한 열 등의 정보를 담고 있다. 그리고 다음과 같은 URL을 이용해서 가져온다. 이 URL은 메타 데이터를 CSV 파일로 가져온다. 만약에 “datasets.csv” 부분에서 “csv” 대신 “json”, “xml”을 사용하면 각각 JSON과 XML 포맷의 파일을 가져오게 된다.

“https://www.quandl.com/api/v3/datasets.csv?database_code=CME&per_page=100&sort_by=id&page=1&”이 호출의 의미는 다음과 같다.

- database_code=CME : 데이터베이스 코드가 CME로 시카고선물거래소의 데이터
- per_page=100 : 호출당 100개의 데이터 조회
- sort_by=id : 조회 데이터를 정렬할 변수의 이름
- page=1 : 호출 페이지의 번호가 1번
- api_key=XXXXXXXXXXXXXX : Quandl에서 제공하는 API Key
 - 회원 가입 후 받는 API Key를 기술
 - 실제로 XXXXXXXXXXXXXXXX를 입력하는 것이 아님

선물 데이터의 메타 데이터 개수는 31,942건이므로 메타 정보의 상세 조회를 위한 API 호출을 320회를 수행해야 모든 정보를 얻을 수 있다.

첫 페이지 100건의 메타 데이터를 가져와 보자.

```
> url_name <- "https://www.quandl.com/api/v3/datasets.csv?database_code=CME&per_page=100&sort_by=id&page=1"
>
> detail_meta <- read.csv(url_name)

> ## dimension
> dim(detail_meta)

[1] 100 13

> ## variable names
> names(detail_meta)

[1] "id"                  "dataset_code"        "database_code"
[4] "name"                 "description"         "refreshed_at"
[7] "newest_available_date" "oldest_available_date" "column_names"
[10] "frequency"           "type"                "premium"
[13] "database_id"

> ## a few datas
> head(detail_meta, n = 3)
```

```

      id dataset_code database_code
1 10048477      KWZ2000        CME
2 10048478      NDM2008        CME
3 10048479      NDZ2000        CME

                           name
1 KC Hard Red Winter (HRW) Wheat Futures, December 2000, KWZ2000, KCBT
2                               NASDAQ-100 Futures, June 2008, NDM2008, CME
3                               NASDAQ-100 Futures, December 2000, NDZ2000, CME

1 Historical Futures Prices: KC Hard Red Winter (HRW) Wheat Futures, December 2000, KWZ2000, Kansas City Board of Trade. Futures prices are quoted in cents per bushel. Futures prices are delivered at a discount to contract price. Protein levels of less than 10.5% are not deliverable. Pricing Unit: Cents and Quarters
2
3

      refreshed_at newest_available_date oldest_available_date
1 2014-02-21 17:20:58 UTC          2000-12-19        1999-09-07
2 2014-02-21 17:20:42 UTC          2008-06-20        2007-06-22
3 2014-02-21 17:20:55 UTC          2000-12-14        2000-03-20

      column_names
1 ["Date", "Open", "High", "Low", "Last", "Change", "Settle", "Volume", "Prev. Day Open Interest"]
2 ["Date", "Open", "High", "Low", "Last", "Change", "Settle", "Volume", "Prev. Day Open Interest"]
3 ["Date", "Open", "High", "Low", "Last", "Change", "Settle", "Volume", "Prev. Day Open Interest"]

      frequency      type premium database_id
1   daily Time Series    false      5896
2   daily Time Series    false      5896
3   daily Time Series    false      5896

> ## summary data
> summary(detail_meta)

      id      dataset_code database_code
Min. :10048477  KWH1977: 1    CME:100
1st Qu.:10048608  KWH1981: 1
Median :10048680  KWH1986: 1
Mean   :10048677  KWH1988: 1
3rd Qu.:10048755  KWH1990: 1
Max.   :10048828  KWH1993: 1
                (Other):94

                           name
KC Hard Red Winter (HRW) Wheat Futures, December 1977, KWZ1977, KCBT: 1
KC Hard Red Winter (HRW) Wheat Futures, December 1980, KWZ1980, KCBT: 1
KC Hard Red Winter (HRW) Wheat Futures, December 1982, KWZ1982, KCBT: 1
KC Hard Red Winter (HRW) Wheat Futures, December 1983, KWZ1983, KCBT: 1
KC Hard Red Winter (HRW) Wheat Futures, December 1985, KWZ1985, KCBT: 1
KC Hard Red Winter (HRW) Wheat Futures, December 1987, KWZ1987, KCBT: 1
(Other)                                         :94

```

```

Historical CBT futures prices: KC HRW Wheat Futures, Contract Size: One contract (5,000 bushels) of de
세 expense, request in the written load-out instructions that the wheat contain no more than 2 ppm of de
Historical CME Futures Prices: NASDAQ-100 Futures, December 1999, NDZ1999, Chicago Mercantile Exchange
Historical CME Futures Prices: NASDAQ-100 Futures, December 2000, NDZ2000, Chicago Mercantile Exchange
Historical CME Futures Prices: NASDAQ-100 Futures, December 2001, NDZ2001, Chicago Mercantile Exchange
Historical CME Futures Prices: NASDAQ-100 Futures, December 2005, NDZ2005, Chicago Mercantile Exchange
Historical CME Futures Prices: NASDAQ-100 Futures, December 2007, NDZ2007, Chicago Mercantile Exchange
(Other)

refreshed_at newest_available_date oldest_available_date
2014-02-21 17:21:41 UTC: 5    1977-03-22: 1      1994-07-27: 2
2014-02-21 17:21:16 UTC: 4    1977-09-21: 1      2010-11-01: 2
2014-02-21 17:21:18 UTC: 4    1977-12-20: 1      1976-04-09: 1
2014-02-21 17:21:24 UTC: 4    1979-05-21: 1      1976-10-05: 1
2014-02-21 17:21:26 UTC: 4    1979-07-20: 1      1977-01-06: 1
2014-02-21 17:21:28 UTC: 4    1980-05-20: 1      1978-05-16: 1
(Other)                      :75      (Other)     :94      (Other)     :92

column_names
["Date", "Open", "High", "Low", "Last", "Change", "Settle", "Volume", "Open Interest"]           : 1
["Date", "Open", "High", "Low", "Last", "Change", "Settle", "Volume", "Prev. Day Open Interest"]]:99

frequency          type      premium   database_id
daily:100    Time Series:100  false:100  Min.      :5896
                           1st Qu.:5896
                           Median   :5896
                           Mean     :5896
                           3rd Qu.:5896
                           Max.     :5896

```

3.3.2 모든 코드의 메타 데이터 가져오기

모든 코드의 메타 데이터는 다음의 호출을 이용하여 가져온다.

“https://www.quandl.com/api/v3/databases/CME/codes?api_key=XXXXXXXXXXXXXX” URL을 호출하면 CSV 파일이 들어있는 ZIP 압축 파일을 다운로드하며, 파일에는 데이터셋의 코드와 데이터셋의 이름 등의 정보를 포함하고 있다.

```

> url_name <- "https://www.quandl.com/api/v3/databases/CME/codes?api_key=XXXXXXXXXXXXXX"
>
> path <- "./datas"
> zip_fname <- paste(path, "CME-datasets-codes.zip", sep = "/")
> csv_fname <- sub("zip", "csv", zip_fname)

```

```
>
> ## download using URL
> download.file(url_name, zip_fname, method = "libcurl")
>
> ## unzip file
> unzip(zip_fname, exdir = path)
>
> CME_datasets <- read.csv(csv_fname, header = FALSE)
```

변수 이름 정의 및 데이터 구조 살펴 보기

```
> names(CME_datasets) <- c("datatset_id", "datatset_name")
>
> dim(CME_datasets)

[1] 32015      2

> head(CME_datasets, n = 3)

  datatset_id          datatset_name
1 CME/OH2014          Oats Futures, March 2014, OH2014, CBOT
2 CME/HGG2004          Copper Futures, February 2004, HGG2004, COMEX
3 CME/BRH2014 Brazilian Real (BRL/USD) Futures, March 2014, BRH2014, CME
```

제 4 절 런던 금괴 시장 협회 데이터 가져오기

런던 금괴 시장 협회는 중앙 은행, 개인 투자자, 생산자, 정제자 및 다른 에이전트로 구성된 런던의 금과 은 시장에 있는 국제 무역협회다. 런던 금괴 시장 협회(London Bullion Market Association)의 거래 데이터는 code name이 LBMA로 4개의 데이터셋을 제공한다.

4.1 Code Formats

런던 금괴 시장 협회 거래 데이터를 가져오는 Quandl 코드 포맷은 LBMA/{CODE}로 CODE의 의미는 다음과 같다.

- GOLD : 금 거래 정보. Gold Price
- SILVER : 은 거래 정보. Silver Price
- GOFO : Gold Forward Offered Rate
- DAILY : London Gold Fixings (1991-1999)

4.2 데이터 가져오기

4.2.1 거래 데이터 가져오기

Quandl 패키지의 Quandl() 함수를 이용해서 데이터를 가져온다. code 인수는 Quandl 코드 포맷을 사용한다.

```
> library(Quandl)
>
> ## get data
> gold <- Quandl(code = "LBMA/GOLD", api_key = "XXXXXXXXXXXXXXXXXX")
> silver <- Quandl(code = "LBMA/SILVER", api_key = "XXXXXXXXXXXXXXXXXX")
> gofo <- Quandl(code = "LBMA/GOLD", api_key = "XXXXXXXXXXXXXXXXXX")
> daily <- Quandl(code = "LBMA/GOLD", api_key = "XXXXXXXXXXXXXXXXXX")
```

4.2.2 거래 데이터 살펴보기

```
> ## dimensiond
> dim(gold)

[1] 12207      7

> dim(silver)

[1] 12218      4

> dim(gifo)

[1] 6457     16

> dim(daily)

[1] 2274      5

> ## variable names
> names(gold)

[1] "Date"       "USD (AM)"   "USD (PM)"   "GBP (AM)"   "GBP (PM)"   "EURO (AM)" "EURO (PM)"

> names(silver)

[1] "Date" "USD"  "GBP"  "EURO"

> names(gifo)

[1] "Date"           "GOFO - 1 Month"      "GOFO - 2 Months"
[4] "GOFO - 3 Months"    "GOFO - 6 Months"      "GOFO - 12 Months"
[7] "LIBOR - 1 Month"    "LIBOR - 2 Months"      "LIBOR - 3 Months"
[10] "LIBOR - 6 Months"   "LIBOR - 12 Months"     "LIBOR-GOFO - 1 Month"
[13] "LIBOR-GOFO - 2 Months" "LIBOR-GOFO - 3 Months" "LIBOR-GOFO - 6 Months"
[16] "LIBOR-GOFO - 12 Months"

> names(daily)

[1] "Date"    "AM USD"  "AM GBP"  "PM USD"  "PM GBP"

> ## a few datas
> head(gold, n = 3)
```

```

      Date USD (AM) USD (PM) GBP (AM) GBP (PM) EURO (AM) EURO (PM)
1 2016-04-19 1241.70 1255.4 867.007 870.753 1095.183 1103.082
2 2016-04-18 1237.70 1234.3 872.450 868.502 1095.016 1091.732
3 2016-04-15 1229.75 1227.1 867.457 865.595 1092.164 1086.395

> head(silver, n = 3)

      Date USD GBP EURO
1 2016-04-19 16.62 11.5674 14.6651
2 2016-04-18 16.20 11.4093 14.3287
3 2016-04-15 16.17 11.3978 14.3338

> head(gofo, n = 3)

      Date GOFO - 1 Month GOFO - 2 Months GOFO - 3 Months GOFO - 6 Months
1 2015-01-30 0.065 0.0825 0.0950 0.1175
2 2015-01-29 0.090 0.1000 0.1100 0.1200
3 2015-01-28 0.085 0.0975 0.1075 0.1250

      GOFO - 12 Months LIBOR - 1 Month LIBOR - 2 Months LIBOR - 3 Months LIBOR - 6 Months
1 0.1525 0.17125 0.2104 0.2531 0.3574
2 0.1575 0.17090 0.2106 0.2546 0.3579
3 0.1600 0.17050 0.2115 0.2546 0.3574

      LIBOR - 12 Months LIBOR-GOFO - 1 Month LIBOR-GOFO - 2 Months LIBOR-GOFO - 3 Months
1 0.6209 NA NA NA
2 0.6204 0.0809 0.1106 0.1446
3 0.6179 0.0855 0.1140 0.1471

      LIBOR-GOFO - 6 Months LIBOR-GOFO - 12 Months
1 NA NA
2 0.2379 0.4629
3 0.2324 0.4579

> head(daily, n = 3)

      Date AM USD AM GBP PM USD PM GBP
1 1999-12-30 290.85 180.093 290.25 179.61
2 1999-12-29 290.65 179.691 290.90 179.79
3 1999-12-24 287.80 178.260 NA NA

```

4.3 메타 데이터 가져오기

4.3.1 상세 메타 데이터 가져오기

선물 데이터의 메타 데이터를 가져오는 것과 동일한 방법으로 수행한다. 다만 database_code 값을 LBMA로 설정한다. 선물 데이터의 메타 데이터 개수는 4건이므로 메타 정보의 상세 조회를 위한 API 호출은 한번으로 족하다.

