CFRM 420 Lecture Notes

Nam Lee

2020-07-10

Contents

Course : CFRM 420 (Introduction to Computational Finance and Financial Econometrics)

Instructor: Nam Lee (elhmn@uw.edu)

Teaching Assistant: To be arranged

Class schedule: Monday & Wednesday 5:00 PM – 6:20 PM (80 minutes) at Loew Hall 216

Final Exam: Mon, Dec 14 6:30 PM ends prior to 8:20 PM (<110 minutes) at Loew Hall 216 (Tentative)

Office hours:

- Instructor:
  - a weekly virtual meeting must be
    - through https://washington.zoom.us/j/9650024596
    - between 5 PM and 6 PM on Thursday
- TA :
  - for virtual meeting, use:
    - through https://washington.zoom.us/
    - (Tuesday 1:00 PM - 3:00 PM) for both virtual and in-person meeting

Course description: This course is an introduction to the main statistical methods used in financial data science. The emphasis is evenly divided between the analysis and implementation of these methods and their applications in the realm of finance. Topics to be covered are: financial data and asset returns, linear regression, non-linear regression, non-parametric regression, ARIMA (Autoregressive integrated moving average) models, and GARCH (Generalize autoregressive conditional heteroscedasticity) models. Depending on the progress of the course, some or all of the following topics will also be discussed: cointegration analysis, factor models (including CAPM), and PCA (principal component analysis). During the course, students will also develop a working knowledge of the statistical software R.

Prerequisites: Students are assumed to have knowledge of multivariate calculus and linear algebra at the level of CFRM 405, and probability and statistics at the level of CFRM 410. Prior experience with R is not needed (though it is helpful). To check if you have the necessary background, please take the "self assessment test" that is posted on the Canvas course website (see below for explanation on how to access the website).

Textbook: The required textbook is Statistics and Data Analysis for Financial Engineering with R Examples, by David Ruppert and David Matteson, second edition (Springer 2015). The electronic version of the book is available at no charge from the publisher (Check UW library Webpage)

The textbook has a website with many useful materials, such as answers to selected problems. See: https://people.orie.cornell.edu/davidr/SDAFE2/index.html

Course Website: The course Canvas website is accessible at https://canvas.uw.edu/ (UW credentials needed). If you are registered in the course, you should see a link to the course website. The instructor will post announcements, assignments, and other course materials on the course webpage.

Discussion forum: There is a discussion forum on the website for you to ask questions and see/respond to questions that other student has asked. Please be courteous and considerate when posting comments, and keep the forum friendly and collaborative such that all students feel welcome to use it. The instructor will regularly monitor the forum.

Honor Code: All students are expected to comply with the usual standards for academic conduct as outlined in the CFRM Student Honor Code. This Honor Code is available in the "Files" section of the Canvas website.

Religious Accommodations: Washington state law requires that UW develop a policy for accommodation of student absences or significant hardship due to reasons of faith or conscience, or for organized religious activities. The UW's policy, including more information about how to request an accommodation, is available at Religious Accommodations Policy (https://registrar.washington.edu/staffandfaculty/religious-accommodations-policy/). Accommodations must be requested within the first two weeks of this course using the Religious Accommodations Request form (https://registrar.washington.edu/students/religious-accommodations-request/).

Coursework: The coursework consists of the following components:

Ingredient	Date	% of Course Grade
Homework	See the course schedule	40%
Midterm	Nov 04, 2020	30%
Final	Dec 13, 2020	30%

