Importing and Managing Financial Data in R

Emily Atkins

- **Importing and Managing Financial Data in R**
  - Chapter 1: Introduction and downloading data
    - 1.1 Introducing getSymbols()
    - 1.2 Data sources
    - 1.3 Make getSymbols() return the data it retrieves
    - 1.4 Introducing Quandl()
    - 1.5 Return data type
    - 1.6 Find stock ticker from Google Finance
    - 1.7 Download exchange rate data from Oanda
    - 1.8 Find and download US Civilian Unemployment Rate data from FRED
  - Chapter 2: Extracting and transforming data
    - 2.1 Extract one column from one instrument
    - 2.2 Extract multiple columns from one instrument
    - 2.3 Use getPrice to extract other columns
    - 2.4 Use Quandl to download weekly returns data
    - 2.5 Combine objects from an environment using do.call and eapply
    - 2.6 Use quantmod to download multiple instruments and extract the close column
- Quiz
  - Retrieve the data for the period between 2010-12-3 and 2013-12-31, using the packages and functions you learned in this course. And plot them using the plot function.
  - 1. Apple stock price (adjusted close)
  - 2. New Hampshire gross domestic product
  - 3. exchange rate, US Dollar per South Korean Won (for the past 180 days)

Importing and Managing Financial Data in R

Chapter 1: Introduction and downloading data

Load packages

```
# Load packages
library(quantmod)
```

1.1 Introducing getSymbols()

```
# Import QQQ data from Yahoo! Finance
getSymbols(Symbols = "QQQ", auto_assign = TRUE)
## [1] "QQQ"

# Look at the structure of the object getSymbols created
str(QQQ)
## An 'xts' object on 2007-01-03/2017-07-14 containing:
##   Data: num [1:2652, 1:6] 39.5 39.3 39.9 39.9 40 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:6] "QQQ.Open" "QQQ.High" "QQQ.Low" "QQQ.Close" ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "yahoo"
##  $ updated: POSIXct[1:1], format: "2017-07-15 12:27:20"

# Look at the first few rows of QQQ
head(QQQ)
##            QQQ.Open QQQ.High QQQ.Low QQQ.Close QQQ.Volume QQQ.Adjusted
## 2007-01-03   39.488   40.033  38.634  39.28798  167689500        43.24
## 2007-01-04   39.342   40.169  39.206  40.03301  136853500        44.06
## 2007-01-05   39.933   39.933  39.506  39.84221  138958800        43.85
## 2007-01-08   39.879   40.088  39.651  39.86947  106401600        43.88
## 2007-01-09   39.988   40.242  39.642  40.06936  121577500        44.10
## 2007-01-10   39.942   40.578  39.815  40.54183  121070100        44.62
```

1.2 Data sources

```
# Import QQQ data from Google Finance
getSymbols(Symbols = "QQQ", src = "google")
## [1] "QQQ"

# Look at the structure of QQQ
str(QQQ)
## An 'xts' object on 2007-01-03/2017-07-14 containing:
##   Data: num [1:2651, 1:5] 43.5 43.3 43.9 43.9 44 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "QQQ.Open" "QQQ.High" "QQQ.Low" "QQQ.Close" ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "google"
##  $ updated: POSIXct[1:1], format: "2017-07-15 12:27:21"

# Import GDP data from FRED
getSymbols(Symbols = "GDP", src = "FRED")
## [1] "GDP"

# Look at the structure of GDP
str(GDP)
## An 'xts' object on 1947-01-01/2017-01-01 containing:
##   Data: num [1:281, 1] 243 246 250 260 266 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "GDP"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "FRED"
##  $ updated: POSIXct[1:1], format: "2017-07-15 12:27:21"
```