4건의 메타 데이터를 가져와 보자.

```

> url_name <- "https://www.quandl.com/api/v3/datasets.csv?database_code=LBMA&per_page=100&sort_by=id&page=1"
>
> detail_meta <- read.csv(url_name)

> ## dimension
> dim(detail_meta)

[1] 4 13

> ## variable names
> names(detail_meta)

[1] "id"                  "dataset_code"      "database_code"
[4] "name"                 "description"       "refreshed_at"
[7] "newest_available_date" "oldest_available_date" "column_names"
[10] "frequency"           "type"             "premium"
[13] "database_id"

> ## a few datas
> detail_meta

      id dataset_code database_code          name
1 11304013      SILVER        LBMA Silver Price: London Fixing
2 11304018       GOFO        LBMA Gold Forward Offered Rates (GOFO)
3 11304240       GOLD        LBMA    Gold Price: London Fixing
4   502594      DAILY        LBMA  London Gold Fixings (1991-1999)

1 Silver Price: London Fixing. London Bullion Market Association (LBMA). Fixing levels are set per troy
2
3
4

      refreshed_at newest_available_date oldest_available_date
1 2016-04-19 12:27:17 UTC      2016-04-19      1968-01-02
2 2015-05-13 13:50:23 UTC      2015-01-30      1989-07-17
3 2016-04-19 15:50:23 UTC      2016-04-19      1968-01-02
4 2013-08-27 12:36:16 UTC      1999-12-30      1991-01-02

1
2 ["Date", "GOFO - 1 Month", "GOFO - 2 Months", "GOFO - 3 Months", "GOFO - 6 Months", "GOFO - 12 Months"]
3
4

      frequency      type premium database_id
1     daily Time Series  false      139
2     daily Time Series  false      139
3     daily Time Series  false      139
4     daily Time Series  false      139

```

4.3.2 모든 코드의 메타 데이터 가져오기

모든 코드의 메타 데이터는 다음의 호출을 이용하여 가져온다.

"https://www.quandl.com/api/v3/databases/LBMA/codes?api_key=XXXXXXXXXXXXXX"
 이 URL을 호출하면 CSV 파일이 들어있는 ZIP 압축 파일을 다운로드하며, 파일에는 데이터셋의 코드와 데이터셋의 이름 등의 정보를 포함하고 있다.

```
> url_name <- "https://www.quandl.com/api/v3/databases/LBMA/codes?api_key=XXXXXXXXXXXXXX"
>
> path <- "./datas"
> zip_fname <- paste(path, "LBMA-datasets-codes.zip", sep = "/")
> csv_fname <- sub("zip", "csv", zip_fname)
>
> ## download using URL
> download.file(url_name, zip_fname, method = "libcurl", mode = "wb")
>
> ## unzip file
> unzip(zip_fname, exdir = path)
>
> LBMA_datasets <- read.csv(csv_fname, header = FALSE)
```

변수 이름 정의 및 데이터 구조 살펴 보기

```
> names(LBMA_datasets) <- c("dataset_id", "dataset_name")
>
> dim(LBMA_datasets)

[1] 4 2

> LBMA_datasets

  dataset_id          dataset_name
1 LBMA/GOLD      Gold Price: London Fixing
2 LBMA/SILVER    Silver Price: London Fixing
3 LBMA/DAILY  London Gold Fixings (1991-1999)
4 LBMA/GOFO Gold Forward Offered Rates (GOFO)
```

제 5 절 미국 에너지 정보 관리국 데이터 가져오기

미국 에너지 정보 관리국(U.S. Energy Information Administration)은 소비, 생산, 수입, 수출 데이터, 전기, 석탄, 천연 가스, 석유 등 주요 에너지 제품에 대한 다양한 지표를 제공한다. 데이터는 월, 분기, 연간 기준으로 주와 국가에 보고된다. 미국 에너지 정보 관리국(EIA) 데이터베이스는 5 가지의 컴포넌트(component)로 구성되어 있으며, 각각의 구성요소는 다음의 표 2.3과 같다.

컴포넌트		컴포넌트 코드
State Energy Data Systems		SEDS
Coal Data		COAL
Petroleum Data		PET
Natural Gas Data		NG
Electricity Data		ELEC

표 2.2: EIA 데이터베이스 종류

5.1 Code Formats

미국 에너지 정보 관리국 데이터를 가져오는 Quandl 코드 포맷은 EIA/{COMPONENT}-{INDICATOR}-{FREQUENCY}로 각 파트의 의미는 다음과 같다.

- EIA : main 데이터베이스 코드
- COMPONENT : 컴포넌트 코드
 - SEDS
 - COAL
 - PET
 - NG
 - ELEC
- INDICATOR : 인디케이터 코드
- FREQUENCY : 데이터 생성 주기
 - A : 년
 - Q : 분기
 - M : 월

Quandl 코드 포맷의 예제는 다음의 표 2.3과 같다.

Full Code		Component	Indicator		Frequency
EIA/SEDS.PAPRB.KS_A		SEDS	PAPP RB_KS: Crude oil production in Kansas		A: Annual
EIA/ELEC_PLANT_GEN_54249_WAT_HY_Q		ELEC	PLANT_GEN_54249_WAT_HY: Smith Falls Hydro Project (Net Generation of Conventional Hydraulic Turbines)		Q: Quarterly
EIA/COAL_MARKET_SALES_MD_OPM_A		COAL	MARKET_SALES_MD_OPM_A: Market Sales : Maryland : Open Market		A: Annual
EIA/NG_RNGC3_M		NG	RNGC3: Natural Gas Futures Contract 3		M: Monthly
EIA/PET_MCRFPFL1_M		PET	MCRFPFL1: Florida Field Production of Crude Oil		M: Monthly

표 2.3: EIA 데이터베이스 종류

5개의 컴포넌트에 대한 인디케이터 코드는 다음의 URL 호출을 통해 얻을 수 있다.

- State Energy Data Systems : 국가 에너지 데이터 시스템

- https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_SEDS_codes.txt
- Coal Data : 석탄 데이터
 - https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_COAL_codes.txt
- Petroleum : 석유 데이터
 - https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_PET_codes.txt
- Natural Gas : 천연가스 데이터
 - https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_NG_codes.txt
- Electricity Data : 전기 데이터
 - https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_ELEC_codes.txt

5.1.1 인디케이터 코드 가져오기

```
> ## get indicator codes
> SEDS <- "https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_SEDS_codes.txt"
> COAL <- "https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_COAL_codes.txt"
> PET <- "https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_PET_codes.txt"
> NG <- "https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_NG_codes.txt"
> ELEC <- "https://s3.amazonaws.com/quandl-static-content/EIA+codes/EIA_ELEC_codes.txt"
>
> SEDS_codes <- read.csv(SEDS, header = FALSE, sep = "^", stringsAsFactors = FALSE)
> SEDS_codes <- do.call("rbind", strsplit(SEDS_codes[, 1], "::"))
> SEDS_codes <- as.data.frame(SEDS_codes)
> names(SEDS_codes) <- c("indicator", "name", "unit")
>
> COAL_codes <- read.csv(COAL, header = FALSE, sep = "^", stringsAsFactors = FALSE)
> COAL_codes <- do.call("rbind", strsplit(COAL_codes[, 1], "::"))
> COAL_codes <- as.data.frame(COAL_codes)
> names(COAL_codes) <- c("indicator", "name", "unit")
>
> PET_codes <- read.csv(PET, header = FALSE, sep = "^", stringsAsFactors = FALSE)
> PET_codes <- do.call("rbind", strsplit(PET_codes[, 1], "::"))
>
> PET_codes[PET_codes[, 1] == PET_codes[, 3], 3] <- ""
> PET_codes <- as.data.frame(PET_codes)
> names(PET_codes) <- c("indicator", "name", "unit")
>
> NG_codes <- read.csv(NG, header = FALSE, sep = "^", stringsAsFactors = FALSE)
> NG_codes <- do.call("rbind", strsplit(NG_codes[, 1], "::"))
> NG_codes <- as.data.frame(NG_codes)
> names(NG_codes) <- c("indicator", "name", "unit")
>
```

```
> ELEC_codes <- read.csv(ELEC, header = FALSE, sep = "^", stringsAsFactors = FALSE)
> ELEC_codes <- do.call("rbind", strsplit(ELEC_codes[, 1], ":"))
> ELEC_codes <- as.data.frame(ELEC_codes)
> names(ELEC_codes) <- c("indicator", "name", "unit")
```

5.1.2 인디케이터 코드 살펴보기

```
> dim(SEDs_codes)
[1] 30842      3

> head(SEDs_codes, n = 3)

      indicator          name      unit
1 SEDS_TPOPP_MO_A Resident population including Armed Forces, Missouri Thousand
2 SEDS_TPOPP_MN_A Resident population including Armed Forces, Minnesota Thousand
3 SEDS_TPOPP_MI_A Resident population including Armed Forces, Michigan Thousand

> dim(COAL_codes)
[1] 204341      3

> head(COAL_codes, n = 3)

      indicator          name
1 COAL_MARKET_SALES_US_OPM_A Market Sales : United States : Open Market : Annual
2 COAL_MARKET_SALES_UT_OPM_A      Market Sales : Utah : Open Market : Annual
3 COAL_MARKET_SALES_VA_OPM_A      Market Sales : Virginia : Open Market : Annual

      unit
1 short tons
2 short tons
3 short tons

> dim(PET_codes)
[1] 458885      3

> head(PET_codes, n = 3)

      indicator          name
1 PET_MCRSFP31_A Gulf Coast (PADD 3) Crude Oil Stocks at Tank Farms, Annual
2 PET_MCRST_YCUOK_1_M      Cushing, OK Ending Stocks of Crude Oil, Monthly
3 PET_MCRSFP51_M West Coast (PADD 5) Crude Oil Stocks at Tank Farms, Monthly

      unit
1 Thousand Barrels
2 Thousand Barrels
3 Thousand Barrels

> dim(NG_codes)
```

```
[1] 31092      3

> head(NG_codes, n = 3)

  indicator          name        unit
1 NG_N9150US1_M    Balancing Item, Monthly Billion Cubic Feet
2 NG_N9090US1_M U.S. Supplemental Gaseous Fuels, Monthly Billion Cubic Feet
3 NG_N9070US1_M U.S. Dry Natural Gas Production, Monthly Billion Cubic Feet

> dim(ELEC_codes)

[1] 459473      3

> head(ELEC_codes, n = 3)

  indicator          name
1 ELEC_SALES_AK_RES_A Retail Sales of Electricity : Alaska : Residential : Annual
2 ELEC_SALES_AL_RES_A Retail Sales of Electricity : Alabama : Residential : Annual
3 ELEC_SALES_AL_RES_M Retail Sales of Electricity : Alabama : Residential : Monthly

  unit
1 Million Kilowatthours
2 Million Kilowatthours
3 Million Kilowatthours
```

현재 사용 가능한 코드의 개수는 SEDS가 30842개, COAL이 204341개, PET가 458885개, NG가 31092개, ELEC가 459473개이다.

5.2 데이터 가져오기

5.2.1 에너지 사용데이터 가져오기

“https://www.quandl.com/api/v1/datasets/EIA/SEDS_CLPRB_UT_A.csv?&trim_start=2002-01-01&trim_end=2008-12-31&rows=4”와 같은 URL을 호출을 통해 미국 에너지 정보 관리국 데이터를 가져오지만 간단하게 Quandl 패키지의 Quandl() 함수를 사용하면 된다.

Quandl에서는 1일에 50회 이내로 API 호출을 제한한다. 그러나 Quandl.api_key() 함수로 발급 받은 API key를 등록하면 50 이상 호출할 수 있게 된다.

