04 Importing and Managing Financial Data

Boni

9/3/2020

Use getSymbols()

```
library(quantmod)
getSymbols(Symbols = "AAPL", src = "av", api.key = "0ST2PMQENLUXD5YT")
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## [1] "AAPL"
```

```
# alphavantage 0ST2PMQENLUXD5YT
# src can be alphavantage (av), google, yahoo, fred
first(AAPL, 5)
```

```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume
## 2020-04-16    287.38    288.20   282.35     286.69    39281300
## 2020-04-17    284.69    286.95   276.86     282.80    53812500
## 2020-04-20    277.95    281.68   276.85     276.93    32503800
## 2020-04-21    276.28    277.25   265.43     268.37    45247900
## 2020-04-22    273.61    277.90   272.20     276.10    29264300
```

```
getSymbols("GDP", src="FRED")
```

```
## [1] "GDP"
```

```
first(GDP, 5)
```

```
##                GDP
## 1947-01-01 243.164
## 1947-04-01 245.968
## 1947-07-01 249.585
## 1947-10-01 259.745
## 1948-01-01 265.742
```

Use Quandl()

```
library(Quandl)
dgs10 <- Quandl::Quandl(code = "FRED/DGS10", type = "xts")
first(dgs10, 5)
```

```
##            [,1]
## 1962-01-02 4.06
## 1962-01-03 4.03
## 1962-01-04 3.99
## 1962-01-05 4.02
## 1962-01-08 4.03
```

Get currency from Oanda

```
# Get available currency in Oanda
head(quantmod::oanda.currencies)
```

```
##     oanda.df.1.length.oanda.df...2....1.
## USD                            US Dollar
## AFN                   Afghanistan Afghani
## ALL                          Albanian Lek
## DZD                        Algerian Dinar
## ADF                        Andorran Franc
## ADP                       Andorran Peseta
```

```
# Create a currency_pair object
currency_pair <- "GBP/CAD"

# Load British Pound to Canadian Dollar exchange rate data
getSymbols(currency_pair, src = "oanda")
```

```
## [1] "GBP/CAD"
```

```
# Examine object using str()
str(GBPCAD)
```

```
## An 'xts' object on 2020-03-10/2020-09-05 containing:
##   Data: num [1:180, 1] 1.78 1.77 1.76 1.73 1.69 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "GBP.CAD"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "oanda"
##  $ updated: POSIXct[1:1], format: "2020-09-05 23:57:14"
```

```r
# Try to load data from 190 days ago
getSymbols(currency_pair, from = Sys.Date() - 190, to = Sys.Date(), src = "oanda")
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Oanda only
## provides historical data for the past 180 days. Symbol: GBP/CAD
```

```
## [1] "GBP/CAD"
```

Unemployment Rate from FRED

```r
# Create a series_name object
series_name <- "UNRATE"

# Load the data using getSymbols
getSymbols(series_name, src = "FRED", type = "xts")
```

```
## [1] "UNRATE"
```

```r
head(UNRATE)
```

```
##            UNRATE
## 1948-01-01    3.4
## 1948-02-01    3.8
## 1948-03-01    4.0
## 1948-04-01    3.9
## 1948-05-01    3.5
## 1948-06-01    3.6
```

```r
# Create a quandl_code object
quandl_code <- "FRED/UNRATE"

# Load the data using Quandl
unemploy_rate <- Quandl(quandl_code, type = "xts")
head(unemploy_rate)
```

```
##          [,1]
## Jan 1948  3.4
## Feb 1948  3.8
## Mar 1948  4.0
## Apr 1948  3.9
## May 1948  3.5
## Jun 1948  3.6
```

Extract OHLC

```r
# Important method to extract OHLC information: Op(), Hi(), Lo(), Cl(), Vo(), Ad(), and OHLC()
# Learn more: help("OHLC.Transformations")

# Get Symbols
getSymbols("TSLA", src = "yahoo", type = "xts")
```

```
## [1] "TSLA"
```

```r
# Extract the close column
tsla_close <- Cl(TSLA)

# Look at the head of dc_close
head(tsla_close)
```

```
##            TSLA.Close
## 2010-06-29      4.778
## 2010-06-30      4.766
## 2010-07-01      4.392
## 2010-07-02      3.840
## 2010-07-06      3.222
## 2010-07-07      3.160
```

```r
# Extract the volume column
tsla_volume <- Vo(TSLA)

# Look at the head of dc_volume
head(tsla_volume)
```

```
##            TSLA.Volume
## 2010-06-29    93831500
## 2010-06-30    85935500
## 2010-07-01    41094000
## 2010-07-02    25699000
## 2010-07-06    34334500
## 2010-07-07    34608500
```

```r
# Extract the high, low, and close columns
tsla_hlc <- HLC(TSLA)

# Look at the head of dc_hlc
head(tsla_hlc)
```

```
##            TSLA.High TSLA.Low TSLA.Close
## 2010-06-29     5.000    3.508      4.778
## 2010-06-30     6.084    4.660      4.766
## 2010-07-01     5.184    4.054      4.392
## 2010-07-02     4.620    3.742      3.840
## 2010-07-06     4.000    3.166      3.222
## 2010-07-07     3.326    2.996      3.160
```

```r
# Extract the open, high, low, close, and volume columns
tsla_ohlcv <- OHLCV(TSLA)

# Look at the head of dc_ohlcv
head(tsla_ohlcv)
```

```
##            TSLA.Open TSLA.High TSLA.Low TSLA.Close TSLA.Volume
## 2010-06-29     3.800     5.000    3.508      4.778    93831500
## 2010-06-30     5.158     6.084    4.660      4.766    85935500
## 2010-07-01     5.000     5.184    4.054      4.392    41094000
## 2010-07-02     4.600     4.620    3.742      3.840    25699000
## 2010-07-06     4.000     4.000    3.166      3.222    34334500
## 2010-07-07     3.280     3.326    2.996      3.160    34608500
```