1.3 Make getSymbols() return the data it retrieves

```
# Load the quantmod package
library(quantmod)

# Assign SPY data to object named 'spy' using auto.assign argument
spy <- getSymbols(Symbols = "SPY", auto.assign = FALSE)


# Look at the structure of the 'spy' object
str(spy)
## An 'xts' object on 2007-01-03/2017-07-14 containing:
##   Data: num [1:2652, 1:6] 114 114 114 113 114 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:6] "SPY.Open" "SPY.High" "SPY.Low" "SPY.Close" ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "yahoo"
##  $ updated: POSIXct[1:1], format: "2017-07-15 12:27:22"

# Assign JNJ data to object named 'jnj' using env argument
jnj <- getSymbols(Symbols = "JNJ", env = NULL)

# Look at the structure of the 'jnj' object
str(jnj)
## An 'xts' object on 2007-01-03/2017-07-14 containing:
##   Data: num [1:2652, 1:6] 48 47.8 48.5 48.3 48.3 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:6] "JNJ.Open" "JNJ.High" "JNJ.Low" "JNJ.Close" ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "yahoo"
##  $ updated: POSIXct[1:1], format: "2017-07-15 12:27:22"
```

1.4 Introducing Quandl()

```
# Load the Quandl package
library(Quandl)

# Import GDP data from FRED
gdp <- Quandl(code = "FRED/GDP")

# Look at the structure of the object returned by Quandl
str(gdp)
## 'data.frame':    281 obs. of  2 variables:
##  $ Date : Date, format: "2017-01-01" "2016-10-01" ...
##  $ Value: num  19027 18869 18675 18450 18282 ...
##  - attr(*, "freq")= chr "quarterly"
```

1.5 Return data type

```
# Import GDP data from FRED as xts
gdp_xts <- Quandl(code = "FRED/GDP", type = "xts")

# Look at the structure of gdp_xts
str(gdp_xts)
## An 'xts' object on 1947 Q1/2017 Q1 containing:
##   Data: num [1:281, 1] 243 246 250 260 266 ...
##   Indexed by objects of class: [yearqtr] TZ:
##   xts Attributes:
##  NULL

# Import GDP data from FRED as zoo
gdp_zoo <- Quandl(code = "FRED/GDP", type = "zoo")

# Look at the structure of gdp_zoo
str(gdp_zoo)
## 'zooreg' series from 1947 Q1 to 2017 Q1
##   Data: num [1:281] 243 246 250 260 266 ...
##   Index: Class 'yearqtr'  num [1:281] 1947 1947 1948 1948 1948 ...
##   Frequency: 4
```

1.6 Find stock ticker from Google Finance

```
# Create an object containing the Pfizer ticker symbol
symbol = "PFE"

# Use getSymbols to import the data
getSymbols(Symbols = symbol, src = "google")
## [1] "PFE"

# Look at the first few rows of data
head(PFE)
##            PFE.Open PFE.High PFE.Low PFE.Close PFE.Volume
## 2007-01-03    26.15    26.42   25.98     26.29   40645100
## 2007-01-04    26.38    26.57   26.29     26.38   32258100
## 2007-01-05    26.55    26.63   26.17     26.30   31355800
## 2007-01-08    26.29    26.42   25.89     26.16   43224400
## 2007-01-09    26.24    26.34   26.04     26.17   31321400
## 2007-01-10    26.10    26.26   26.00     26.20   34548400
```

1.7 Download exchange rate data from Oanda

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

# Load British Pound to Canadian Dollar exchange rate data
getSymbols(Symbols = currency_pair, src = "oanda")
## [1] "GBPCAD"

# Examine object using str()
str(GBPCAD)
## An 'xts' object on 2017-01-17/2017-07-14 containing:
##   Data: num [1:179, 1] 1.6 1.62 1.63 1.64 1.65 ...
##  - 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: "2017-07-15 12:27:24"

# Try to load data from 190 days ago
getSymbols(Symbols = currency_pair, from = Sys.Date() - 190, to = Sys.Date(), src = "oanda")
## [1] "GBPCAD"
```

1.8 Find and download US Civilian Unemployment Rate data from FRED

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

# Load the data using getSymbols
getSymbols(Symbols = series_name, src = "FRED")
## [1] "UNRATE"

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

# Load the data using Quandl
# unemploy_rate <- Quandl(code = quandl_code)
```