국가 에너지 데이터 시스템의 코드 데이터인 NG_codes의 첫번째 코드에 해당하는 데이터를 가져와 보자.

```
##      indicator          name        unit
## 1 NG_N9150US1_M    Balancing Item, Monthly Billion Cubic Feet
```

```
> NG_codes[1, ]
>
> # API Key
> Quandl.api_key("XXXXXXXXXXXXXX")
>
> # data.frame
```

```
> NG_N9150US1_M_df <- Quandl("EIA/NG_N9150US1_M", start_date = "2011-01-31",
+     end_date = "2015-12-31")
> # xts
> NG_N9150US1_M_xts <- Quandl("EIA/NG_N9150US1_M", start_date = "2011-01-31",
+     end_date = "2015-12-31", type = "xts")
```

5.2.2 에너지 사용데이터 살펴보기

```
> # data.frame
> class(NG_N9150US1_M_df)

[1] "data.frame"

> dim(NG_N9150US1_M_df)

[1] 60  2

> head(NG_N9150US1_M_df)

      Date Value
1 2015-12-31   -27
2 2015-11-30   -50
3 2015-10-31   -42
4 2015-09-30   -23
5 2015-08-31   -15
6 2015-07-31   -19

> summary(NG_N9150US1_M_df)

      Date           Value    
Min.   :2011-01-31   Min.   :-99.000  
1st Qu.:2012-04-22   1st Qu.:-23.250  
Median :2013-07-15   Median : -0.500  
Mean   :2013-07-15   Mean   : -3.583  
3rd Qu.:2014-10-07   3rd Qu.: 17.500  
Max.   :2015-12-31   Max.   : 43.000  

> # data.frame
> dim(NG_N9150US1_M_xts)

[1] 60  1

> head(NG_N9150US1_M_xts)

[,1]
1 2011 -24
2 2011  20
3 2011  -4
4 2011  17
5 2011  -7
6 2011 -11
```

```
> summary(NG_N9150US1_M_xts)

Index      NG_N9150US1_M_xts
Min.   :2011   Min.   :-99.000
1st Qu.:2012   1st Qu.:-23.250
Median :2013   Median  :-0.500
Mean    :2013   Mean    :-3.583
3rd Qu.:2015   3rd Qu.: 17.500
Max.   :2016   Max.   : 43.000

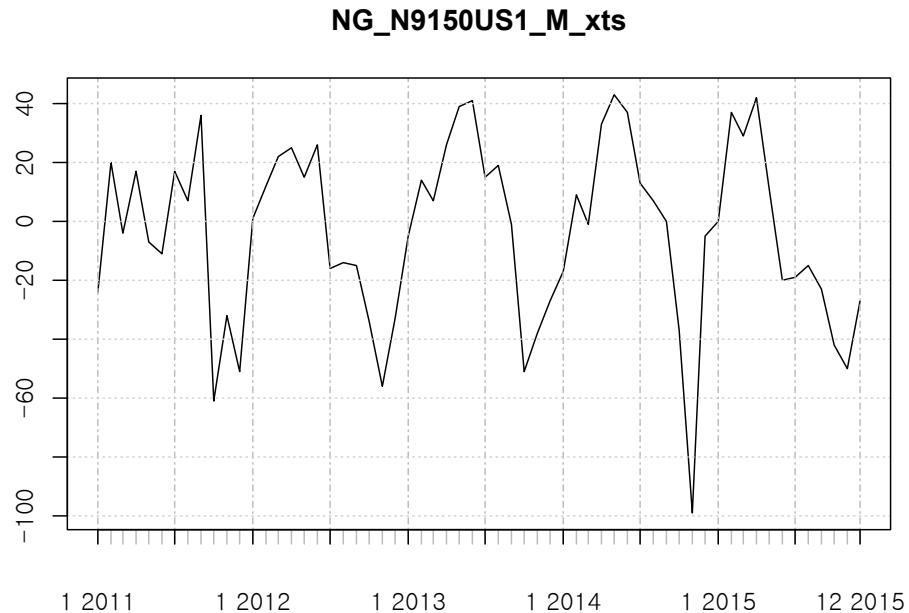
> class(NG_N9150US1_M_xts)

[1] "xts" "zoo"
```

5.2.3 에너지 사용데이터 시각화

에너지 사용데이터의 추이를 다음과 같이 시각화하면 그림 ??이 같은 플롯을 출력한다.

```
> plot(NG_N9150US1_M_xts)
```



5.3 메타 데이터 가져오기

5.3.1 상세 메타 데이터 가져오기

미국 에너지 정보 관리국 데이터의 메타 데이터는 데이터셋 아이디, 데이터셋 이름, 시작일자, 종료일자, 마지막으로 업데이트된 일자, 사용 가능한 열 등의 정보를 담고 있다. 그리고 다음과 같은 URL을 이용해서 가져온다. 이 URL은 메타 데이터를 CSV 파일로 가져온다. 만약에 “datasets.csv” 부분에서 “csv” 대신 “json”, “xml”을 사용하면 각각 JSON과 XML 포맷의 파일을 가져오게 된다.

“https://www.quandl.com/api/v3/datasets.csv?database_code=EIA&per_page=100&sort_by=id&page=1&api_key=XXXXXXXXXXXXXX”이 호출의 의미는 다음과 같다.

- database_code=EIA : 데이터베이스 코드가 EIA로 시카고선물거래소의 데이터
- per_page=100 : 호출당 100개의 데이터 조회
- sort_by=id : 조회 데이터를 정렬할 변수의 이름
- page=1 : 호출 페이지의 번호가 1번
- api_key=XXXXXXXXXXXXXX : Quandl에서 제공하는 API Key
 - 회원 가입 후 받는 API Key를 기술
 - 실제로 XXXXXXXXXXXXXXXX를 입력하는 것이 아님

선물 데이터의 메타 데이터 개수는 1,425,095건이므로 메타 정보의 상세 조회를 위한 API 호출을 14,251회를 수행해야 모든 정보를 얻을 수 있다.

첫 페이지 100건의 메타 데이터를 가져와 보자.

```
> url_name <- "https://www.quandl.com/api/v3/datasets.csv?database_code=EIA&per_page=100&sort_by=id&page=1"
>
> detail_meta <- read.csv(url_name)

> ## dimension
> dim(detail_meta)

[1] 100 13

> ## variable names
> names(detail_meta)

[1] "id"                  "dataset_code"          "database_code"
[4] "name"                 "description"           "refreshed_at"
[7] "newest_available_date" "oldest_available_date" "column_names"
[10] "frequency"            "type"                  "premium"
[13] "database_id"

> ## a few datas
> head(detail_meta, n = 3)

   id dataset_code database_code          name
1 11825462 IES_5_53_1_NOAM          EIA Total Oil Supply - North America
2 11825463 IES_5_53_1_BRM          EIA      Total Oil Supply - Bermuda
3 11825464 IES_5_53_1_CAN          EIA      Total Oil Supply - Canada

1 Units=Thousand Barrels Per Day. The U.S. Energy Information Administration (EIA) collects, analyzes, and disseminates energy information to support sound decision making across the nation's energy needs.
2 Units=Thousand Barrels Per Day. The U.S. Energy Information Administration (EIA) collects, analyzes, and disseminates energy information to support sound decision making across the nation's energy needs.
3 Units=Thousand Barrels Per Day. The U.S. Energy Information Administration (EIA) collects, analyzes, and disseminates energy information to support sound decision making across the nation's energy needs.
```

```

refreshed_at newest_available_date oldest_available_date
1 2015-01-30 19:35:37 UTC      2013-12-31      1980-12-31
2 2015-01-30 19:35:33 UTC      2013-12-31      1980-12-31
3 2015-01-30 19:35:29 UTC      2013-12-31      1980-12-31

column_names frequency type premium database_id
1 ["Year", "Thousand Barrels Per Day"] annual Time Series false 661
2 ["Year", "Thousand Barrels Per Day"] annual Time Series false 661
3 ["Year", "Thousand Barrels Per Day"] annual Time Series false 661

> ## summary
> summary(detail_meta)

id dataset_code database_code
Min. :11825462 IES_5_53_1_ALB : 1 EIA:100
1st Qu.:11825487 IES_5_53_1_ANTA: 1
Median :11825512 IES_5_53_1_ARG : 1
Mean   :11825512 IES_5_53_1_ARM : 1
3rd Qu.:11825536 IES_5_53_1_ARU : 1
Max.   :11825561 IES_5_53_1_ATG : 1
(Other)          :94

name
Total Oil Supply - Albania      : 1
Total Oil Supply - Antarctica   : 1
Total Oil Supply - Antigua and Barbuda: 1
Total Oil Supply - Argentina    : 1
Total Oil Supply - Armenia      : 1
Total Oil Supply - Aruba        : 1
(Other)                         :94

```

Units=Thousands Barrels Per Day. The U.S. Energy Information Administration (EIA) collects, analyzes, and disseminates energy information nationwide to support sound decisionmaking across all energy sectors.

```

refreshed_at newest_available_date oldest_available_date
2015-01-30 19:35:38 UTC:16 1990-12-31: 2 1980-12-31:88
2015-01-30 19:35:33 UTC:11 1991-12-31: 1 1986-12-31: 1
2015-01-30 19:35:35 UTC:10 1992-12-31: 1 1991-12-31: 1
2015-01-30 19:35:36 UTC:10 2005-12-31: 1 1992-12-31: 6
2015-01-30 19:35:30 UTC: 9 2013-12-31:95 1993-12-31: 2
2015-01-30 19:35:31 UTC: 9                      2006-12-31: 2
(Other)          :35

column_names frequency type premium
["Year", "Thousand Barrels Per Day"] :100 annual:100 Time Series:100 false:100

```

```
database_id
Min.    :661
1st Qu.:661
Median  :661
Mean    :661
3rd Qu.:661
Max.    :661
```

5.3.2 모든 코드의 메타 데이터 가져오기

모든 코드의 메타 데이터는 다음의 호출을 이용하여 가져온다.

“[“https://www.quandl.com/api/v3/databases/EIA/codes?api_key=XXXXXXXXXXXXXX”](https://www.quandl.com/api/v3/databases/EIA/codes?api_key=XXXXXXXXXXXXXX)” URL을 호출하면 CSV 파일이 들어있는 ZIP 압축 파일을 다운로드하며, 파일에는 데이터셋의 코드와 데이터셋의 이름 등의 정보를 포함하고 있다.

메타데이터의 개수가 1,425,095로 제법 커서 `read.csv()` 함수가 아니라 `readr` 패키지의 `read_csv()` 함수를 이용하였다. 이 함수는 데이터 건수가 많을 경우 발생하는 `read.csv()` 함수의 속도 저하의 문제를 해결해준다.

```
> url_name <- "https://www.quandl.com/api/v3/databases/EIA/codes?api_key=XXXXXXXXXXXXXX"
>
> path <- "./datas"
> zip_fname <- paste(path, "EIA-datasets-codes.zip", sep = "/")
> csv_fname <- sub("zip", "csv", zip_fname)
>
> ## download using URL
> download.file(url_name, zip_fname, method = "libcurl", mode = "wb")
>
> ## unzip file
> unzip(zip_fname, exdir = path)
>
> library(readr)
> EIA_datasets <- read_csv(csv_fname, col_names = FALSE)
```

변수 이름 정의 및 데이터 구조 살펴 보기

```
> names(EIA_datasets) <- c("dataset_id", "dataset_name")
>
> dim(EIA_datasets)

[1] 1425095      2
```

```
> head(EIA_datasets, n = 3)

      dataset_id
1 EIA/PET_MLUEXTH2_M
2 EIA/PET_MTXNT_NUS_NHO_2_M
3 EIA/PET_MLURPP12_A

      dataset_name
1 U.S. Exports to Thailand of Lubricants, Monthly
2 U.S. Net Imports from Honduras of Total Petroleum Products, Monthly
3 East Coast (PADD 1) Refinery and Blender Net Production of Lubricants, Annual
```

제 6 절 상하이 선물 거래소 데이터 가져오기

상하이 선물 거래소 데이터(Shanghai Futures Exchange)는 code name이 CME로 31,942개의 데이터셋을 제공한다. 이것은 상품 (금속, 곡물, 에너지) 및 금융 상품 (주식, 통화, 금리)의 광범위한 선물 데이터다.

6.1 Code Formats

선물 계약정보를 가져오는 Quandl 코드 포맷은 SHFE/{CODE}{MONTH}{YEAR}로 각 파트의 의미는 다음과 같다.