- Homework: There will be 8 homework sets. Each homework has equal contribution to the total grade, regardless of the total points assigned to it. Homework sets are posted on Canvas, and they are to be submitted online through Canvas as well. Late or non-submission of homework sets is not acceptable under any circumstance, and a zero grade will be assigned if this happens.
- Participation bonus: To encourage students to participate in the end-of-class course evaluation questionnaire, your lowest homework grade will be dropped if at least 80% of the students respond to the questionnaire. In such case, only your highest 7 homework grades will be counted towards the course grade.
- Midterm and Final Exams: The exams are closed-book. Make-up exams will be given at the discretion of the instructor and only in case of a justifiable reason, such as a medical or family emergency that is communicated with the instructor in a timely manner.
- Exam for online students: Local online students may take exams on the UW Seattle campus if there is space available in the classroom. Check with the TA to confirm your seat. Otherwise, online exams must be proctored. Midterms and finals must be completed on the same day and (if possible) at the same time as the in-class course. If the indicated schedule cannot be arranged, contact the professor or TA to discuss an alternative time. For reference, Seattle is in the Pacific time zone and observes daylight saving time. Please refer to http://cfrm.uw.edu/info/exam-proctors/
- UW grades: A linear grade scale will be used to convert percentage course grades into UW grades. That is, the posted course grade will be the percentage course grade divided by 25, and then rounded up to the nearest tenth.

Tentative Course Schedule

Week	Date	Lecture	Topic	Chapter	Homework
1	Sep 26	1	Financial Data and Asset Returns	2	
2	Oct 1	2	Basic Regression	9	HW 1 Posted
2	Oct 3	3	Basic Regression	9	

Week	Date	Lecture	Topic	Chapter	Homework
3	Oct 8	4	Basic Regression	9	HW 1 Due, HW
3	Oct 10	5	Regression Diagnostics	10	
4	Oct 15	6	Non-linear Regression	11	HW 2 Due, HW
4	Oct 17	7	Non-parametric Regression	21	
5	Oct 22	8	ARIMA Models	12	HW 3 Due, HW
5	Oct 24	9	ARIMA Models	12	
6	Oct 29	10	ARIMA Models	12	HW 4 Due
6	Oct 31	-	Review for the Midterm Exam		
7	Nov 5	-	Midterm Exam (during the regular class time)		
7	7	11	GARCH Models	14	HW 5 Posted
8	12	12	Cointegration Analysis	15	
8	14	13	PCA and factor models	18	HW 5 Due, HW
9	19	14	PCA and factor models	18	
9	21	15	Special Topics	-	HW 6 Due, HW
10	26	16	Special Topics	-	
10	28	17	Thanksgiving Holiday		HW 7 Due*, H
11	Dec 3*	18	Special Topics	-	
11	Dec 5	-	Review for Final Exam		HW 8 Due

Chapter 1

Intro Class

- Chapters 1, 2, and 4:
  - Obtaining financial data;
  - principles of financial data science;
  - exploratory (or empirical) data analysis,
  - calculating asset returns

1.1 Getting Acquainted with R and RStudio

R is a programming language for statistical computing and graphics. It is widely used among statisticians and contains a large number of ready-to-use functions (called packages).

Start by installing R. Please visit:

https://www.r-project.org/.

For Jupyter Notebook style, visit:

colab.fan/r

This will install the R software. It is also recommended to install an IDE (integrated development environment) for R, such as RStudio. Please visit:

https://www.rstudio.com/products/rstudio/download/.

1.1.1 Install "Quantmod"

There are many online resources to learn R. I will briefly explain the codes presented in each class. However, you should thoroughly examine each code on your own, and use the R documentation to learn how the are used.

Let us open RStudio, and install our first package. As mentioned before, packages include ready-to-use functions. quantmod is a package that can be used for obtaining financial data.

To install the package use Tools -> Install Packages... menu in RStudio, or directly use the following command:

```r
install.packages("quantmod")
```

1.1.2 Obtaining financial data

However, to be able to use an installed package, we need to load it. This is done by the following command.

```r
library('quantmod')
```

As you can see, R automatically load other packages that quantmod relies on, namely, packages xtx and zoo (both are packages that define time-series data types).

quantmod provides the function getSymbols() which can be used to find historical financial and economical data from various online resources.