Extract the Close column from many instruments

```r
Cl <- function (x)
{
    if (has.Cl(x))
        return(x[, grep("Close", colnames(x), ignore.case = TRUE)])
    stop("subscript out of bounds: no column name containing \"Close\"")
}

# Symbols
symbols <- c("AAPL", "MSFT", "IBM")

# Create new environment
data_env <- new.env()

# Load symbols into data_env
getSymbols(symbols, env = data_env)
```

```
## [1] "AAPL" "MSFT" "IBM"
```

```r
# Extract the close column from each object and combine into one xts object
close_data <- do.call(merge, eapply(data_env, Cl))

# View the head of close_data
head(close_data)
```

```
##            AAPL.Close IBM.Close MSFT.Close
## 2007-01-03   2.992857     97.27      29.86
## 2007-01-04   3.059286     98.31      29.81
## 2007-01-05   3.037500     97.42      29.64
## 2007-01-08   3.052500     98.90      29.93
## 2007-01-09   3.306072    100.07      29.96
## 2007-01-10   3.464286     98.89      29.66
```

Change default getSymbols()

```
# Look at getSymbols.yahoo arguments
args(getSymbols.yahoo)
```

```
## function (Symbols, env, return.class = "xts", index.class = "Date",
##     from = "2007-01-01", to = Sys.Date(), ..., periodicity = "daily",
##     curl.options = list())
## NULL
```

```
# Set default 'from' value for getSymbols.yahoo
setDefaults(getSymbols.yahoo, from = "2000-01-01")

# Confirm defaults were set correctly
getDefaults(name = "getSymbols.yahoo")
```

```
## $from
## [1] "'2000-01-01'"
```

Handling instrument symbols that clash or are not valid R names

```
# get BRK-A
getSymbols("BRK-A")
```

```
## [1] "BRK-A"
```

```
# Assign the result to BRK.A
BRK.A <- get("BRK-A")

head(BRK.A)
```

```
##            BRK-A.Open BRK-A.High BRK-A.Low BRK-A.Close BRK-A.Volume
## 2000-01-03      56100      56100     53800       54800        36000
## 2000-01-04      53700      53800     52000       52000        44000
## 2000-01-05      51700      54700     51700       53200        51000
## 2000-01-06      53300      55000     53100       55000        57000
## 2000-01-07      55600      56500     55200       56500        67000
## 2000-01-10      57300      58000     56000       56000        31000
##            BRK-A.Adjusted
## 2000-01-03          54800
## 2000-01-04          52000
## 2000-01-05          53200
## 2000-01-06          55000
## 2000-01-07          56500
## 2000-01-10          56000
```

Another way to handle invalid names (1)

```r
# Create BRK.A object
BRK.A <- getSymbols("BRK-A", auto.assign = FALSE)

# Create col_names object with the column names of BRK.A
col_names <- colnames(BRK.A)

# Set BRK.A column names to syntactically valid names
colnames(BRK.A) <- make.names(col_names)

head(BRK.A)
```

```
##            BRK.A.Open BRK.A.High BRK.A.Low BRK.A.Close BRK.A.Volume
## 2000-01-03      56100      56100     53800       54800        36000
## 2000-01-04      53700      53800     52000       52000        44000
## 2000-01-05      51700      54700     51700       53200        51000
## 2000-01-06      53300      55000     53100       55000        57000
## 2000-01-07      55600      56500     55200       56500        67000
## 2000-01-10      57300      58000     56000       56000        31000
##            BRK.A.Adjusted
## 2000-01-03          54800
## 2000-01-04          52000
## 2000-01-05          53200
## 2000-01-06          55000
## 2000-01-07          56500
## 2000-01-10          56000
```

Another day to handle invalid names (2)

```r
# Set name for BRK-A to BRK.A
setSymbolLookup(BRK.A = list(name = "BRK-A"))

# Set name for T (AT&T) to ATT
setSymbolLookup(ATT = list(name = "T"))

# Load BRK.A and ATT data
getSymbols(c("BRK.A", "ATT"))
```

```
## [1] "BRK.A" "ATT"
```