Chapter 2: Extracting and transforming data

```
# Provide your api_key
Quandl.api_key('v55sdwPxRDW1BeHgtzz2')

#Import GDP data from FRED
gdp <- Quandl(code = "FRED/GDP")
```

2.1 Extract one column from one instrument

```
# Load data
DC <- read.csv("~/resources/rstudio/DC.csv")
DC$Date <- as.POSIXct(DC$Date, format("%Y-%m-%d %H:%M:%S"))
DC <- xts(DC[, 2:6], order.by = DC$Date)
str(DC)
## An 'xts' object on 2016-01-16 01:00:00/2016-01-17 23:00:00 containing:
##   Data: num [1:47, 1:5] 20.8 20.8 20.8 20.8 20.8 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "DC.Open" "DC.High" "DC.Low" "DC.Close" ...
##   Indexed by objects of class: [POSIXct,POSIXt] TZ: %Y-%m-%d %H:%M:%S
##   xts Attributes:
##  NULL

# Look at the head of DC
head(DC)
##                     DC.Open DC.High DC.Low DC.Close DC.Volume
## 2016-01-16 01:00:00  20.845  20.850 20.835   20.845       157
## 2016-01-16 02:00:00  20.845  20.850 20.835   20.845       214
## 2016-01-16 03:00:00  20.845  20.850 20.835   20.845       103
## 2016-01-16 04:00:00  20.845  20.855 20.835   20.845       180
## 2016-01-16 05:00:00  20.845  20.845 20.845   20.845       211
## 2016-01-16 06:00:00  20.845  20.845 20.840   20.845        35

# Extract the close column
dc_close <- Cl(DC)

# Look at the head of dc_close
head(dc_close)
##                     DC.Close
## 2016-01-16 01:00:00   20.845
## 2016-01-16 02:00:00   20.845
## 2016-01-16 03:00:00   20.845
## 2016-01-16 04:00:00   20.845
## 2016-01-16 05:00:00   20.845
## 2016-01-16 06:00:00   20.845

# Extract the volume column
```

```
dc_volume <- Vo(DC)


# Look at the head of dc_volume
head(dc_volume)
##                      DC.Volume
## 2016-01-16 01:00:00       157
## 2016-01-16 02:00:00       214
## 2016-01-16 03:00:00       103
## 2016-01-16 04:00:00       180
## 2016-01-16 05:00:00       211
## 2016-01-16 06:00:00        35
```

2.2 Extract multiple columns from one instrument

```
# Extract the high, low, and close columns
dc_hlc <- HLC(DC)

# Look at the head of dc_hlc
head(dc_hlc)
##                     DC.High DC.Low DC.Close
## 2016-01-16 01:00:00  20.850 20.835   20.845
## 2016-01-16 02:00:00  20.850 20.835   20.845
## 2016-01-16 03:00:00  20.850 20.835   20.845
## 2016-01-16 04:00:00  20.855 20.835   20.845
## 2016-01-16 05:00:00  20.845 20.845   20.845
## 2016-01-16 06:00:00  20.845 20.840   20.845

# Extract the open, high, low, close, and volume columns
dc_ohlcv <- OHLCV(DC)

# Look at the head of dc_ohlcv
head(dc_ohlcv)
##                     DC.Open DC.High DC.Low DC.Close DC.Volume
## 2016-01-16 01:00:00  20.845  20.850 20.835   20.845       157
## 2016-01-16 02:00:00  20.845  20.850 20.835   20.845       214
## 2016-01-16 03:00:00  20.845  20.850 20.835   20.845       103
## 2016-01-16 04:00:00  20.845  20.855 20.835   20.845       180
## 2016-01-16 05:00:00  20.845  20.845 20.845   20.845       211
## 2016-01-16 06:00:00  20.845  20.845 20.840   20.845        35
```