- CODE : 선물 시세 코드
- MONTH : 단일 문자 개월 코드
- YEAR : 4 자리 연도
 - F: January
 - G: February
 - H: March
 - J: April
 - K: May
 - M: June
 - N: July
 - Q: August
 - U: September
 - V: October
 - X: November
 - Z: December

상하이 선물 거래소(SHFE)에서 사용 가능한 모든 선물의 목록에 대한 코드는 “<https://s3.amazonaws.com/quandl-static-content/Ticker+CSV%27s/Futures/SHEF.csv>”에서 CSV 파일로 다운로드 받을 수 있다. 그리고 이 파일의 레이아웃은 Symbol, Exchange, Name, Contract Months, Quandl Code이다. 그러므로 다음과 같은 코드로 얻을 수 있다.

```
> ## get code group
> url_name <- "https://s3.amazonaws.com/quandl-static-content/Ticker+CSV%27s/Futures/SHEF.csv"
> shfe_codes <- read.csv(url_name, header = TRUE)
> names(shfe_codes) <- c("symbol", "exchange", "name", "contract_months", "quandl_code")
>
> dim(shfe_codes)

[1] 10 5

> head(shfe_codes)

  symbol exchange      name contract_months quandl_code
1     AG      SHFE   Silver    FGHJKMNQUVXZ    SHFE/AG
2     AL      SHFE Aluminium FGHJKMNQUVXZ    SHFE/AL
3     AU      SHFE      Gold    FGHJKMNQUVXZ    SHFE/AU
4     CU      SHFE    Copper   FGHJKMNQUVXZ    SHFE/CU
5     FU      SHFE Fuel Oil  FGHJKMNQUVXZ    SHFE/FU
6     PB      SHFE     Lead   FGHJKMNQUVXZ    SHFE/PB
```

현재 사용 가능한 코드의 개수는 10개이다.

6.2 데이터 가져오기

상하이 선물 거래소의 선물 데이터를 가져오는 방법은 “<https://www.quandl.com/api/v1/datasets/Quandl/코드.csv>”의 URL을 호출하면 된다. R에서 CSV 파일을 읽는 `read.csv()` 함수가 인터넷 URL로 정의한 파일도 읽을 수 있으므로 간단하게 작업할 수 있다.

6.2.1 가능한 계약월 가져오기

가능한 계약월 가져오는 사용자 정의함수 `getContractMonths()`를 다음과 같이 작성하였다. 사용 가능한 코드 중에서 첫번째인 “bovespa Futures”에 대한 가능한 계약월은 2, 4, 6, 8, 10, 12월임을 알 수 있다.

```
> ## get data
> getContractMonths <- function(symbol = "AG") {
+   months <- c("F", "G", "H", "J", "K", "M", "N", "Q", "U", "V", "X", "Z")
+
+   cm <- shfe_codes[shfe_codes == symbol, "contract_months"]
+   cm <- as.character(cm)
+   cm <- rawToChar(charToRaw(cm), multiple = TRUE)
+
+   which(months %in% cm)
+ }
```

```
> getHContractMonths()
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

6.2.2 선물 데이터 가져오기

선물 데이터를 가져오는 사용자 정의함수 getCMEData()를 다음과 같이 작성하였다. 사용가능한 코드 중에서 첫번째인 “bovespa Futures”에 대한 2015년 2월의 데이터와 옥수수 선물 계약인 “Corn Futures”의 2013년 3월 데이터를 가져왔다.

```
> ## use defined function
> getSHFEData <- function(symbol = "AG", month = getHContractMonths(symbol)[1], year = "2015") {
+   url_prefix <- "https://www.quandl.com/api/v1/datasets/SHFE/"

+   months <- c("F", "G", "H", "J", "K", "M", "N", "Q", "U", "V", "X", "Z")

+   code <- paste0(symbol, months[month], year)

+   url_name <- paste0(url_prefix, code, ".csv")

+   read.csv(url_name, header = TRUE)
+ }
>
> AG201502 <- getSHFEData()
>
> shfe_codes[2, ]

  symbol exchange      name contract_months quandl_code
2     AL      SHFE Aluminium      FGHJKMNQUVXZ      SHFE/AL

> AL201303 <- getSHFEData("AL", 3, "2013")
```

6.2.3 선물 데이터 구조보기

```
> ## dimension
> dim(AL201303)

[1] 242 12

> ## variable names
> names(AL201303)

[1] "Date"                  "Pre.Settle"           "Open"
[4] "High"                  "Low"                 "Close"
[7] "Settle"                "CH1"                "CH2"
[10] "Volume"               "Prev..Day.Open.Interest" "Change"
```

```
> ## a few datas
> head(AL201303)

  Date Pre.Settle Open High Low Close Settle CH1 CH2 Volume
1 2013-03-15     14595 14610 14625 14570 14600 14595   5   0    3460
2 2013-03-14     14620 14625 14635 14565 14635 14595  15 -25    3190
3 2013-03-13     14600 14600 14670 14590 14670 14620  70  20    2790
4 2013-03-12     14650 14605 14615 14585 14600 14600 -50 -50    4450
5 2013-03-11     14690 14700 14700 14620 14665 14650 -25 -40    2940
6 2013-03-08     14625 14730 14730 14655 14720 14690  95  65    4110

  Prev..Day.Open.Interest Change
1                      30810   -780
2                      31590   -850
3                      32440   -590
4                      33030    90
5                      32940   -30
6                      32970   -130

> ## summary
> summary(AL201303)

  Date      Pre.Settle        Open          High          Low
2012-03-16: 1  Min. :14410  Min. :14440  Min. :14470  Min. :14355
2012-03-19: 1  1st Qu.:15310 1st Qu.:15270 1st Qu.:15282 1st Qu.:15240
2012-03-20: 1  Median :15460 Median :15410 Median :15425 Median :15380
2012-03-21: 1  Mean   :15626 Mean   :15479 Mean   :15500 Mean   :15449
2012-03-22: 1  3rd Qu.:15932 3rd Qu.:15642 3rd Qu.:15675 3rd Qu.:15610
2012-03-23: 1  Max.  :17030 Max.  :17030 Max.  :17055 Max.  :17030
(Other)  :236 NA's  :3    NA's  :47  NA's  :47  NA's  :47

  Close       Settle        CH1          CH2          Volume
Min. :14415  Min. :14410  Min. :-405.000  Min. :-250.000  Min. :    2
1st Qu.:15304 1st Qu.:15306 1st Qu.:-50.000 1st Qu.:-45.000 1st Qu.:    17
Median :15458 Median :15458 Median :  0.000 Median :  0.000 Median : 1272
Mean   :15623 Mean   :15622 Mean   : -9.146 Mean   : -9.771 Mean   : 2977
3rd Qu.:15929 3rd Qu.:15925 3rd Qu.: 26.250 3rd Qu.: 20.000 3rd Qu.: 4849
Max.  :17055  Max. :17030  Max. :265.000  Max. :265.000  Max. :20090
NA's  :2        NA's  :2    NA's  :2    NA's  :2    NA's  :47

  Prev..Day.Open.Interest      Change
Min. :    6.0      Min. : -3366.0
1st Qu.: 108.5    1st Qu.:   -2.5
Median : 2130.0    Median :    0.0
Mean   :18346.0    Mean   : 119.8
3rd Qu.:35896.0    3rd Qu.: 138.5
Max.  :65636.0    Max.   : 4370.0
NA's  :2            NA's  :2
```

6.3 메타 데이터 가져오기

6.3.1 상세 메타 데이터 가져오기

선물 데이터의 메타 데이터는 데이터셋 아이디, 데이터셋 이름, 시작일자, 종료일자, 마지막으로 업데이트된 일자, 사용 가능한 열 등의 정보를 담고 있다. 그리고 다음과 같은 URL을 이용해서 가져온다. 이 URL은 메타 데이터를 CSV 파일로 가져온다. 만약에 “datasets.csv” 부분에서 “csv” 대신 “json”, “xml”을 사용하면 각각 JSON과 XML 포맷의 파일을 가져오게 된다.

“https://www.quandl.com/api/v3/datasets.csv?database_code=SHFE&per_page=100&sort_by=id&page=1&api_key=XXXXXXXXXXXXXX”
이 호출의 의미는 다음과 같다.

- database_code=SHFE : 데이터베이스 코드가 SHFE로 상하이 선물거래소의 데이터
- per_page=100 : 호출당 100개의 데이터 조회
- sort_by=id : 조회 데이터를 정렬할 변수의 이름
- page=1 : 호출 페이지의 번호가 1번
- api_key=XXXXXXXXXXXXXX : Quandl에서 제공하는 API Key
 - 회원 가입 후 받는 API Key를 기술
 - 실제로 XXXXXXXXXXXXXXXX를 입력하는 것이 아님

선물 데이터의 메타 데이터 개수는 31,942건이므로 메타 정보의 상세 조회를 위한 API 호출을 320회를 수행해야 모든 정보를 얻을 수 있다.

첫 페이지 100건의 메타 데이터를 가져와 보자.

```
> url_name <- "https://www.quandl.com/api/v3/datasets.csv?database_code=SHFE&per_page=100&sort_by=id&page=1&api_key=XXXXXXXXXXXXXX"
>
> detail_meta <- read.csv(url_name)

> ## dimension
> dim(detail_meta)

[1] 100 13

> ## variable names
> names(detail_meta)

[1] "id"                  "dataset_code"          "database_code"
[4] "name"                "description"          "refreshed_at"
[7] "newest_available_date" "oldest_available_date" "column_names"
[10] "frequency"           "type"                 "premium"
[13] "database_id"

> ## a few datas
> head(detail_meta, n = 3)
```

```

      id dataset_code database_code          name
1 10113116      CUG2015        SHFE  Shanghai Copper Futures, February 2015 (CUG2015)
2 10113117      ALG2015        SHFE  Shanghai Aluminium Futures, February 2015 (ALG2015)
3 10113118      ZNG2015        SHFE  Shanghai Zinc Futures, February 2015 (ZNG2015)

                                         description
1   Historical SHFE Futures Prices: Shanghai Copper Futures, February 2015 (CUG2015)
2 Historical SHFE Futures Prices: Shanghai Aluminium Futures, February 2015 (ALG2015)
3   Historical SHFE Futures Prices: Shanghai Zinc Futures, February 2015 (ZNG2015)

      refreshed_at newest_available_date oldest_available_date
1 2015-06-25 17:04:14 UTC            2015-02-10    2014-02-18
2 2015-06-25 17:04:17 UTC            2015-02-10    2014-02-18
3 2015-06-25 17:04:18 UTC            2015-02-10    2014-02-18

1 ["Date", "Pre Settle", "Open", "High", "Low", "Close", "Settle", "CH1", "CH2", "Volume", "Prev. Day Op
2 ["Date", "Pre Settle", "Open", "High", "Low", "Close", "Settle", "CH1", "CH2", "Volume", "Prev. Day Op
3 ["Date", "Pre Settle", "Open", "High", "Low", "Close", "Settle", "CH1", "CH2", "Volume", "Prev. Day Op

      frequency      type premium database_id
1   daily Time Series  false      1160
2   daily Time Series  false      1160
3   daily Time Series  false      1160

> ## summary
> summary(detail_meta)

      id          dataset_code database_code
Min. :10113116 AGG2015: 1     SHFE:100
1st Qu.:11822341 AGH2015: 1
Median :13082060 AGJ2015: 1
Mean   :12891184 AGK2015: 1
3rd Qu.:13995511 AGM2015: 1
Max.   :14686013 AGN2015: 1
                           (Other):94

                                         name
Shanghai Aluminium Futures, April 2015 (ALJ2015) : 1
Shanghai Aluminium Futures, August 2015 (ALQ2015) : 1
Shanghai Aluminium Futures, February 2015 (ALG2015): 1
Shanghai Aluminium Futures, July 2015 (ALN2015)   : 1
Shanghai Aluminium Futures, June 2015 (ALM2015)   : 1
Shanghai Aluminium Futures, March 2015 (ALH2015)  : 1
(Other)                                     :94

Historical Shanghai Futures Prices: Copper Futures, September 2015 (CUU2015). Trading Unit : 5 ton/lot.
동99.95% Substitutions : 1.High grade Copper,GB/T467-1997 2. The LME Registered Brand,BS6017-1981, AMD72
Historical Shanghai Futures Prices: Zinc Futures, July 2015 (ZNN2015). Trading Unit : 5 ton/lot. Quotat
竊<8d>1997. For more details about the contract please visit : http://www.shfe.com.cn/docview/docview_72
Historical Shanghai Futures Prices: Zinc Futures, September 2015 (ZNU2015). Trading Unit : 5 ton/lot. Q
竊<8d>1997. For more details about the contract please visit : http://www.shfe.com.cn/docview/docview_72

```