Let us load historical data from Apple Inc. from Yahoo! finance. First, we need to find the symbol for apple. Searching the web, we find the symbol at

https://finance.yahoo.com/quote/AAPL/

Thus, the symbol is AAPL. Now, we can use the following R command to obtain the data.

```r
getSymbols("AAPL", from="2007-01-01", to="2018-05-30", src="yahoo")
```

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

We have obtained the historical data of symbol AAPL, from January 2007 until May 2018, using the data source yahoo (Yahoo! finance), and store it as a variable called AAPL.

```r
class(AAPL)
```

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

The variable is an extended time series (xts) object, which is a subclass of type zoo. These are essentially 2 dimensional arrays, where the index rows are date/time objects.

To get the total number of rows and columns, we can use:

```
dim(AAPL)
```

```
## [1] 2871    6
```

So, AAPL has 2871 rows (trading days) and 6 columns.

```
names(AAPL)
```

```
## [1] "AAPL.Open"     "AAPL.High"     "AAPL.Low"      "AAPL.Close"
## [5] "AAPL.Volume"   "AAPL.Adjusted"
```

- open price (the first trade of the day),
- high price (the highest price of the day),
- low price (the lowest price of the day),
- close price (the last trade of the day),
- volume (the total number of shares traded during the day),
- the adjusted prices (the price that is adjusted to include the effect of corporate events such as dividend payments and stock splits, more on this later)

1.1.3 Subset by Position

To get the first few rows of the xts object, we can use the function head(), and to get the last few rows, we can use the function tails().

```
head(AAPL,n=5)
```

```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2007-01-03  12.32714  12.36857 11.70000   11.97143   309579900      10.36364
## 2007-01-04  12.00714  12.27857 11.97429   12.23714   211815100      10.59366
## 2007-01-05  12.25286  12.31428 12.05714   12.15000   208685400      10.51822
## 2007-01-08  12.28000  12.36143 12.18286   12.21000   199276700      10.57016
## 2007-01-09  12.35000  13.28286 12.16429   13.22429   837324600      11.44823
```

```
tail(AAPL,n=5)
```

```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2018-05-22    188.38    188.88   186.78     187.16    15240700      182.1795
## 2018-05-23    186.35    188.50   185.76     188.36    20058400      183.3476
## 2018-05-24    188.77    188.84   186.21     188.15    23234000      183.1432
## 2018-05-25    188.23    189.65   187.65     188.58    17461000      183.5618
## 2018-05-29    187.60    188.75   186.87     187.90    22514100      182.8999
```

1.1.4 Subset by Date Range

```
AAPL["2015-12-23/2015-12-24"]
```

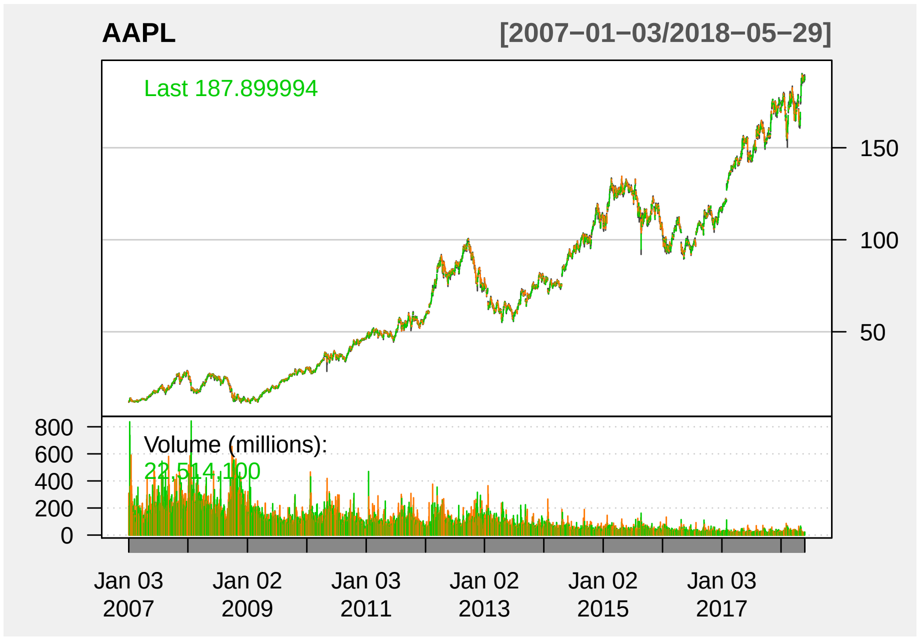
```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2015-12-23    107.27    108.85   107.20     108.61    32657400      100.9794
## 2015-12-24    109.00    109.00   107.95     108.03    13570400      100.4401
```

```
AAPL['2015-12']
```