2.3 Use getPrice to extract other columns

```
# Download CME data for CL and BZ as an xts object
oil_data <- Quandl(code = c("CME/CLH2016", "CME/BZH2016"), type = "xts")

# Look at the column names of the oil_data object
colnames(oil_data)
##  [1] "CME.CLH2016 - Open"          "CME.CLH2016 - High"
##  [3] "CME.CLH2016 - Low"           "CME.CLH2016 - Last"
##  [5] "CME.CLH2016 - Change"        "CME.CLH2016 - Settle"
##  [7] "CME.CLH2016 - Volume"        "CME.CLH2016 - Open Interest"
##  [9] "CME.BZH2016 - Open"          "CME.BZH2016 - High"
## [11] "CME.BZH2016 - Low"           "CME.BZH2016 - Last"
## [13] "CME.BZH2016 - Change"        "CME.BZH2016 - Settle"
## [15] "CME.BZH2016 - Volume"        "CME.BZH2016 - Open Interest"

# Extract the Open price for CLH2016
cl_open <- getPrice(oil_data, symbol = "CLH2016", prefer = "Open$")

# Look at January, 2016 using xts' ISO-8601 subsetting
cl_open["2016-01"]
##            CME.CLH2016 - Open
## 2016-01-04              38.75
## 2016-01-05              38.07
## 2016-01-06              37.39
## 2016-01-07              35.35
## 2016-01-08              34.50
## 2016-01-11              34.11
## 2016-01-12              32.24
## 2016-01-13              31.60
## 2016-01-14              31.47
## 2016-01-15              32.11
## 2016-01-19              30.17
## 2016-01-20              29.47
## 2016-01-21              28.35
## 2016-01-22              29.84
## 2016-01-25              32.05
## 2016-01-26              29.81
## 2016-01-27              30.55
```

```
## 2016-01-28              32.19
## 2016-01-29              33.70
```

2.4 Use Quandl to download weekly returns data

```
# CL and BZ Quandl codes
quandl_codes <- c("CME/CLH2016","CME/BZH2016")

# Download quarterly CL and BZ prices
qtr_price <- Quandl(quandl_codes, collapse = "quarterly", type = "xts")

# View the high prices for both series
Hi(qtr_price)
##         CME.CLH2016 - High CME.BZH2016 - High
## 2010 Q4              92.94                 NA
## 2011 Q1             101.43                 NA
## 2011 Q2             100.19                 NA
## 2011 Q3              91.59                 NA
## 2011 Q4               0.00               0.00
## 2012 Q1               0.00               0.00
## 2012 Q2               0.00               0.00
## 2012 Q3               0.00               0.00
## 2012 Q4               0.00               0.00
## 2013 Q1               0.00               0.00
## 2013 Q2               0.00               0.00
## 2013 Q3               0.00               0.00
## 2013 Q4               0.00               0.00
## 2014 Q1              86.00               0.00
## 2014 Q2              93.05                 NA
## 2014 Q3               0.00                 NA
## 2014 Q4              61.03              69.45
## 2015 Q1              57.17              64.45
## 2015 Q2              61.63              66.72
## 2015 Q3              47.95              51.37
## 2015 Q4              38.87              38.34
## 2016 Q1              32.05              35.00

# Download quarterly CL and BZ returns
qtr_return <- Quandl(quandl_codes, collapse = "quarterly", type = "xts", transform = "rdiff")

# View the settle price returns for both series
getPrice(qtr_return, prefer = "Settle")
##         CME.CLH2016 - Settle CME.BZH2016 - Settle
```

```
## 2011 Q1          0.091349258                NA
## 2011 Q2         -0.012225180                NA
## 2011 Q3         -0.085836910                NA
## 2011 Q4         -0.012337591                NA
## 2012 Q1          0.036038028         0.040508511
## 2012 Q2         -0.074797268        -0.040381031
## 2012 Q3          0.016030446         0.029779888
## 2012 Q4          0.008286039         0.006496228
## 2013 Q1         -0.011595182         0.001561524
## 2013 Q2         -0.045899772        -0.041679659
## 2013 Q3          0.025784887         0.027440347
## 2013 Q4         -0.011521005         0.038847250
## 2014 Q1          0.014480810         0.008535718
## 2014 Q2          0.079377974         0.052191436
## 2014 Q3         -0.069454897        -0.070669348
## 2014 Q4         -0.295436164        -0.306543019
## 2015 Q1         -0.073958675        -0.073551263
## 2015 Q2          0.088188419         0.071692061
## 2015 Q3         -0.229129373        -0.237503741
## 2015 Q4         -0.194215748        -0.260647694
## 2016 Q1         -0.175268535        -0.079108044
```

