Adil Gokturk_HW1R

February 15, 2020

0.0.1 Adil Gokturk

0.1 HW1 Estimating Volatility from Historical Data

Set the working directory and load the libraries

[6]:
```
setwd('C:/Users/hgokturk/Desktop/Spring2020/FIN659/Assignments/hw1')
getwd()
```

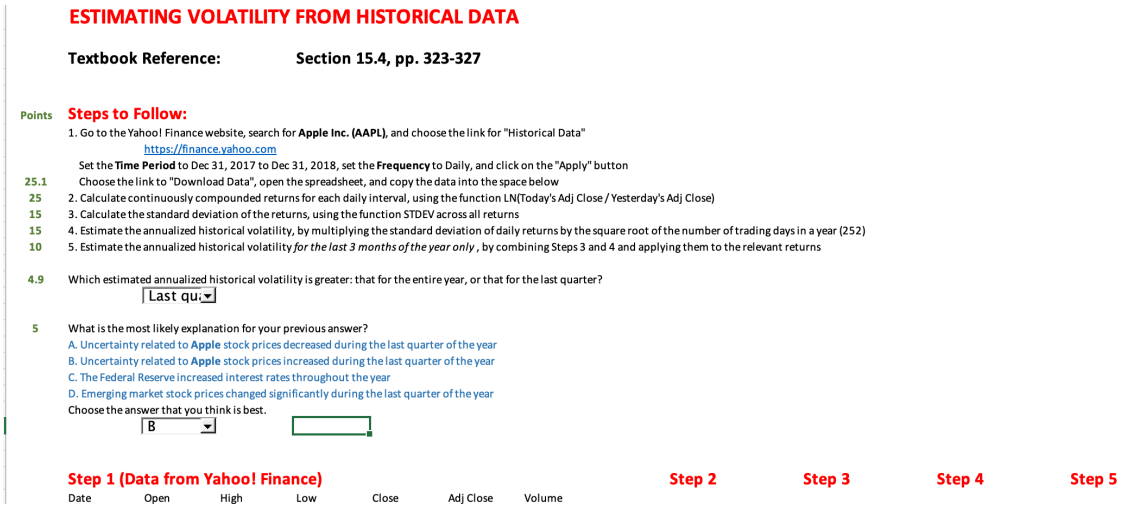
'C:/Users/hgokturk/Desktop/Spring2020/FIN659/Assignments/hw1'

[7]:
```
# Load libraries
library(ggplot2)
library(IRdisplay)
library(psych)
library(quantmod)
library(tidyverse)
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
Warning message:
"package 'psych' was built under R version 3.6.2"
Attaching package: 'psych'

The following objects are masked from 'package:ggplot2':

    %+%, alpha

Warning message:
"package 'quantmod' was built under R version 3.6.2"Loading required package:
xts
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':
```

```
    as.Date, as.Date.numeric

Loading required package: TTR
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
Version 0.4-0 included new data defaults. See ?getSymbols.
-- Attaching packages --------------------------------------- tidyverse 1.2.1 --
v tibble  2.1.1       v purrr   0.3.2
v tidyr   0.8.3       v dplyr   0.8.0.1
v readr   1.3.1       v stringr 1.4.0
v tibble  2.1.1       v forcats 0.4.0
Warning message:
"package 'stringr' was built under R version 3.6.2"-- Conflicts
------------------------------------------ tidyverse_conflicts() --
x psych::%+%()    masks ggplot2::%+%()
x psych::alpha()  masks ggplot2::alpha()
x dplyr::filter() masks stats::filter()
x dplyr::first()  masks xts::first()
x dplyr::lag()    masks stats::lag()
x dplyr::last()   masks xts::last()
x purrr::reduce() masks rugarch::reduce()
```