```

Historical Shanghai Futures Prices: Steel Rebar Futures, July 2015 (RBN2015). Contract Size : \t10 tons/lot. Price
방art 2: Hot-rolled Ribbed Bar, GB1499.2-2007, with grades of HRB400 or HRBF400 with a diameter of 16mm, 18mm, 20mm
Historical Shanghai Futures Prices: Copper Futures, August 2015 (CUQ2015). Trading Unit : 5 ton/lot. Quotation Unit
동99.95% Substitutions : 1.High grade Copper,GB/T467-1997 2. The LME Registered Brand,BS6017-1981, AMD5725(CU-CAS)
Historical Shanghai Futures Prices: Copper Futures, July 2015 (CUN2015). Trading Unit : 5 ton/lot. Quotation Unit
동99.95% Substitutions : 1.High grade Copper,GB/T467-1997 2. The LME Registered Brand,BS6017-1981, AMD5725(CU-CAS)
(Other)

refreshed_at newest_available_date oldest_available_date
2015-08-17 21:07:13 UTC: 7    2015-04-15:10          2014-07-16:10
2015-09-15 21:07:23 UTC: 7    2015-02-10: 9           2014-08-18:10
2015-10-15 21:07:22 UTC: 7    2015-06-15: 9           2014-09-16:10
2015-06-25 17:04:19 UTC: 6    2015-08-17: 9           2014-10-16:10
2015-11-16 21:07:26 UTC: 6    2015-10-15: 9           2014-11-18: 7
2015-06-25 17:04:14 UTC: 5    2015-05-15: 8           2014-03-18: 6
(Other)                  :62   (Other)     :46           (Other)     :47

["Date", "Pre Settle", "Open", "High", "Low", "Close", "Settle", "CH1", "CH2", "Volume", "O.I.", "Change"]
["Date", "Pre Settle", "Open", "High", "Low", "Close", "Settle", "CH1", "CH2", "Volume", "Prev. Day Open Inter
frequency          type      premium   database_id
daily:100    Time Series:100  false:100  Min.       :1160
                           1st Qu.:1160
                           Median :1160
                           Mean    :1160
                           3rd Qu.:1160
                           Max.    :1160

```

6.3.2 모든 코드의 메타 데이터 가져오기

모든 코드의 메타 데이터는 다음의 호출을 이용하여 가져온다.

“https://www.quandl.com/api/v3/databases/SHFE/codes?api_key=XXXXXXXXXXXXXX” URL을 호출하면 CSV 파일이 들어있는 ZIP 압축 파일을 다운로드하며, 파일에는 데이터셋의 코드와 데이터셋의 이름 등의 정보를 포함하고 있다.

```

> url_name <- "https://www.quandl.com/api/v3/databases/SHFE/codes?api_key=XXXXXXXXXXXXXX"
>
> path <- "./datas"
> zip_fname <- paste(path, "SHFE-datasets-codes.zip", sep = "/")
> csv_fname <- sub("zip", "csv", zip_fname)
>
```

```
> ## download
> download.file(url_name, zip_fname, method = "libcurl", mode = "wb")
>
> ## unzip file
> unzip(zip_fname, exdir = path)
>
> SHFE_datasets <- read.csv(csv_fname, header = FALSE)
```

변수 이름 정의 및 데이터 구조 살펴 보기

```
> names(SHFE_datasets) <- c("dataset_id", "dataset_name")
>
> dim(SHFE_datasets)

[1] 1312     2

> head(SHFE_datasets, n = 3)

  dataset_id          dataset_name
1 SHFE/CUQ2003  Shanghai Copper Futures, August 2003 (CUQ2003)
2 SHFE/FUH2012  Shanghai Fuel Oil Futures, March 2012 (FUH2012)
3 SHFE/RUF2000 Shanghai Natural Rubber Futures, January 2000 (RUF2000)
```

제 7 절 연방 준비 제도 이사회 (FRB) 경제 지표 가져오기

세인트 루이스 연방 준비 은행(Federal Reserve Economic)의 연구 부서에서 생산한 성장, 고용, 물가, 노동, 제조 등의 경제 통계를 제공한다.

7.1 Code Formats

연방 준비 제도 이사회 (FRB) 경제 지표를 가져오는 Quandl 코드 포맷은 FRED/{CODE}로 구성되어 있으며, 각각의 CODE는 그룹으로 나뉘어져 있다. 대표적인 몇 개의 코드는 다음과 같다.

7.1.1 Growth

Growth 관련 코드는 표 2.4와 같다.

Code	Indicator
GDP	Gross Domestic Product
GDPC1	Real Gross Domestic Product
GDPPOT	Real Potential Gross Domestic Product

표 2.4: Growth 코드 포맷 종류

7.1.2 Prices and Inflation

Prices and Inflation 관련 코드는 표 2.5와 같다.

Code	Indicator
CPIAUCSL	Consumer Price Index for All Urban Consumers: All Items
CPILFESL	Consumer Price Index for All Urban Consumers: All Items Less Food & Energy
GDPDEF	Gross Domestic Product: Implicit Price Deflator

표 2.5: Prices and Inflation 코드 포맷 종류

7.1.3 Money Supply

Money Supply 관련 코드는 표 2.6과 같다.

Code	Indicator
BASE	St. Louis Adjusted Monetary Base
M1	M1 Money Stock
M2	M2 Money Stock
M1V	Velocity of M1 Money Stock
M2V	Velocity of M2 Money Stock

표 2.6: Money Supply 코드 포맷 종류

7.1.4 Interest Rates

Interest Rates 관련 코드는 표 2.7과 같다.

Code	Indicator
DFF	Effective Federal Funds Rate
DTB3	3-Month Treasury Bill: Secondary Market Rate
DGS5	5-Year Treasury Constant Maturity Rate
DGS10	10-Year Treasury Constant Maturity Rate
DGS30	30-Year Treasury Constant Maturity Rate
T5YIE	5-year Breakeven Inflation Rate
T10YIE	10-year Breakeven Inflation Rate
T5YIFR	5-Year, 5-Year Forward Inflation Expectation Rate
TEDRATE	TED Spread
DPRIME	Bank Prime Loan Rate

표 2.7: Interest Rates 코드 포맷 종류

7.1.5 Employment

Employment 관련 코드는 표 2.8과 같다.

7.1.6 Income and Expenditure

Income and Expenditure 관련 코드는 표 2.9와 같다.

7.1.7 Other Economic Indicators

Other Economic Indicators 관련 코드는 표 2.10과 같다.

Code	Indicator
UNRATE	Civilian Unemployment Rate
NROU	Natural Rate of Unemployment (Long-Term)
NROUST	Natural Rate of Unemployment (Short-Term)
CIVPART	Civilian Labor Force Participation Rate
EMRATIO	Civilian Employment-Population Ratio
UNEMPLOY	Unemployed
PAYEMS	All Employees: Total nonfarm
MANEMP	All Employees: Manufacturing
ICSA	Initial Claims
IC4WSA	4-Week Moving Average of Initial Claims

표 2.8: Employment 코드 포맷 종류

Code	Indicator
MEHOINUSA672N	Real Median Household Income in the United States
DSPIC96	Real Disposable Personal Income
PCE	Personal Consumption Expenditures
PCEDG	Personal Consumption Expenditures: Durable Goods
PSAVERT	Personal Saving Rate
RRSFS	Real Retail and Food Services Sales
DSPI	Disposable personal income

표 2.9: Income and Expenditure 코드 포맷 종류

7.1.8 Debt

Debt 관련 코드는 표 2.11과 같다.

7.2 데이터 가져오기

7.2.1 연방 준비 제도 이사회 데이터 가져오기

Quandl 패키지의 Quandl() 함수를 이용해서 데이터를 가져온다. code 인수는 Quandl 코드 포맷을 사용한다.

국내 총생산(Gross Domestic Product) 데이터를 가져오기 위해서는 표 2.4에 의거해서 코드인 **GDP**를 사용한다. 물론 가져온 데이터는 미국의 국내 총생산을 의미한다.

```
> library(Quandl)
>
> ## get data
> GDP <- Quandl(code = "FRED/GDP", api_key = "XXXXXXXXXXXXXXXXXXXX")
```

7.2.2 연방 준비 제도 이사회 데이터 살펴보기

```
> ## dimensiond
> dim(GDP)
[1] 276   2
```

Code	Indicator
INDPRO	Industrial Production Index
TCU	Capacity Utilization: Total Industry
HOUST	Housing Starts: Total: New Privately Owned Housing Units Started
GPD1	Gross Private Domestic Investment
CP	Corporate Profits After Tax (without IVA and CCAdj)
STLFSI	St. Louis Fed Financial Stress Index
DCOILWTICO	Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma
USSLIND	Leading Index for the United States
DTWEXM	Trade Weighted U.S. Dollar Index: Major Currencies
DTWEXB	Trade Weighted U.S. Dollar Index: Broad

표 2.10: Other Economic Indicators 코드 포맷 종류

Code	Indicator
GFDEBTN	Federal Debt: Total Public Debt
GFDEGDQ188S	Federal Debt: Total Public Debt as Percent of Gross Domestic Product
EXCSRESNW	Excess Reserves of Depository Institutions
TOTCI	Commercial and Industrial Loans, All Commercial Banks

표 2.11: Debt 코드 포맷 종류

```
> ## variable names
> names(GDP)

[1] "DATE"   "VALUE"

> ## a few datas
> head(GDP, n = 3)

      DATE   VALUE
1 2015-10-01 18164.8
2 2015-07-01 18060.2
3 2015-04-01 17913.7
```

7.3 메타 데이터 가져오기

7.3.1 상세 메타 데이터 가져오기

연방 준비 제도 이사회 데이터의 메타 데이터는 데이터셋 아이디, 데이터셋 이름, 시작일자, 종료일자, 마지막으로 업데이트된 일자, 사용 가능한 열 등의 정보를 담고 있다. 그리고 다음과 같은 URL을 이용해서 가져온다. 이 URL은 메타 데이터를 CSV 파일로 가져온다. 만약에 “datasets.csv” 부분에서 “csv” 대신 “json”, “xml”을 사용하면 각각 JSON과 XML 포맷의 파일을 가져오게 된다.

“https://www.quandl.com/api/v3/datasets.csv?database_code=FRED&per_page=100&sort_by=id&page=1”
이 호출의 의미는 다음과 같다.

- database_code=FRED : 데이터베이스 코드가 FRED로 연방 준비 제도 이사회의 데이터
- per_page=100 : 호출당 100개의 데이터 조회

- sort_by=id : 조회 데이터를 정렬할 변수의 이름
- page=1 : 호출 페이지의 번호가 1번

선물 데이터의 메타 데이터 개수는 194,820건이므로 메타 정보의 상세 조회를 위한 API 호출을 1,949회를 수행해야 모든 정보를 얻을 수 있다.

첫 페이지 100건의 메타 데이터를 가져와 보자.