2.5 Combine objects from an environment using do.call and eapply

```r
# Creat a new environment
data_env <- new.env()

# Load data into the environment
getSymbols(Symbols = c("QQQ"), env = data_env, auto.assign = TRUE)
## [1] "QQQ"

# Call head on each object in data_env using eapply
data_list <- eapply(data_env, head)
head(data_list)
## $QQQ
##            QQQ.Open QQQ.High QQQ.Low QQQ.Close QQQ.Volume QQQ.Adjusted
## 2007-01-03   39.488   40.033  38.634  39.28798  167689500        43.24
## 2007-01-04   39.342   40.169  39.206  40.03301  136853500        44.06
## 2007-01-05   39.933   39.933  39.506  39.84221  138958800        43.85
## 2007-01-08   39.879   40.088  39.651  39.86947  106401600        43.88
## 2007-01-09   39.988   40.242  39.642  40.06936  121577500        44.10
## 2007-01-10   39.942   40.578  39.815  40.54183  121070100        44.62

# Merge all the list elements into one xts object
data_merged <- do.call(merge, data_list)
head(data_merged)
##            QQQ.Open QQQ.High QQQ.Low QQQ.Close QQQ.Volume QQQ.Adjusted
## 2007-01-03   39.488   40.033  38.634  39.28798  167689500        43.24
## 2007-01-04   39.342   40.169  39.206  40.03301  136853500        44.06
## 2007-01-05   39.933   39.933  39.506  39.84221  138958800        43.85
## 2007-01-08   39.879   40.088  39.651  39.86947  106401600        43.88
## 2007-01-09   39.988   40.242  39.642  40.06936  121577500        44.10
## 2007-01-10   39.942   40.578  39.815  40.54183  121070100        44.62

# Ensure the columns are ordered: open, high, low, close
data_ohlc <- OHLC(data_merged)
head(data_ohlc)
##            QQQ.Open QQQ.High QQQ.Low QQQ.Close
## 2007-01-03   39.488   40.033  38.634  39.28798
## 2007-01-04   39.342   40.169  39.206  40.03301
## 2007-01-05   39.933   39.933  39.506  39.84221
## 2007-01-08   39.879   40.088  39.651  39.86947
```

```
## 2007-01-09   39.988   40.242  39.642  40.06936
## 2007-01-10   39.942   40.578  39.815  40.54183
```