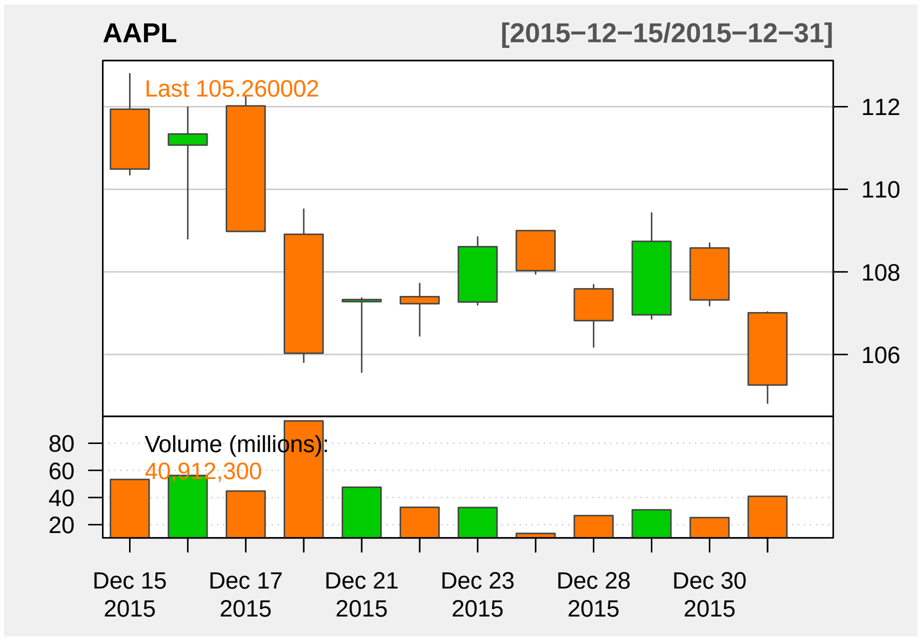
```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2015-12-01    118.75    118.81   116.86     117.34    34852400     109.09604
## 2015-12-02    117.34    118.11   116.08     116.28    33386600     108.11053
## 2015-12-03    116.55    116.79   114.22     115.20    41569500     107.10641
## 2015-12-04    115.29    119.25   115.11     119.03    57777000     110.66733
## 2015-12-07    118.98    119.86   117.81     118.28    32084200     109.97002
## 2015-12-08    117.52    118.60   116.86     118.23    34309500     109.92352
## 2015-12-09    117.64    117.69   115.08     115.62    46361400     107.49690
## 2015-12-10    116.04    116.94   115.51     116.17    29212700     108.00826
## 2015-12-11    115.19    115.39   112.85     113.18    46886200     105.22832
## 2015-12-14    112.18    112.68   109.79     112.48    64318700     104.57750
## 2015-12-15    111.94    112.80   110.35     110.49    53323100     102.72733
## 2015-12-16    111.07    111.99   108.80     111.34    56238500     103.51762
## 2015-12-17    112.02    112.25   108.98     108.98    44772800     101.32341
## 2015-12-18    108.91    109.52   105.81     106.03    96453300      98.58065
## 2015-12-21    107.28    107.37   105.57     107.33    47590600      99.78932
## 2015-12-22    107.40    107.72   106.45     107.23    32789400      99.69637
## 2015-12-23    107.27    108.85   107.20     108.61    32657400     100.97939
## 2015-12-24    109.00    109.00   107.95     108.03    13570400     100.44014
## 2015-12-28    107.59    107.69   106.18     106.82    26704200      99.31517
## 2015-12-29    106.96    109.43   106.86     108.74    30931200     101.10027
## 2015-12-30    108.58    108.70   107.18     107.32    25213800      99.78002
## 2015-12-31    107.01    107.03   104.82     105.26    40912300      97.86475
```

1.1.5 Visualize Data as Timeseries

```r
chartSeries(AAPL, theme="white", echo=TRUE)
```



```r
chartSeries(AAPL, theme="white", subset="2015-12-15::2015-12-31")
```



1.1.6 Understanding chartSeries

- What does chartSeries do?

```
?chartSeries
```

- What is **OHLC** charts?