```
url_name <- "https://www.quandl.com/api/v3/datasets.csv?database_code=FRED&per_page=100&sort_by=id&page=1"

detail_meta <- read.csv(url_name)

> ## dimension
> dim(detail_meta)

[1] 100 13

> ## variable names
> names(detail_meta)

[1] "id"           "dataset_code"      "database_code"
[4] "name"          "description"       "refreshed_at"
[7] "newest_available_date" "oldest_available_date" "column_names"
[10] "frequency"     "type"            "premium"
[13] "database_id"

> ## few data
> head(detail_meta, n = 3)

      id dataset_code database_code
1 10919628 EXPEF54171ALLEST      FRED
2 10919629 EXPEF54181ALLEST      FRED
3 10919630 REVEF54181ALLEST      FRED

1 Total Expense for Research and Development In The Physical, Engineering, and Life Sciences, All Establishments
2 Total Expense for Advertising Agencies, All Establishments
3 Total Revenue for Advertising Agencies, All Establishments

1 Millions of Dollars Not Seasonally Adjusted, For further information, please refer to the US Census Bureau's website.
2 Millions of Dollars Not Seasonally Adjusted, For further information, please refer to the US Census Bureau's website.
3 Millions of Dollars Not Seasonally Adjusted, For further information, please refer to the US Census Bureau's website.

      refreshed_at newest_available_date oldest_available_date column_names
1 2016-04-11 14:33:21 UTC      2014-01-01      2003-01-01 ["DATE", "VALUE"]
2 2016-04-10 21:05:50 UTC      2014-01-01      2003-01-01 ["DATE", "VALUE"]
3 2016-04-11 00:00:16 UTC      2014-01-01      1998-01-01 ["DATE", "VALUE"]

      frequency      type premium database_id
1 annual Time Series    false      118
2 annual Time Series    false      118
3 annual Time Series    false      118
```

```
> ## summary
> summary(detail_meta)

      id          dataset_code database_code
Min. :10919628  ECREF71ALLEST   : 1   FRED:100
1st Qu.:10919653 ERVEF7111ALLEST : 1
Median :10919678 EXPAA00EF51111ALLEST: 1
Mean   :10919678 EXPAA00EF51224ALLEST: 1
3rd Qu.:10919702 EXPAA00EF51913ALLEST: 1
Max.  :10919727 EXPDACEF484ALLEST  : 1
(Other)           :94

Depreciation and Amortization Charges for Securities, Commodity Contracts, and Other Financial Investments and
Operating Interest Expense for Securities, Commodity Contracts, and Other Financial Investments and Related Ac
Total Revenue for All Other Consumer Goods Rental, All Establishments, Employer Firms
All Other Operating Expenses for Internet Publishing and Broadcasting and Web Search Portals, All Establishmen
All Other Operating Expenses for Newspaper Publishers, All Establishments, Employer Firms
All Other Operating Expenses for Sound Recording Studios, All Establishments, Employer Firms
(Other)

Millions of Dollars Not Seasonally Adjusted, For further information, please refer to the US Census Bureau's An
```

	refreshed_at	newest_available_date	oldest_available_date
2016-04-11 14:16:50 UTC: 4	2006-01-01: 1	2010-01-01:31	
2016-04-10 19:40:14 UTC: 3	2007-01-01: 8	1998-01-01:23	
2016-04-10 21:05:47 UTC: 3	2014-01-01:91	2005-01-01:16	
2016-04-10 19:40:26 UTC: 2		2009-01-01:11	
2016-04-10 21:05:48 UTC: 2		2003-01-01: 6	
2016-04-11 14:14:23 UTC: 2		2004-01-01: 4	
(Other) :84	(Other) : 9		

	column_names	frequency	type	premium	database_id
["DATE", "VALUE"] :100	annual:100	Time Series:100	false:100	Min. :118	
				1st Qu.:118	
				Median :118	
				Mean :118	
				3rd Qu.:118	
				Max. :118	

7.3.2 모든 코드의 메타 데이터 가져오기

모든 코드의 메타 데이터는 다음의 호출을 이용하여 가져온다.

“https://www.quandl.com/api/v3/databases/FRED/codes?api_key=XXXXXXXXXXXXXX” 이 URL을 호출하면 CSV 파일이 들어있는 ZIP 압축 파일을 다운로드하며, 파일에는 데이터셋의 코드와 데이터셋의 이름 등의 정보를 포함하고 있다.

```
> url_name <- "https://www.quandl.com/api/v3/databases/FRED/codes?api_key=XXXXXXXXXXXXXX"
>
> path <- "./datas"
> zip_fname <- paste(path, "FRED-datasets-codes.zip", sep = "/")
> csv_fname <- sub("zip", "csv", zip_fname)
>
> ## file download
> download.file(url_name, zip_fname, method = "libcurl", mode = "wb")
>
> ## unzip file
> unzip(zip_fname, exdir = path)
>
> FRED_datasets <- read.csv(csv_fname, header = FALSE)
```

변수 이름 정의 및 데이터 구조 살펴 보기

```
> names(FRED_datasets) <- c("dataset_id", "dataset_name")
>
> dim(FRED_datasets)

[1] 194820      2

> head(FRED_datasets, n = 3)

  dataset_id
1 FRED/SMU27334601523610001
2 FRED/SMU56000003100000001
3 FRED/SMU36935615552410001

1 All Employees: Mining, Logging, and Construction: Residential Building Construction in Minneapolis-St.
2                                         All Employees: Manufactu
3                                         All Employees: Financial Activities: Insurance
```

제 3 장

주가 데이터 수집하기

제 1 절 국내 주가데이터 수집

1.1 종목코드 데이터 수집

국내 주가 정보는 야후(yahoo)와 구글(google)에서 제공하고 있으며, 각 사이트에서의 종목코드 포맷은 다음의 구조로 정의되어 있음

- 야후
 - KOSPI : 종목코드.KS
 - 코스닥 : 종목코드.KQ
 - 구글
 - KOSPI : KRX:종목코드
 - 코스닥 : KOSDAQ:종목코드
- 그래서 우선 KRX의 상장회사검색(<http://marketdata.krx.co.kr/mdi#document=040601>)메뉴에서 국내 상장 주식회사의 종목코드 정보를 수집하였음. 각각의 코드는 다음의 excel 파일로 저장함.
- KOSPI
 - KRX_KOSPI_CODE.xls
 - KOSDAQ
 - KRX_KOSDAQ_CODE.xls
 - KONEX
 - KRX_KONEX_CODE.xls

1.2 종목코드 데이터 적재

```

> library(xlsx)
>
> path <- "./datas/KRX/CODE"
>
> KOSPI <- "KRX_KOSPI_CODE.xls"
> KOSDAQ <- "KRX_KOSDAQ_CODE.xls"
> KONEX <- "KRX_KONEX_CODE.xls"
>
> fKOSPI <- paste(path, KOSPI, sep = "/")
> fKOSDAQ <- paste(path, KOSDAQ, sep = "/")
> fKONEX <- paste(path, KONEX, sep = "/")
>
> KOSPI <- read.xlsx(fKOSPI, sheetIndex = 1, encoding = "UTF-8")
> KOSDAQ <- read.xlsx(fKOSDAQ, sheetIndex = 1, encoding = "UTF-8")
> KONEX <- read.xlsx(fKONEX, sheetIndex = 1, encoding = "UTF-8")
>
> KOSPI <- KOSPI[, -1]
> names(KOSPI) <- c("symbols", "company", "industry_cd", "industry_nm", "stock_cnt", "capital",
+   "face_value", "currency", "phone", "address")
>
> KOSPI$symbols <- as.character(KOSPI$symbols)
> KOSPI$stock_cnt <- as.numeric(gsub(", ", "", KOSPI$stock_cnt))
> KOSPI$capital <- as.numeric(gsub(", ", "", KOSPI$capital))
> KOSPI$face_value <- as.integer(gsub(", ", "", KOSPI$face_value))
>
> KOSDAQ <- KOSDAQ[, -1]
> names(KOSDAQ) <- c("symbols", "company", "industry_cd", "industry_nm", "stock_cnt", "capital",
+   "face_value", "currency", "phone", "address")
>
> KOSDAQ$symbols <- as.character(KOSDAQ$symbols)
> KOSDAQ$stock_cnt <- as.numeric(gsub(", ", "", KOSDAQ$stock_cnt))
> KOSDAQ$capital <- as.numeric(gsub(", ", "", KOSDAQ$capital))
> KOSDAQ$face_value <- as.integer(gsub(", ", "", KOSDAQ$face_value))
>
> KONEX <- KONEX[, -1]
> names(KONEX) <- c("symbols", "company", "industry_cd", "industry_nm", "stock_cnt", "capital",
+   "face_value", "currency", "phone", "address")
>
> KONEX$symbols <- as.character(KONEX$symbols)
> KONEX$stock_cnt <- as.numeric(gsub(", ", "", KONEX$stock_cnt))
> KONEX$capital <- as.numeric(gsub(", ", "", KONEX$capital))
> KONEX$face_value <- as.integer(gsub(", ", "", KONEX$face_value))
>
> KOSPI[KOSPI$symbols == "088350", ]

```

717 088350 한화생명	116501	보험업 868530000 4.34265e+12	5000 원(KRW)
phone		address	
717 02-789-5114 서울특별시 영등포구 63로 50			

1.3 종목별 거래정보 가져오기

종목별 거래정보는 시스템을 구축할 경우에는 적재된 종목코드 데이터의 첫번째 코드부터 마지막 코드까지 반복(iteration) 수행을 통해 개별개별 데이터를 가져와서 쌓아야 함

본 종목별 거래정보 가져오기 예제에서는 야후에서 가져옴. 실제로 구글에 비해서 야후의 데이터가 더 많은 기간의 데이터를 수용하고 있음. 다음 예제는 KOSPI의 첫 3종목의 주가 정보를 가져오는 예제임.

```
> library("quantmod")
>
> # getting stock data which 1st KOSPI code from Yahoo
> symbols <- paste(KOSPI$symbols[1], "KS", sep=".") 
> KOSPI_STOCKS <- getSymbols(symbols, from='1990-01-01', auto.assign = FALSE)
> KOSPI_STOCKS <- data.frame(date=index(KOSPI_STOCKS), symbols=KOSPI$symbols[1],
+                                Open=as.vector(KOSPI_STOCKS[, 1]),
+                                High=as.vector(KOSPI_STOCKS[, 2]),
+                                Low=as.vector(KOSPI_STOCKS[, 3]),
+                                Close=as.vector(KOSPI_STOCKS[, 4]),
+                                Volume=as.vector(KOSPI_STOCKS[, 5]),
+                                Adjusted=as.vector(KOSPI_STOCKS[, 6]))
> row.names(KOSPI_STOCKS) <- NULL
>
> n <- length(KOSPI[1:3])
>
> # getting stock data which 2nd, 3rd KOSPI code from Yahoo
> for (i in 2:n) {
+   symbols <- paste(KOSPI$symbols[i], "KS", sep=".") 
+   x <- getSymbols(symbols, from='1990-01-01', auto.assign = FALSE)
+   x <- data.frame(date=index(x), symbols=KOSPI$symbols[i],
+                    Open=as.vector(x[, 1]), High=as.vector(x[, 2]),
+                    Low=as.vector(x[, 3]), Close=as.vector(x[, 4]),
+                    Volume=as.vector(x[, 5]), Adjusted=as.vector(x[, 6]))
+   row.names(x) <- NULL
+
+   KOSPI_STOCKS <- rbind(KOSPI_STOCKS, x)
+ }
>
> head(KOSPI_STOCKS)

  date symbols  Open  High   Low Close  Volume Adjusted
1 2015-08-21 095570 30900 36500 28000 35500 1994000    35500
```

```

2 2015-08-24 095570 33500 34700 31200 32400 392800 32400
3 2015-08-25 095570 33050 33650 30950 31100 242700 31100
4 2015-08-26 095570 31300 38800 31300 35450 703700 35450
5 2015-08-27 095570 36200 40300 34800 37850 499900 37850
6 2015-08-28 095570 38300 39200 35450 35700 191300 35700

```

```
> tail(KOSPI_STOCKS)
```

	date	symbols	Open	High	Low	Close	Volume	Adjusted
3739	2016-04-12	006840	63500	63600	61600	62900	53600	62900
3740	2016-04-13	006840	62900	62900	62900	62900	0	62900
3741	2016-04-14	006840	61900	62800	61000	61100	79200	61100
3742	2016-04-15	006840	60600	62200	60600	61800	37000	61800
3743	2016-04-18	006840	61300	62600	61300	61800	28400	61800
3744	2016-04-19	006840	62400	63700	61900	63400	38500	63400

1.4 관심종목 거래정보 가져오기

1.4.1 삼성전자 거래정보 가져오기

삼성전자의 종목코드는 ‘005930’이다. 이 종목의 거래정보를 Yahoo로부터 가져오기 위해서 종목코드를 ‘005930.KS’로 변환 후 조회한다.