2.6 Use quantmod to download multiple instruments and extract the close column

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

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

# Load symbols into data_env
getSymbols(Symbols = symbols, env = data_env)
## [1] "AAPL" "MSFT" "IBM"

# Extract the close column from each object and merge 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   10.81246  76.98791   23.20394
## 2007-01-04   11.05245  77.81103   23.16508
## 2007-01-05   10.97374  77.10663   23.03297
## 2007-01-08   11.02793  78.27800   23.25834
## 2007-01-09   11.94403  79.20407   23.28165
## 2007-01-10   12.51562  78.27012   23.04851
```

Quiz

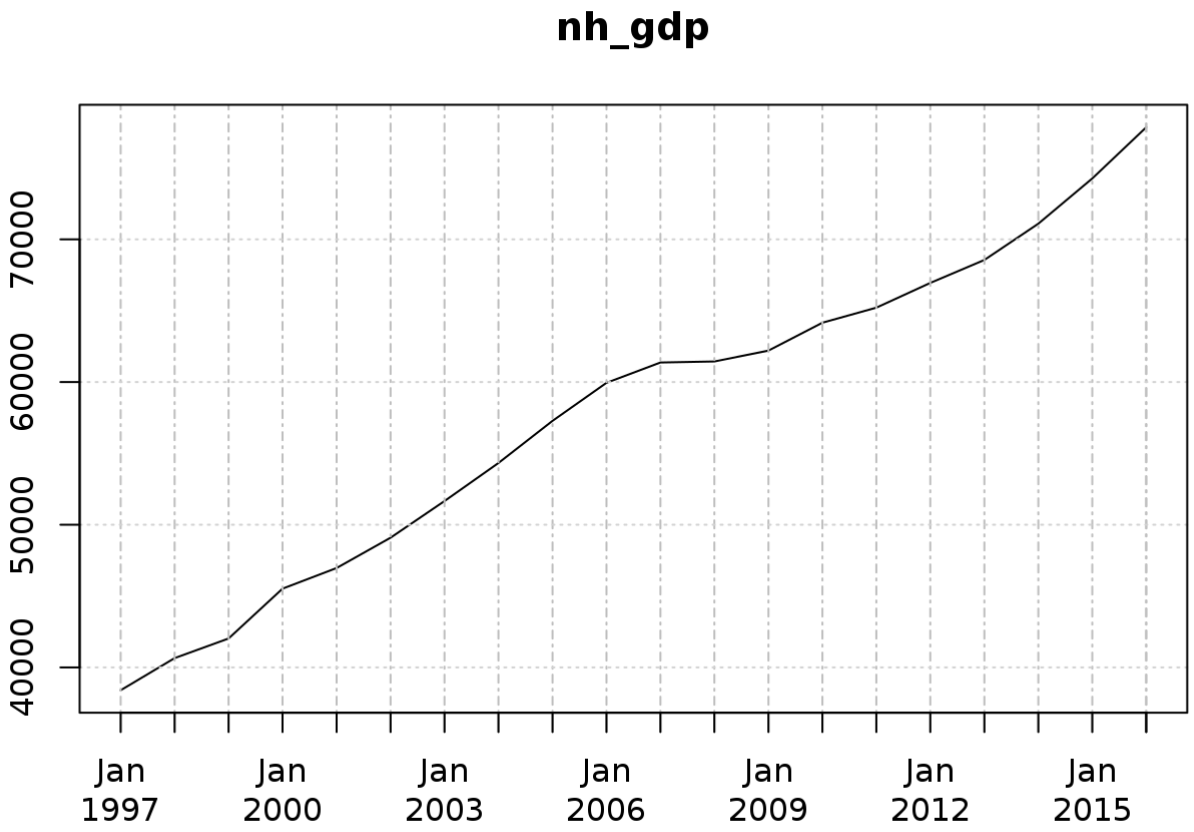
Retrieve the data for the period between 2010-12-3 and 2013-12-31, using the packages and functions you learned in this course. And plot them using the plot function.

1. Apple stock price (adjusted close)

```
apple_stock <- getSymbols("AAPL", from = "2010-12-31", to = "2013-12-31", auto.assign = FALSE)

head(apple_stock)
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume
## 2010-12-31    41.669    41.738   41.458   41.61895    48377000
## 2011-01-03    42.016    42.612   41.913   42.52343   111284600
## 2011-01-04    42.894    42.901   42.340   42.74536    77270200
## 2011-01-05    42.521    43.139   42.514   43.09501    63879900
## 2011-01-06    43.188    43.256   42.953   43.06018    75107200
## 2011-01-07    43.094    43.398   42.824   43.36855    77982800
##            AAPL.Adjusted
## 2010-12-31      46.08000
## 2011-01-03      47.08143
## 2011-01-04      47.32715
## 2011-01-05      47.71429
## 2011-01-06      47.67571
## 2011-01-07      48.01714