[8]:
```
# Display HW1
display_png(file = "hw1.png")
```



ESTIMATING VOLATILITY FROM HISTORICAL DATA

Textbook Reference: **Section 15.4, pp. 323-327**

Points	**Steps to Follow:**
	1. Go to the Yahoo! Finance website, search for **Apple Inc. (AAPL)**, and choose the link for "Historical Data"
	https://finance.yahoo.com
	Set the **Time Period** to Dec 31, 2017 to Dec 31, 2018, set the **Frequency** to Daily, and click on the "Apply" button
25.1	Choose the link to "Download Data", open the spreadsheet, and copy the data into the space below
25	2. Calculate continuously compounded returns for each daily interval, using the function LN(Today's Adj Close / Yesterday's Adj Close)
15	3. Calculate the standard deviation of the returns, using the function STDEV across all returns
15	4. Estimate the annualized historical volatility, by multiplying the standard deviation of daily returns by the square root of the number of trading days in a year (252)
10	5. Estimate the annualized historical volatility *for the last 3 months of the year only* , by combining Steps 3 and 4 and applying them to the relevant returns
4.9	Which estimated annualized historical volatility is greater: that for the entire year, or that for the last quarter?
	Last qu:
5	What is the most likely explanation for your previous answer?
	A. Uncertainty related to **Apple** stock prices decreased during the last quarter of the year
	B. Uncertainty related to **Apple** stock prices increased during the last quarter of the year
	C. The Federal Reserve increased interest rates throughout the year
	D. Emerging market stock prices changed significantly during the last quarter of the year
	Choose the answer that you think is best.
	B

Step 1 (Data from Yahoo! Finance) **Step 2** **Step 3** **Step 4** **Step 5**

Date Open High Low Close Adj Close Volume

0.2 1. Go to the Yahoo! Finance website, search for Apple Inc. (AAPL), and download the "Historical Data"

Set the Time Period to __Dec 31, 2017 to Dec 31, 2018,__ set the Frequency to Daily, and click
Choose the link to "Download Data", open the spreadsheet, and copy the data into the space bel

https://finance.yahoo.com

I used quantmod package to get data and set the date from="2018-1-1", to="2019-1-1" to get all the 2018 trading days

[9]:
```
getSymbols(Symbols = "AAPL", from="2018-1-1", to="2019-1-1")
```

```
'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.
```

'AAPL'

[10]:
```
length(AAPL$AAPL.Adjusted)
head(AAPL)
```

251

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2018-01-02	170.16	172.30	169.26	172.26	25555900	166.8040
2018-01-03	172.53	174.55	171.96	172.23	29517900	166.7750
2018-01-04	172.54	173.47	172.08	173.03	22434600	167.5496
2018-01-05	173.44	175.37	173.05	175.00	23660000	169.4572
2018-01-08	174.35	175.61	173.93	174.35	20567800	168.8278
2018-01-09	174.55	175.06	173.41	174.33	21584000	168.8085

[11]:
```
tail(AAPL)
```

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2018-12-21	156.86	158.16	149.63	150.73	95744600	148.1472
2018-12-24	148.15	151.55	146.59	146.83	37169200	144.3140
2018-12-26	148.30	157.23	146.72	157.17	58582500	154.4769
2018-12-27	155.84	156.77	150.07	156.15	53117100	153.4743
2018-12-28	157.50	158.52	154.55	156.23	42291400	153.5530
2018-12-31	158.53	159.36	156.48	157.74	35003500	155.0371

0.2.1 2. Calculate continuously compounded returns for each daily interval, using the function LN(Today's Adj Close / Yesterday's Adj Close)

[12]:
```
apple2018 <- dailyReturn(x = AAPL$AAPL.Adjusted) %>% round(4)
```

[13]:
```
# let's take a look at the first 6 days' return
(apple2018)
```

	daily.returns
2018-01-02	0.0000
2018-01-03	-0.0002
2018-01-04	0.0046
2018-01-05	0.0114
2018-01-08	-0.0037
2018-01-09	-0.0001
2018-01-10	-0.0002
2018-01-11	0.0057
2018-01-12	0.0103
2018-01-16	-0.0051
2018-01-17	0.0165
2018-01-18	0.0009
2018-01-19	-0.0045
2018-01-22	-0.0082
2018-01-23	0.0002
2018-01-24	-0.0159
2018-01-25	-0.0179
2018-01-26	0.0023
2018-01-29	-0.0207
2018-01-30	-0.0059
2018-01-31	0.0028
2018-02-01	0.0021
2018-02-02	-0.0434
2018-02-05	-0.0250
2018-02-06	0.0418
2018-02-07	-0.0214
2018-02-08	-0.0275
2018-02-09	0.0122
2018-02-12	0.0403
2018-02-13	0.0100
2018-02-14	0.0184
2018-02-15	0.0336
2018-02-16	-0.0032
2018-02-20	-0.0034
2018-02-21	-0.0045
2018-02-22	0.0084
2018-02-23	0.0174
2018-02-26	0.0198
2018-02-27	-0.0032
2018-02-28	-0.0015
2018-03-01	-0.0175
2018-03-02	0.0069
2018-03-05	0.0035