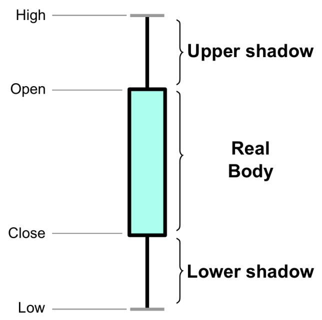


Figure 1.1: OHLC

1.1.7 Obtain Unemployment Data

We can use other sources to obtain data. For example, let us obtain monthly unemployment rate from Federal Reserve Economic Data (FRED).

We may find the symbol for unemployment rate in the following page

https://fred.stlouisfed.org/series/UNRATE,

which reveals that the symbol is UNRATE.

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

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

1.1.8 Visualize Unemployment Data

```
dim(UNRATE)
```

```
## [1] 870   1
```

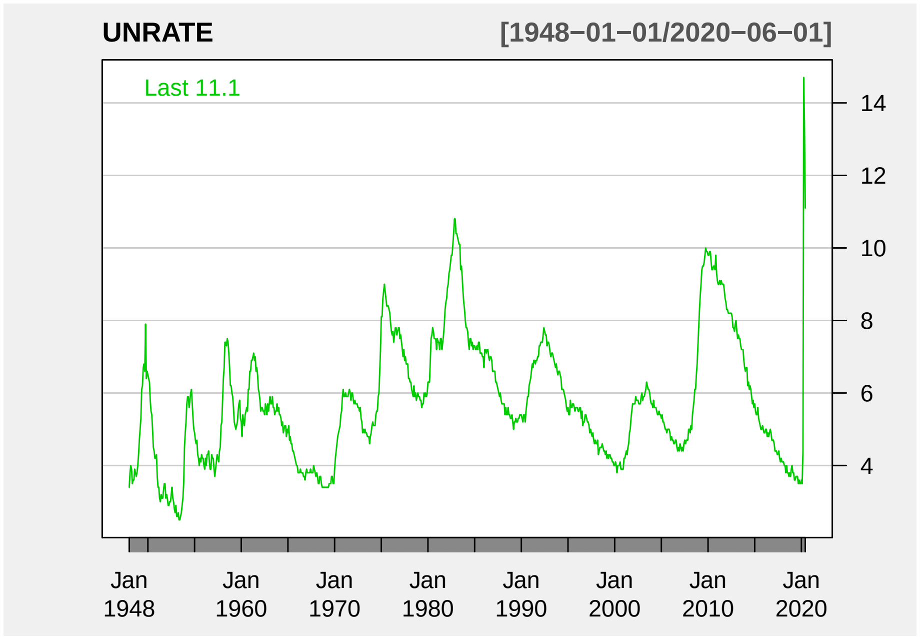
```
head(UNRATE)
```

```
##            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
```

```
tail(UNRATE)
```

```
##            UNRATE
## 2020-01-01    3.6
## 2020-02-01    3.5
## 2020-03-01    4.4
## 2020-04-01   14.7
## 2020-05-01   13.3
## 2020-06-01   11.1
```

```
chartSeries(UNRATE, theme="white")
```



1.2 Asset Return

1.2.1 Single Period Return

- Let $P_0, P_1, \ldots$ be the price of an equity. For now, let's assume no "corporate actions" such as dividends, split, etc.
- Net return over $[t-1, t]$:

$$R_t := (P_t - P_{t-1})/P_{t-1}.$$

- Gross return over $[t-1, t]$ is

$$P_t/P_{t-1} = 1 + R_t$$

- Log returns over $[t-1, t]$:

$$r_t := \log(P_t/P_{t-1})$$

1.2.2 Multi Period Returns

- $R_t(k)$, the net return over $[t-k, t]$, is given by:

$$\begin{aligned} R_t(k) &:= \frac{P_t}{P_{t-k}} - 1 \\ &= (1+R_t)(1+R_{t-1})\cdots(1+R_{t-k+1}) - 1, \end{aligned}$$