```

> symbols <- "005930"
> symbols <- paste(symbols, "KS", sep=".") 
> symbols
[1] "005930.KS"

> samsung.electric <- getSymbols(symbols, from='1990-01-01', auto.assign = FALSE)
>
> tail(samsung.electric)

          005930.KS.Open 005930.KS.High 005930.KS.Low 005930.KS.Close 005930.KS.Volume
2016-04-12      1270000      1281000      1266000      1275000      134000
2016-04-13      1275000      1275000      1275000      1275000          0
2016-04-14      1300000      1302000      1289000      1300000      335300
2016-04-15      1309000      1310000      1290000      1300000      136500
2016-04-18      1295000      1305000      1292000      1299000      128400
2016-04-19      1294000      1296000      1283000      1288000      144700
          005930.KS.Adjusted
2016-04-12      1275000
2016-04-13      1275000
2016-04-14      1300000
2016-04-15      1300000
2016-04-18      1299000
2016-04-19      1288000

```

1.4.2 삼성전자 거래정보 시각화

주가 거래정보를 시각화하기 위해서는 quantmod 패키지의 chartSeries() 함수를 사용한다. 삼성전자의 주가를 주가 그래프로 즐겨보는 candle sticks chart를 그려보면 그림 3.1과 같다.

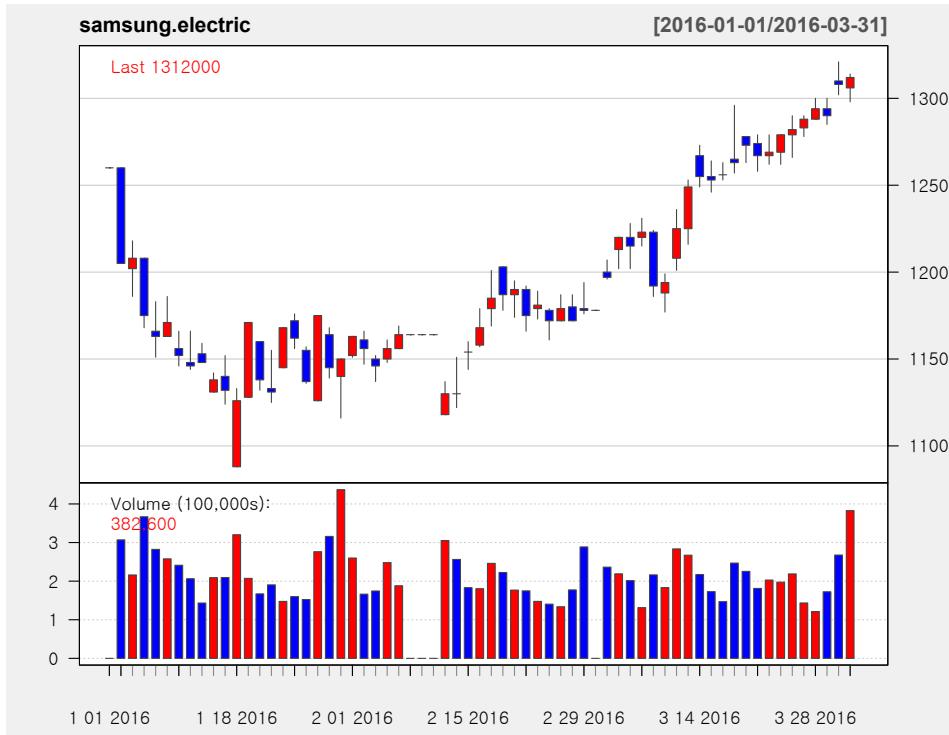


그림 3.1: 삼성전자 candle sticks chart

1.4.3 한화생명/삼성생명 거래정보 가져오기

한화생명과 삼성생명의 종목코드는 각각 ‘088350’와 ‘032830’이다. 이 종목의 거래정보를 Yahoo로부터 가져오기 위해서 종목코드를 ‘088350.KS’, ‘032830.KS’로 변환 후 조회한다.

```
> symbols <- "088350"
> symbols <- paste(symbols, "KS", sep=". ")
> symbols

[1] "088350.KS"

> hli <- getSymbols(symbols, from='1990-01-01', auto.assign = FALSE)
>
> tail(hli)

  088350.KS.Open 088350.KS.High 088350.KS.Low 088350.KS.Close 088350.KS.Volume
2016-04-12      6220        6280       6200        6260      554400
2016-04-13      6260        6260       6260        6260          0
2016-04-14      6220        6440       6220        6420     1334300
2016-04-15      6370        6580       6360        6530     892900
2016-04-18      6440        6560       6440        6480      585800
```

```

2016-04-19      6450      6860      6450      6850      1298800
                  088350.KS.Adjusted
2016-04-12      6260
2016-04-13      6260
2016-04-14      6420
2016-04-15      6530
2016-04-18      6480
2016-04-19      6850

> symbols <- "032830"
> symbols <- paste(symbols, "KS", sep=".") 
> symbols

[1] "032830.KS"

> sli <- getSymbols(symbols, from='1990-01-01', auto.assign = FALSE)
>
> tail(sli)

  032830.KS.Open 032830.KS.High 032830.KS.Low 032830.KS.Close 032830.KS.Volume
2016-04-12      114500      115500      113500      115500      202000
2016-04-13      115500      115500      115500      115500          0
2016-04-14      118000      121000      118000      121000     662500
2016-04-15      120000      121000      119000      121000     274200
2016-04-18      116500      116500      109500      110000     1079300
2016-04-19      110500      113000      110500      112500     270900

  032830.KS.Adjusted
2016-04-12      115500
2016-04-13      115500
2016-04-14      121000
2016-04-15      121000
2016-04-18      110000
2016-04-19      112500

```

1.4.4 한화생명/삼성생명 거래정보 시각화

주가 거래정보를 시각화하기 위해서 chartSeries() 함수로 그린 한화생명과 삼성생명의 candle sticks chart는 각각 그림 3.2와 그림 3.3과 같다.

제 2 절 해외 주가데이터 수집

2.1 미국 종목코드 데이터 수집

TTR 패키지를 이용해서 미국의 AMEX, NASDAQ, NYSE 시장의 종목코드를 가져온다.

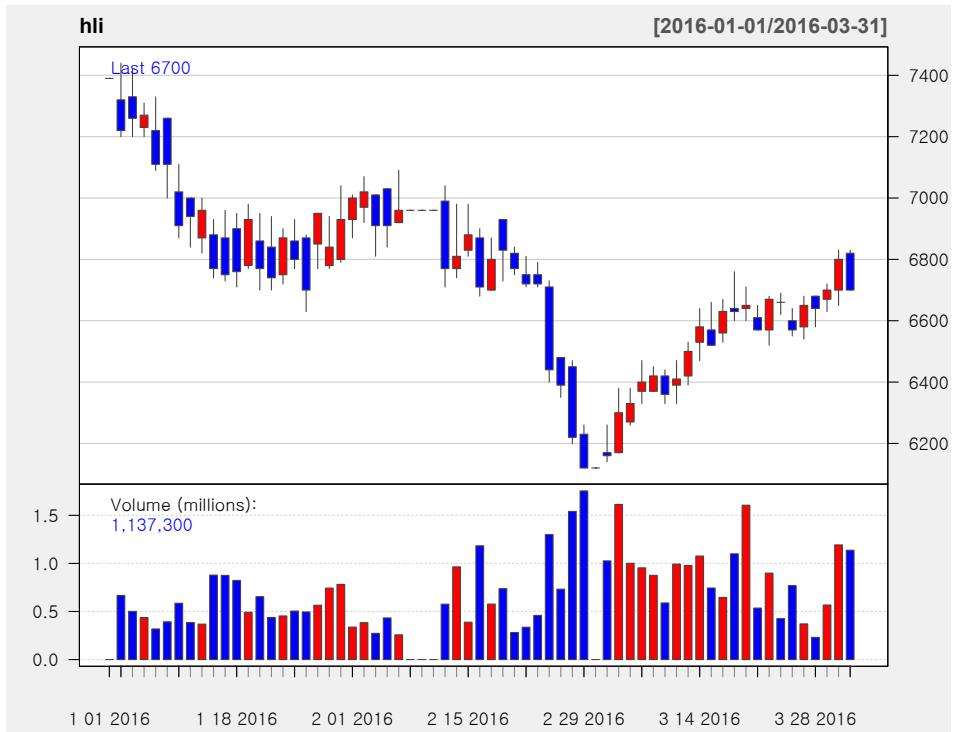


그림 3.2: 한화생명 candle sticks chart

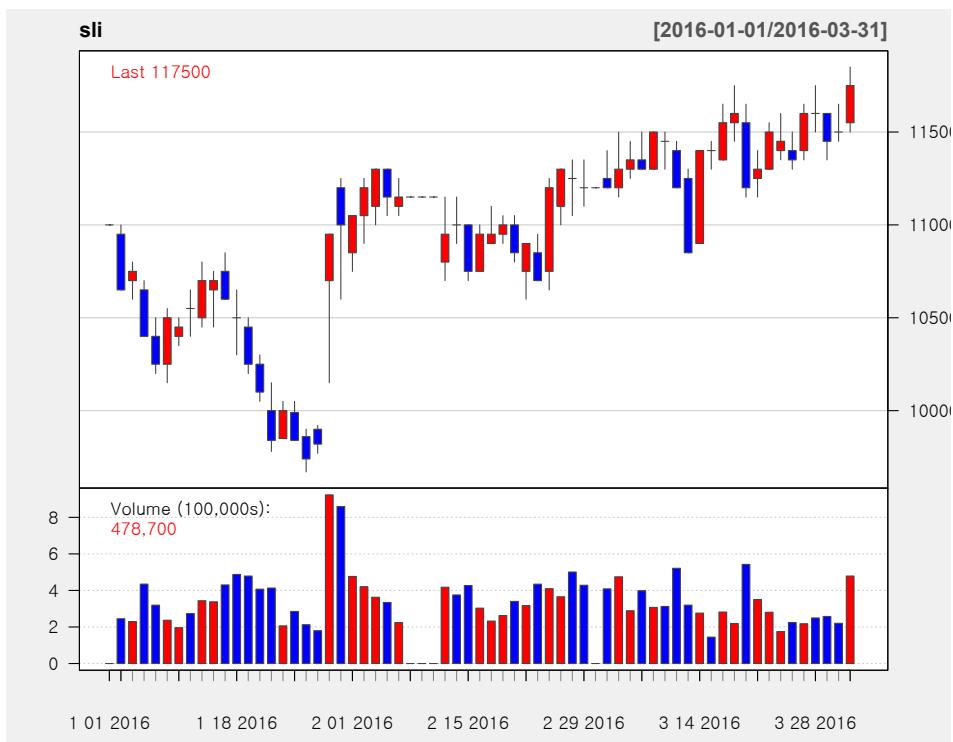


그림 3.3: 삼성생명 candle sticks chart

```
> AMEX <- TTR:::stockSymbols(exchange = c("AMEX"))
> NASDAQ <- TTR:::stockSymbols(exchange = c("NASDAQ"))
```

```
> NYSE <- TTR::stockSymbols(exchange = c("NYSE"))
>
> dim(AMEX)
[1] 368    8

> dim(NASDAQ)
[1] 3107    8

> dim(NYSE)
[1] 3214    8

> head(NYSE)

  Symbol           Name LastSale MarketCap IPOyear       Sector
1     A Agilent Technologies, Inc.   41.40 $13.57B  1999 Capital Goods
2    AA          Alcoa Inc.      10.53 $13.85B     NA Capital Goods
3  AA-PB          Alcoa Inc.     36.23    <NA>     NA      <NA>
4   AAC        AAC Holdings, Inc.   19.55 $449.36M  2014 Health Care
5   AAN        Aaron's, Inc.     26.64   $1.93B     NA Technology
6   AAP Advance Auto Parts Inc 160.79   $11.79B     NA Consumer Services
                                         Industry Exchange
1 Biotechnology: Laboratory Analytical Instruments      NYSE
2                           Metal Fabrications      NYSE
3                               <NA>      NYSE
4                   Medical Specialities      NYSE
5 Diversified Commercial Services      NYSE
6             Other Specialty Stores      NYSE
```

2.2 관심종목 거래정보 가져오기

2.2.1 애플 거래정보 가져오기

애플의 종목코드는 ‘AAPL’이다. 이 종목의 거래정보를 Yahoo로부터 가져오자.