2018-03-06	-0.0008
2018-03-07	-0.0093
2018-03-08	0.0109
2018-03-09	0.0172
2018-03-12	0.0097
2018-03-13	-0.0096
2018-03-14	-0.0085
2018-03-15	0.0012
2018-03-16	-0.0035
2018-03-19	-0.0153
2018-03-20	-0.0003
2018-03-21	-0.0227
2018-03-22	-0.0141
2018-03-23	-0.0232
2018-03-26	0.0475
2018-03-27	-0.0256
2018-03-28	-0.0110
2018-03-29	0.0078
2018-04-02	-0.0066
2018-04-03	0.0103
2018-04-04	0.0191
2018-04-05	0.0069
2018-04-06	-0.0256
2018-04-09	0.0099
2018-04-10	0.0188
2018-04-11	-0.0047
2018-04-12	0.0099
2018-04-13	0.0034
2018-04-16	0.0062
2018-04-17	0.0138
2018-04-18	-0.0022
2018-04-19	-0.0283
2018-04-20	-0.0410
2018-04-23	-0.0029
2018-04-24	-0.0139
2018-04-25	0.0044
2018-04-26	0.0035
2018-04-27	-0.0116
2018-04-30	0.0181
2018-05-01	0.0232
2018-05-02	0.0442
2018-05-03	0.0018
2018-05-04	0.0392
2018-05-07	0.0072
2018-05-08	0.0048
2018-05-09	0.0070
2018-05-10	0.0143
2018-05-11	-0.0038

2018-05-14	-0.0023
2018-05-15	-0.0091
2018-05-16	0.0093
2018-05-17	-0.0063
2018-05-18	-0.0036
2018-05-21	0.0071
2018-05-22	-0.0025
2018-05-23	0.0064
2018-05-24	-0.0011
2018-05-25	0.0023
2018-05-29	-0.0036
2018-05-30	-0.0021
2018-05-31	-0.0034
2018-06-01	0.0180
2018-06-04	0.0084
2018-06-05	0.0077
2018-06-06	0.0035
2018-06-07	-0.0027
2018-06-08	-0.0091
2018-06-11	-0.0025
2018-06-12	0.0055
2018-06-13	-0.0082
2018-06-14	0.0005
2018-06-15	-0.0103
2018-06-18	-0.0005
2018-06-19	-0.0162
2018-06-20	0.0044
2018-06-21	-0.0056
2018-06-22	-0.0029
2018-06-25	-0.0149
2018-06-26	0.0124
2018-06-27	-0.0015
2018-06-28	0.0073
2018-06-29	-0.0021
2018-07-02	0.0112
2018-07-03	-0.0174
2018-07-05	0.0080
2018-07-06	0.0139
2018-07-09	0.0139
2018-07-10	-0.0012
2018-07-11	-0.0130
2018-07-12	0.0168
2018-07-13	0.0016
2018-07-16	-0.0022
2018-07-17	0.0028
2018-07-18	-0.0055
2018-07-19	0.0078
2018-07-20	-0.0023

2018-07-23	0.0009
2018-07-24	0.0073
2018-07-25	0.0094
2018-07-26	-0.0031
2018-07-27	-0.0166
2018-07-30	-0.0056
2018-07-31	0.0020
2018-08-01	0.0589
2018-08-02	0.0292
2018-08-03	0.0029
2018-08-06	0.0052
2018-08-07	-0.0094
2018-08-08	0.0007
2018-08-09	0.0079
2018-08-10	-0.0030
2018-08-13	0.0065
2018-08-14	0.0042
2018-08-15	0.0023
2018-08-16	0.0146
2018-08-17	0.0200
2018-08-20	-0.0097
2018-08-21	-0.0019
2018-08-22	0.0000
2018-08-23	0.0020
2018-08-24	0.0031
2018-08-27	0.0082
2018-08-28	0.0081
2018-08-29	0.0149
2018-08-30	0.0092
2018-08-31	0.0116
2018-09-04	0.0032
2018-09-05	-0.0065
2018-09-06	-0.0166
2018-09-07	-0.0081
2018-09-10	-0.0134
2018-09-11	0.0253
2018-09-12	-0.0124
2018-09-13	0.0242
2018-09-14	-0.0114
2018-09-17	-0.0266
2018-09-18	0.0017
2018-09-19	0.0006
2018-09-20	0.0076
2018-09-21	-0.0108
2018-09-24	0.0144
2018-09-25	0.0063
2018-09-26	-0.0080
2018-09-27	0.0206