- $r_t(k)$, the log return over $[t-k, t]$, is given by:

$$\begin{aligned} r_t(k) &:= \log \frac{P_t}{P_{t-k}} \\ &= r_t + r_{t-1} + \cdots + r_{t-k+1} \end{aligned}$$

1.2.3 Simple Example

The following table gives two daily (adjusted) closing price of Apple stock in Dec. 2015

Date

12/23

12/24

Price($)

104.56

104.00

- Gross (daily) return on 12/24 is

$$104/104.56 = 0.994644$$

- Net return on 12/24 is

$$0.994644 - 1 = -0.005356$$

- Log return on 12/24 is

$$\log(104.00) - \log(104.56) = -0.005370$$

1.2.4 Approximation Example

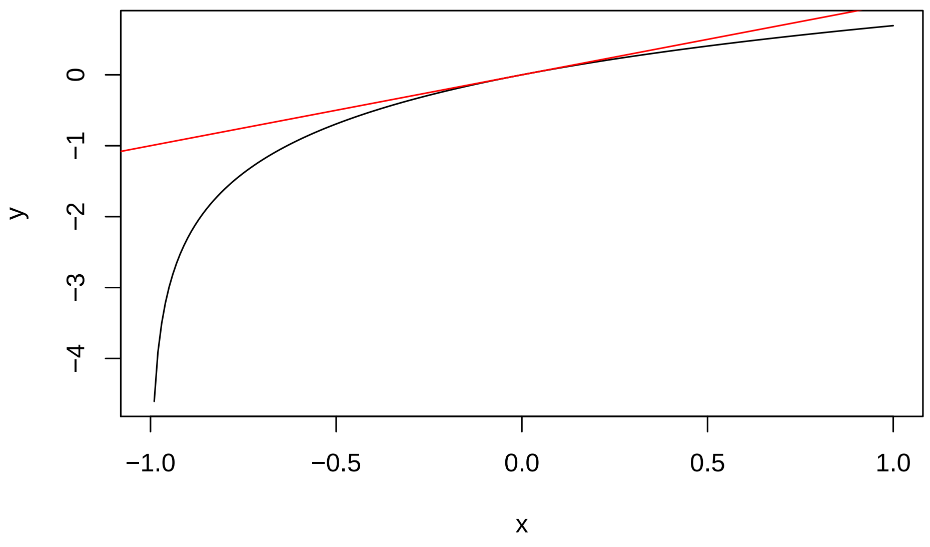
- For small $|x|$, we have

$$x \approx \log(1 + x),$$

  whence log and net returns are good approximation of each other

- Suppose that we use log returns instead of net returns.
  - How are our estimations? for losses and for gains?

```r
x=seq(-1,1,by=0.01)
y = log(1+x)
plot(x,y,type='l')
abline(a=0,b=1,col='red')
```



1.3 Adjusted Price

There are many market events that makes our return formula inappropriate for quoted prices, e.g. dividend payments, stock splits, etc.

1.3.1 Net Return with Dividend

Let D_1, D_2, ... be the dividends of a stock with quoted prices $P_1, P_2, \ldots$, where D_t is the dividend paid at t. The appropriate definition of the net return **should** be

$$R_t = \frac{P_t + D_t - P_{t-1}}{P_{t-1}}.$$

Similarly, the multi-period net return $R_t(k)$ **should** be

$$\begin{aligned}
&1 + R_t(k) \\
&= (1 + R_t)(1 + R_{t-1}) \ldots (1 + R_{t-k+1}) \\
&= \left(\frac{P_t + D_t}{P_{t-1}}\right)\left(\frac{P_{t-1} + D_{t-1}}{P_{t-2}}\right) \cdots \left(\frac{P_{t-k+1} + D_{t-k+1}}{P_{t-k}}\right) \\
&\neq \frac{P_t}{P_{t-k}}
\end{aligned}$$