```
> NASDAQ[grep("Apple", NASDAQ>Name), ]
  Symbol           Name LastSale MarketCap IPOyear       Sector
7   AAPL Apple Inc.   106.91 $592.77B  1980 Technology Computer Manufacturing   NASDAQ
                                         Industry Exchange
> symbols <- NASDAQ[grep("Apple", NASDAQ>Name), "Symbol"]
>
> apple <- getSymbols(symbols, from='1990-01-01', auto.assign = FALSE)
>
> tail(apple)

AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
```

2016-04-12	109.34	110.50	108.66	110.44	26812000	110.44
2016-04-13	110.80	112.34	110.80	112.04	32691800	112.04
2016-04-14	111.62	112.39	111.33	112.10	25337400	112.10
2016-04-15	112.11	112.30	109.73	109.85	46418500	109.85
2016-04-18	108.89	108.95	106.94	107.48	60834000	107.48
2016-04-19	107.88	108.00	106.23	106.91	32292300	106.91

2.2.2 애플 거래정보 시각화

애플의 candle sticks chart는 그림 3.4와 같다.

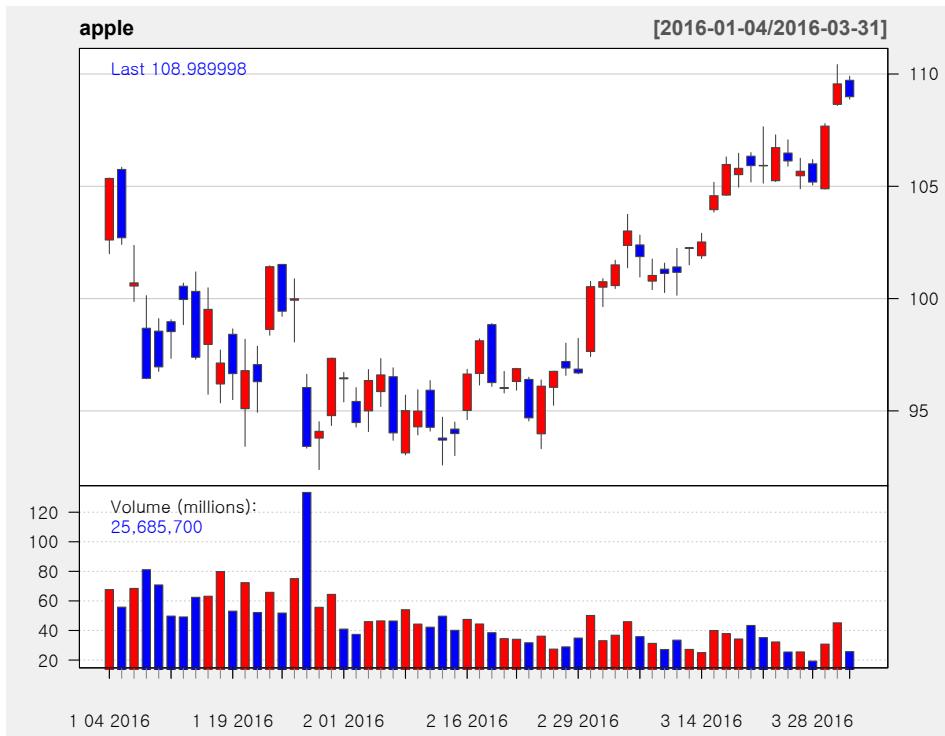


그림 3.4: Apple candle sticks chart

제 3 절 수익률 분석

3.1 한화생명 수익률 분석

주식 분할 및 배당 정보를 반영한 종가인 'Adjusted' 변수로 간단한 수익률 분석을 수행한다.

3.2 최고가/최저가

2010-03-17 이후의 한화생명이 최고가를 친 날은 2015-11-03로 8,352.69원이다. 반대로, 최저가를 친 날은 2011-09-23로 4,840.9원이다.

```
> hli1 <- hli[, "088350.KS.Adjusted"]
> hli1[which.max(hli1)]

 088350.KS.Adjusted
2015-11-03      8352.69

> hli1[which.min(hli1)]

 088350.KS.Adjusted
2011-09-23      4840.9
```

3.3 단순 수익률/복리 수익률

매일매일의 단순 수익률과 복리 수익률은 각각 다음과 같이 구한다.

```
> ret.simple <- diff(hli1) / lag(hli1, k = -1) * 100
> tail(ret.simple)

 088350.KS.Adjusted
2016-04-12      0.1597444
2016-04-13      0.0000000
2016-04-14      2.4502297
2016-04-15      1.6975309
2016-04-18      -0.7299270
2016-04-19      NA

> ret.cont <- diff(log(hli1)) * 100
> tail(ret.cont)

 088350.KS.Adjusted
2016-04-12      0.1598721
2016-04-13      0.0000000
2016-04-14      2.5237933
2016-04-15      1.6988826
2016-04-18      -0.7686433
2016-04-19      5.5528142
```

단순 수익률을 요약하면 다음과 같다. 가장 큰 일일 손실율은 2013-02-26의 -10.8%이고, 가장 큰 일일 수익율은 2011-11-01의 10.69%이다. 그런데 평균 일일 수익율은 -0.02%이다.

```
> summary(coredata(ret.simple))

088350.KS.Adjusted
Min.   :-10.81824
1st Qu.: -0.96861
Median :  0.00000
Mean   : -0.02332
3rd Qu.:  0.90142
```

```

Max.     : 10.68809
NA's     :2

> ret.simple[which.min(ret.simple)]

088350.KS.Adjusted
2013-02-26      -10.81824

> ret.simple[which.max(ret.simple)]

088350.KS.Adjusted
2011-11-01      10.68809

```

일일 단순 손실율의 분포를 히스토그램으로 그리면 그림 3.5와 같다. 결과를 보면 0% 근접의 마이너스(-) 수익률의 비율이 가장 높음을 알 수 있다.

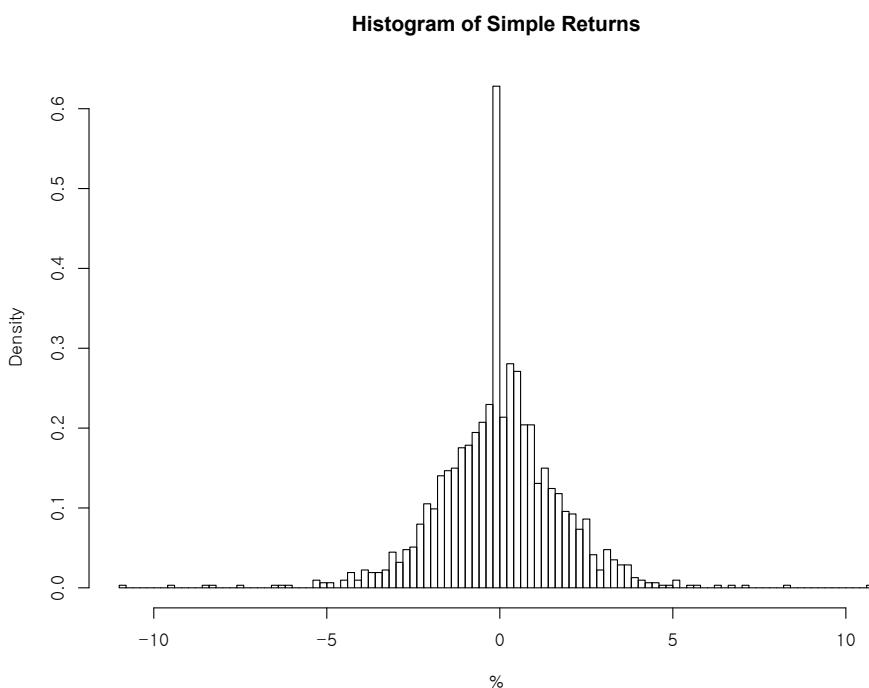


그림 3.5: Apple candle sticks chart

3.4 Value-at-Risk

일일 99% Value-at-Risk 값을 과거 자료로 분석하면 일일 손실율이 4.48% 이상의 확률은 1% 가량 발생한다. 즉, working day 기준 일년에 약 2.5번 정도가 최대 손실율인 4.48%가 발생한다는 것을 의미한다.

```

> quantile(ret.simple, probs = 0.01, na.rm = TRUE)

1%
-4.484582

```

3.5 복합 연간 성장률

C.A.G.R. : 복합 연간 성장률 (CAGR, 연평균성장률)을 구하면 다음과 같다.

```
> periodReturn(hli, period='yearly')

      yearly.returns
2010-12-30 -0.08620690
2011-12-30 -0.06918239
2012-12-31  0.04864865
2013-12-31 -0.02190722
2014-12-31  0.09222661
2015-12-31 -0.10856454
2016-04-19 -0.07307172
```

3.6 삼성생명 수익률 분석

주식 분할 및 배당 정보를 반영한 종가인 ‘Adjusted’ 변수로 간단한 수익률 분석을 수행한다.

3.7 최고가/최저가

2010-05-12 이후의 삼성생명이 최고가를 친 날은 2014-12-04로 123,014.9원이다. 반대로, 최저가를 친 날은 2012-01-16로 74,320.07원이다.

```
> sli1 <- sli[, "032830.KS.Adjusted"]
> sli1[which.max(sli1)]

      032830.KS.Adjusted
2014-12-04          123014.9

> sli1[which.min(sli1)]

      032830.KS.Adjusted
2012-01-16          74320.07
```

3.8 단순 수익률/복리 수익률

매일매일의 단순 수익률과 복리 수익률은 각각 다음과 같이 구한다.

```
> ret.simple <- diff(sli1) / lag(sli1, k = -1) * 100
> tail(ret.simple)

      032830.KS.Adjusted
2016-04-12          1.298701
2016-04-13          0.000000
2016-04-14          4.545455
2016-04-15          0.000000
2016-04-18         -9.777778
2016-04-19            NA
```

```
> ret.cont <- diff(log(sli1)) * 100
> tail(ret.cont)

032830.KS.Adjusted
2016-04-12      1.307208
2016-04-13      0.000000
2016-04-14      4.652002
2016-04-15      0.000000
2016-04-18     -9.531018
2016-04-19      2.247286
```

단순 수익률을 요약하면 다음과 같다. 가장 큰 일일 손실율은 2016-04-18의 -9.8%이고, 가장 큰 일일 수익율은 2010-05-26의 11.0%이다. 그런데 평균 일일 수익율은 -0.006%이다.

```
> summary(coredata(ret.simple))

032830.KS.Adjusted
Min.    :-9.777778
1st Qu.:-0.909096
Median  : 0.000000
Mean    :-0.006653
3rd Qu.: 0.877198
Max.    :11.009177
NA's    :2

> ret.simple[which.min(ret.simple)]

032830.KS.Adjusted
2016-04-18      -9.777778

> ret.simple[which.max(ret.simple)]

032830.KS.Adjusted
2010-05-26      11.00918
```

일일 단순 손실율의 분포를 히스토그램으로 그리면 그림 3.6과 같다. 결과를 보면 0% 근처의 마이너스(-) 수익률의 비율이 가장 높음을 알 수 있다.

3.9 Value-at-Risk

일일 99% Value-at-Risk 값을 과거 자료로 분석하면 일일 손실율이 3.91% 이상의 확률은 1% 가량 발생한다. 즉, working day 기준 일년에 약 2.5번 정도가 최대 손실율인 3.91%가 발생한다는 것을 의미한다. 한화생명보다는 리스크가 작다.

```
> quantile(ret.simple, probs = 0.01, na.rm = TRUE)

1%
-3.908303
```

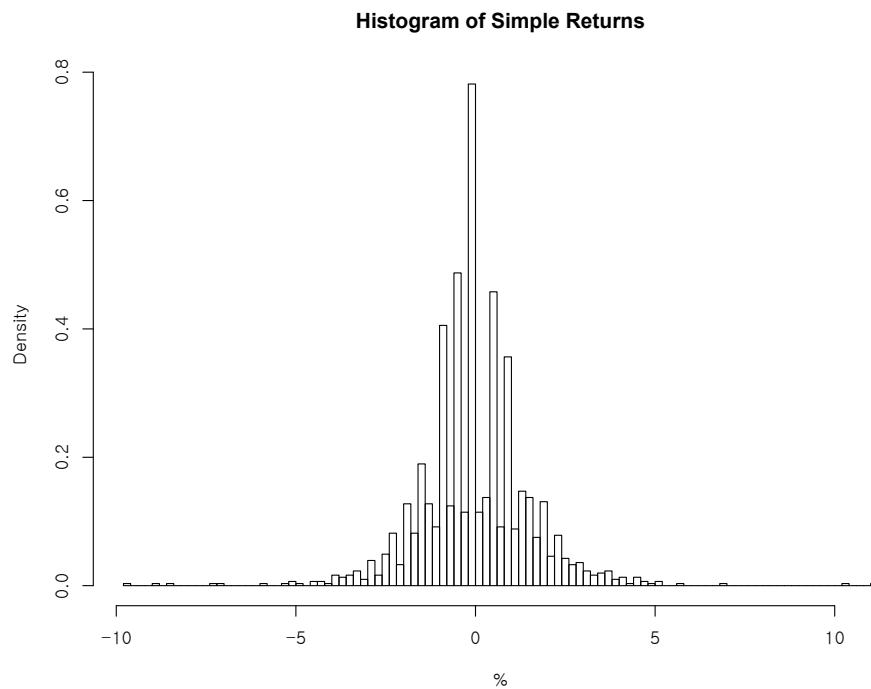


그림 3.6: Apple candle sticks chart

3.10 복합 연간 성장률

C.A.G.R. : 복합 연간 성장률 (CAGR, 연평균성장률)을 구하면 다음과 같다.

```
> periodReturn(sli, period='yearly')

      yearly.returns
2010-12-30    -0.14225941
2011-12-30    -0.21073171
2012-12-31     0.16563659
2013-12-31     0.10286320
2014-12-31     0.12019231
2015-12-31    -0.05579399
2016-04-19     0.02272727
```