2018-09-28	0.0035
2018-10-01	0.0067
2018-10-02	0.0089
2018-10-03	0.0122
2018-10-04	-0.0176
2018-10-05	-0.0162
2018-10-08	-0.0023
2018-10-09	0.0139
2018-10-10	-0.0463
2018-10-11	-0.0088
2018-10-12	0.0357
2018-10-15	-0.0214
2018-10-16	0.0220
2018-10-17	-0.0043
2018-10-18	-0.0234
2018-10-19	0.0152
2018-10-22	0.0061
2018-10-23	0.0094
2018-10-24	-0.0343
2018-10-25	0.0219
2018-10-26	-0.0159
2018-10-29	-0.0188
2018-10-30	0.0050
2018-10-31	0.0261
2018-11-01	0.0154
2018-11-02	-0.0663
2018-11-05	-0.0284
2018-11-06	0.0108
2018-11-07	0.0303
2018-11-08	-0.0035
2018-11-09	-0.0193
2018-11-12	-0.0504
2018-11-13	-0.0100
2018-11-14	-0.0282
2018-11-15	0.0247
2018-11-16	0.0111
2018-11-19	-0.0396
2018-11-20	-0.0478
2018-11-21	-0.0011
2018-11-23	-0.0254
2018-11-26	0.0135
2018-11-27	-0.0022
2018-11-28	0.0385
2018-11-29	-0.0077
2018-11-30	-0.0054
2018-12-03	0.0349
2018-12-04	-0.0440
2018-12-06	-0.0111

```
2018-12-07       -0.0357
2018-12-10        0.0066
2018-12-11       -0.0057
2018-12-12        0.0028
2018-12-13        0.0109
2018-12-14       -0.0320
2018-12-17       -0.0093
2018-12-18        0.0130
2018-12-19       -0.0312
2018-12-20       -0.0252
2018-12-21       -0.0389
2018-12-24       -0.0259
2018-12-26        0.0704
2018-12-27       -0.0065
2018-12-28        0.0005
2018-12-31        0.0097
```

[14]:
```
# remove the ZERO valued First day form the data
apple2018 <- apple2018[-1,]
```

[15]:
```
# let's check it
head(apple2018)
```

```
           daily.returns
2018-01-03       -0.0002
2018-01-04        0.0046
2018-01-05        0.0114
2018-01-08       -0.0037
2018-01-09       -0.0001
2018-01-10       -0.0002
```

[16]:
```
tail(apple2018)
```

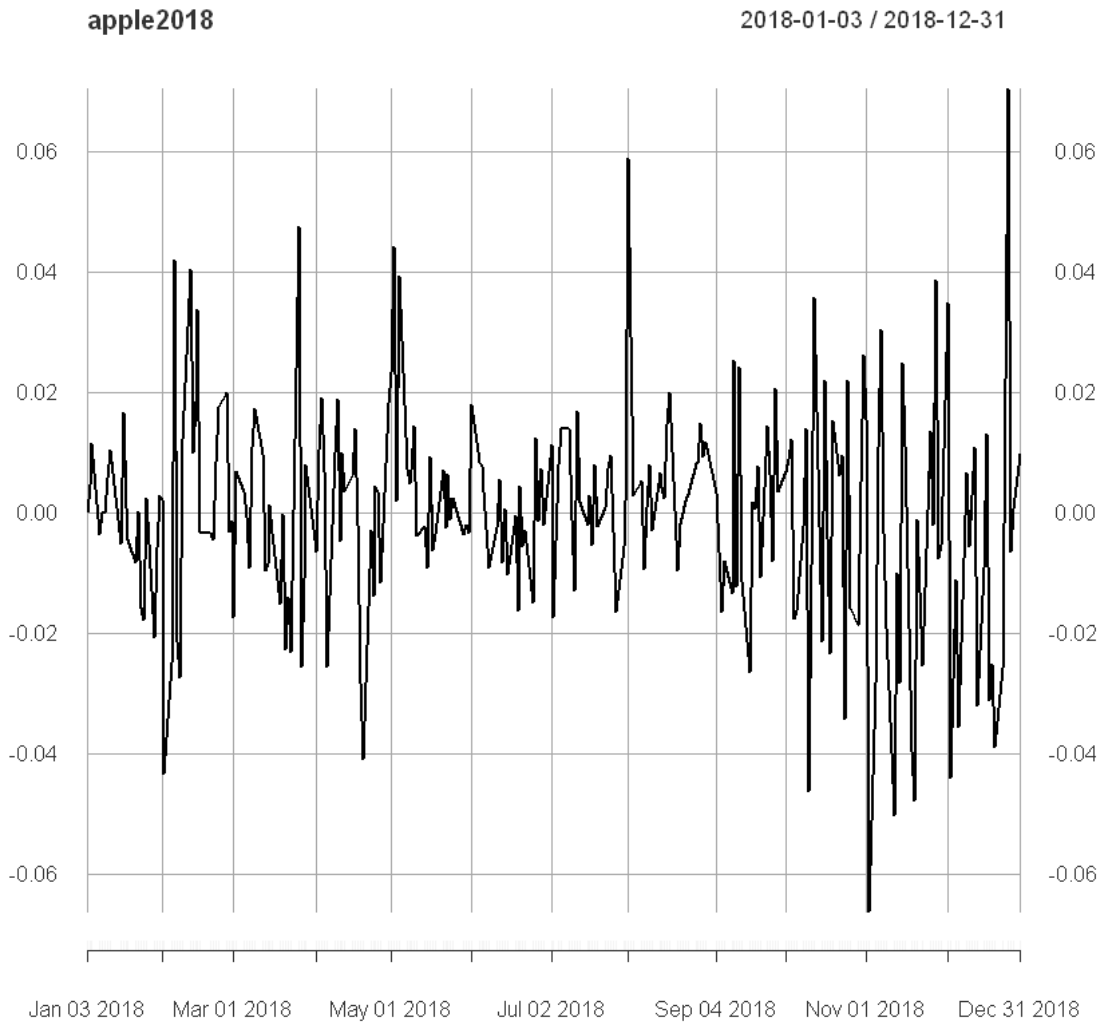
```
           daily.returns
2018-12-21       -0.0389
2018-12-24       -0.0259
2018-12-26        0.0704
2018-12-27       -0.0065
2018-12-28        0.0005
2018-12-31        0.0097
```