1.3.2 The idea behind "adjusted prices"

Find prices $\widetilde{P}_{t_1}$, $\widetilde{P}_{t_2}$, ..., such that return are given by "simple" formulas applied to $\widetilde{P}_t$. In other words, we **want**

$$\begin{aligned}
1 + R_t &= \frac{\widetilde{P}_t}{\widetilde{P}_{t-1}} \\
\Longrightarrow \frac{P_t + D_t}{P_{t-1}} &= \frac{\widetilde{P}_t}{\widetilde{P}_{t-1}} \\
\Longrightarrow \widetilde{P}_t &= \left(\frac{P_t + D_t}{P_{t-1}}\right)\widetilde{P}_{t-1}
\end{aligned}$$

- There are many such adjustment for different types of events. Always use "adjust prices" to calculate returns.

1.3.3 Using Adjusted AAPL Prices

- Next, we show two ways to calculate asset returns using R. Let us first obtain the prices of Apple Inc.

```r
getSymbols("AAPL", from="2017-10-03", to="2017-10-13", src="yahoo")
```

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

```
names(AAPL)
```

```
## [1] "AAPL.Open"     "AAPL.High"     "AAPL.Low"      "AAPL.Close"
## [5] "AAPL.Volume"   "AAPL.Adjusted"
```

1.3.4 Adjusted Prices

As mentioned before, the adjusted price is the column AAPL.Adjusted. We can obtain the column by using the $ operator:

```
AAPL$AAPL.Adjusted
```

```
##            AAPL.Adjusted
## 2017-10-03      148.6490
## 2017-10-04      147.6867
## 2017-10-05      149.5246
## 2017-10-06      149.4380
## 2017-10-09      149.9576
## 2017-10-10      150.0153
## 2017-10-11      150.6408
## 2017-10-12      150.1116
```

1.3.5 Log and Net Return of AAPL

One can use the lag() and diff() functions (take a look at the R documentation to learn what they do).

```
P <- AAPL$AAPL.Adjusted
R <- diff(P)/lag(P)
class(R)
```

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

```
r <- diff(log(P))
class(r)
```

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

```r
head(r)
```

```
##            AAPL.Adjusted
## 2017-10-03            NA
## 2017-10-04 -0.0064943407
## 2017-10-05  0.0123677222
## 2017-10-06 -0.0005793969
## 2017-10-09  0.0034711500
## 2017-10-10  0.0003848614
```

```r
head(R)
```

```
##            AAPL.Adjusted
## 2017-10-03            NA
## 2017-10-04 -0.0064732981
## 2017-10-05  0.0124445188
## 2017-10-06 -0.0005792291
## 2017-10-09  0.0034771815
## 2017-10-10  0.0003849354
```

Let us calculate the net and log returns for a longer period and plot the result.

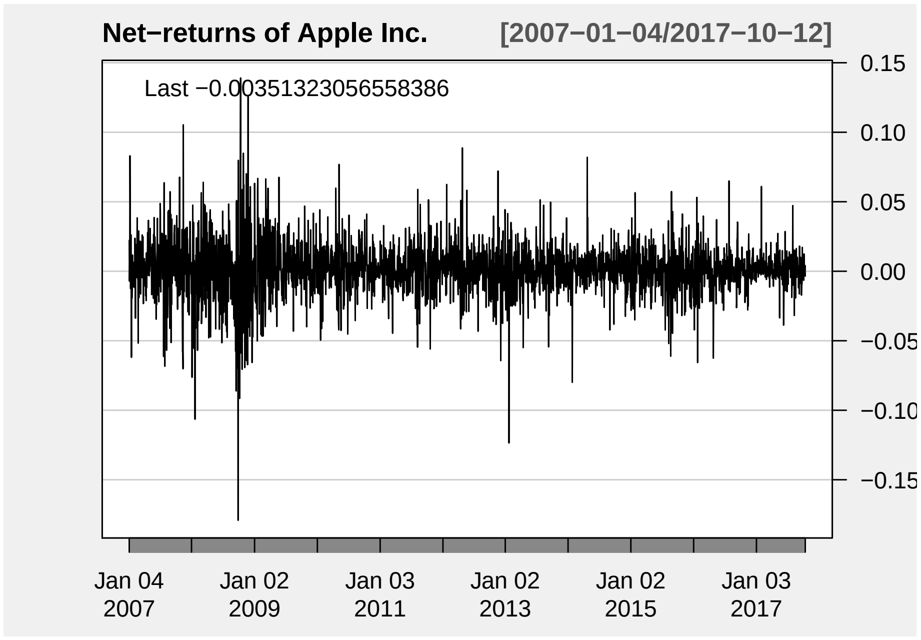
```r
getSymbols("AAPL", from="2007-01-03", to="2017-10-13", src="yahoo")
```