plot(apple_stock)
```

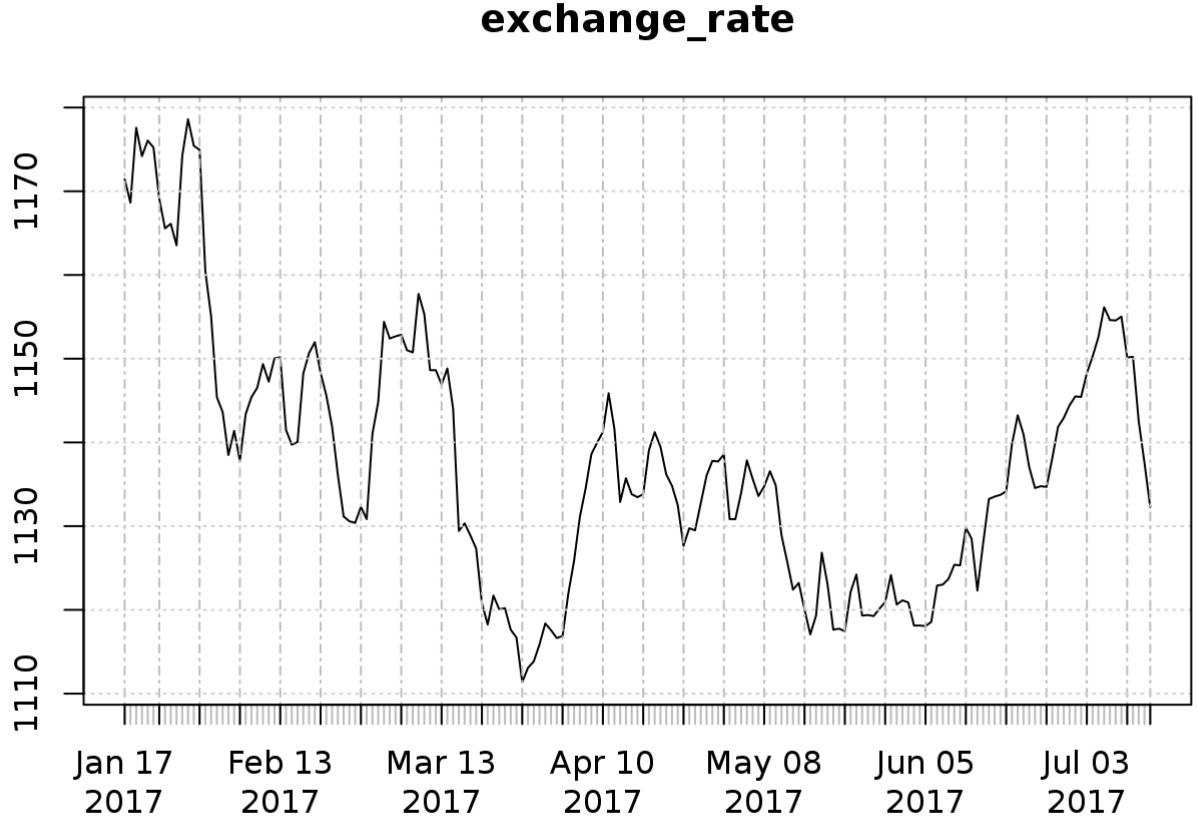
2. New Hampshire gross domestic product

```
nh_gdp <- getSymbols("NHNGSP", src = "FRED", from = "2010-12-31", to = "2013-12-31", auto.assign = FALSE)

head(nh_gdp)
##            NHNGSP
## 1997-01-01  38400
## 1998-01-01  40652
## 1999-01-01  42027
## 2000-01-01  45519
## 2001-01-01  46967
## 2002-01-01  49091

plot(nh_gdp)
```



3. exchange rate, US Dollar per South Korean Won (for the past 180 days)

```
exchange_rate <- getSymbols("USD/KRW", src = "oanda", auto.assign = FALSE)

head(exchange_rate)
##             USD.KRW
## 2017-01-17 1171.440
## 2017-01-18 1168.635
## 2017-01-19 1177.580
## 2017-01-20 1174.195
## 2017-01-21 1176.055
## 2017-01-22 1175.230

plot(exchange_rate)
```


exchange_rate
1110
1130
1150
1170
Jan 17
2017
Feb 13
2017
Mar 13
2017
Apr 10
2017
May 08
2017
Jun 05
2017
Jul 03
2017