[17]:
```
# check the structure
str(apple2018)
```

```
An 'xts' object on 2018-01-03/2018-12-31 containing:
  Data: num [1:250, 1] -0.0002 0.0046 0.0114 -0.0037 -0.0001 -0.0002 0.0057
```

```
0.0103 -0.0051 0.0165 …
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr "daily.returns"
  Indexed by objects of class: [Date] TZ: UTC
  xts Attributes:
 NULL
```

[18]:
```
# Let's take a look at the daily volatility of the Apple stock price in 2018
plot(apple2018)
```



0.2.2 3. Calculate the standard deviation of the returns, using the function STDEV across all returns

[19]:
```
# standard deviation of the returns
apple2018.sd <- sd(apple2018)
apple2018.sd
(apple2018.sd *100) %>% round(2) # Percentage %
```

0.0181071952637973

1.81

0.2.3 Calculated standard deviation of the returns= 0.0181071952637973 or 1.81 %

0.2.4 4. Estimate the annualized historical volatility, by multiplying the standard deviation of daily returns by the square root of the number of trading days in a year (252)

[20]:
```
apple2018.hist.volatility <- apple2018.sd*sqrt(252)
apple2018.hist.volatility
(apple2018.hist.volatility*100) %>% round(2) # Percentage %
```

0.287442813653366

28.74

0.2.5 Annualized historical volatility = 28.74 %

It is slightly different than the solution of 28.78%!. I assume R still use more than 4 decimal to calculate.

0.2.6 5. Estimate the annualized historical volatility for the last 3 months of the year only, by combining Steps 3 and 4 and applying them to the relevant returns

[21]:
```
#  Let's get the last 3 months return data
length(apple2018) # 2018 data
length(apple2018[188:250,]) # last 3 months
head(apple2018[188:250,]) # Let's check it
tail(apple2018[188:250,])
```

250

63

	daily.returns
2018-10-01	0.0067
2018-10-02	0.0089
2018-10-03	0.0122
2018-10-04	-0.0176
2018-10-05	-0.0162
2018-10-08	-0.0023

```
            daily.returns
2018-12-21       -0.0389
2018-12-24       -0.0259
2018-12-26        0.0704
2018-12-27       -0.0065
2018-12-28        0.0005
2018-12-31        0.0097
```

[22]:
```
# Calculate the  annualized historical volatility for the last 3 months of the␣
↪year
(apple2018.quarter4.volatility <- sd(apple2018[188:250,])*sqrt(252))
(apple2018.quarter4.volatility *100) %>% round(2)
```

0.402237257862699

40.22

0.2.7 The annualized historical volatility for the last 3 months of the year = 40.22%

It is slightly different than the solution of 40.42! I assume R still use more than 4 decimal to calculate.

0.2.8 Which estimated annualized historical volatility is greater: that for the entire year, or that for the last quarter?

0.3 Last Quarter

0.3.1 What is the most likely explanation for your previous answer?

0.4 B. Uncertainty related to Apple stock prices increased during the last quarter of the year!

I found another R package–PerformanceAnalytics, I think it minimizes algebra usage in Financial analytics.