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

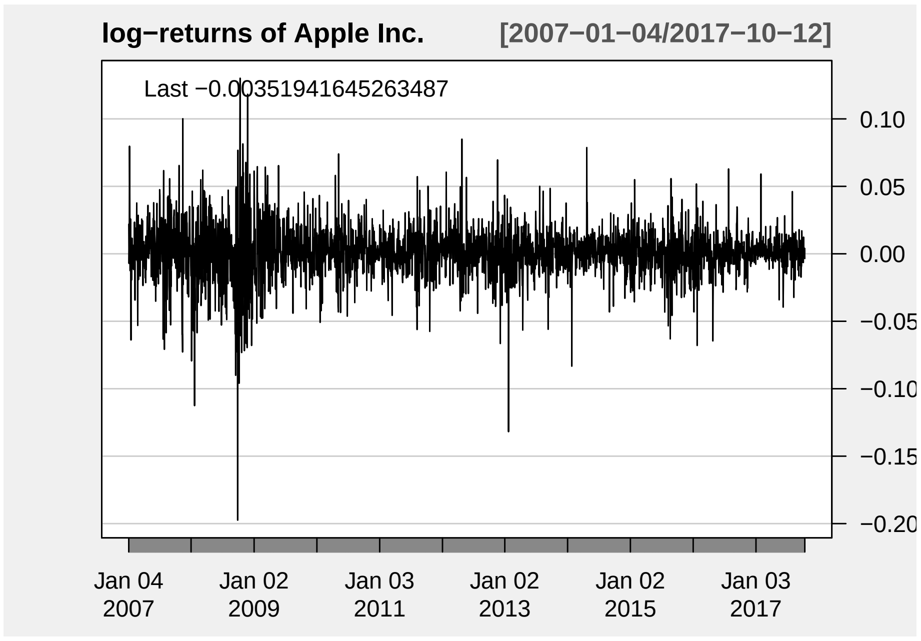
```r
AAPL.rtn = diff(AAPL$AAPL.Adjusted)/lag(AAPL$AAPL.Adjusted) # Compute net returns
colnames(AAPL.rtn)[1] = "net"
AAPL.rtn$log = diff(log(AAPL$AAPL.Adjusted)) # Compute log returns
head(AAPL.rtn)
```

```
##                     net          log
## 2007-01-03           NA           NA
## 2007-01-04  0.022195488  0.021952753
## 2007-01-05 -0.007121143 -0.007146620
## 2007-01-08  0.004938096  0.004925943
## 2007-01-09  0.083070321  0.079799898
## 2007-01-10  0.047855424  0.046745622
```

```r
chartSeries(AAPL.rtn$net, name = "Net-returns of Apple Inc.", theme=chartTheme("white", up.col='b
```



```
chartSeries(AAPL.rtn$log, name = "log-returns of Apple Inc.", theme=chartTheme("white"
```


log−returns of Apple Inc.
[2007−01−04/2017−10−12]
Last −0.00351941645263487
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
Jan 04 2007
Jan 02 2009
Jan 03 2011
Jan 02 2013
Jan 02 2015
Jan 03 2017