MSCA32001 Assignment1

Duo Zhou

6/23/2020

1. Read Chapter 1 of the book

2. Download and analyze excess returns of S&P 500

Calculate continuous daily excess returns of SP500("^GSPC") for the period from 1/1/2014 until 12/31/2014 using overnight Fed Funds rates as risk-free rates.

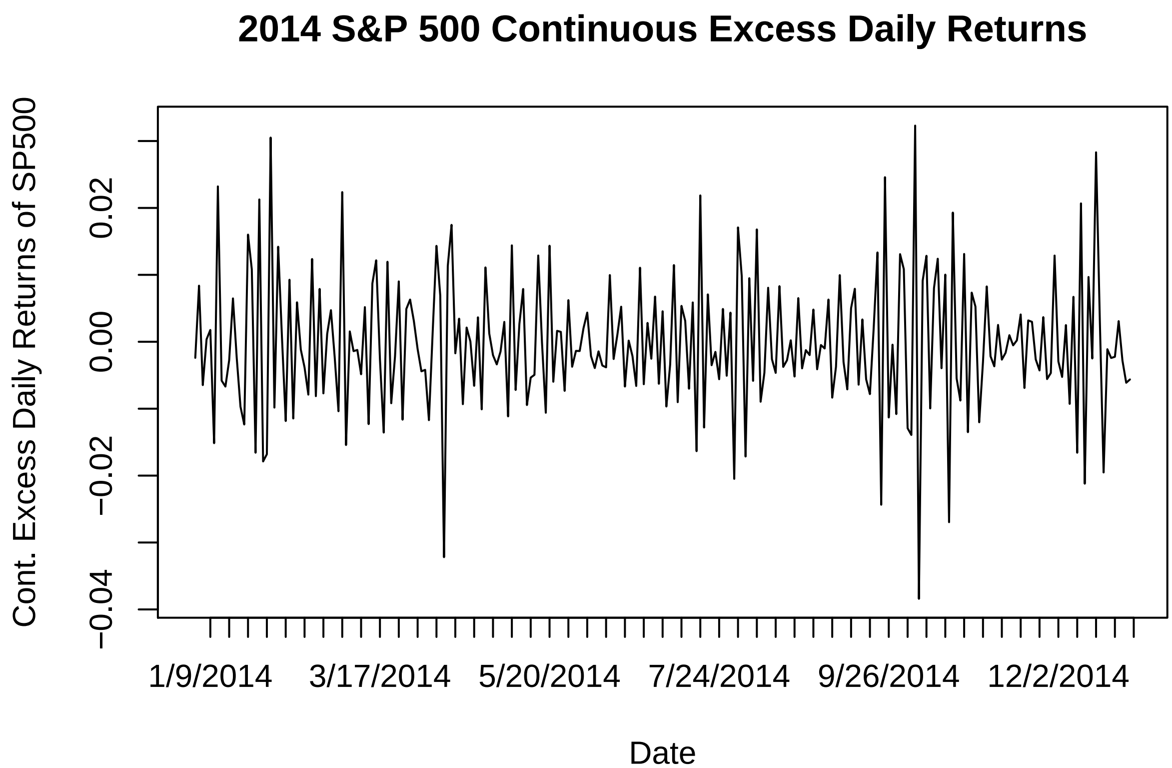
The risk free rate is compounded to the ith day using both log and cumulative product methods. Both methods should produce very similar results. SNP500 returns are not compounded. The daily returns of both risk free rate and SNP500 simple returns are calculated using the return difference of two adjacent days. The daily excess return are the difference between SNP500 and risk free rate.

```r
# read data
datapath<-"C:/Users/zd000/Desktop/MSCA/Financial Analytics/Assignments/week1/"
SNP500 <- read.csv(file = paste(datapath, 'SP500_NB2014.csv', sep='/'))
# Exam data
#head(SNP500)
#Calculate the continuous daily excess return using log formula

RIFSPFF.cont.return<-c()

for(i in 1:length(SNP500$SP500Returns)){

RIFSPFF.cont.return[i]<-log(prod(1+SNP500$RIFSPFF_N.B[1:i]/360)) # compounded log return of Risk Free
}
SNP500.daily.cont.excess.return<- diff(SNP500$SP500Returns)-diff(RIFSPFF.cont.return)
head(cbind(SNP500.Return.daily=diff(SNP500$SP500Returns),
  Risk.Free.Log.Compounded.daily=diff(RIFSPFF.cont.return),
  SNP500.daily.cont.excess.return=SNP500.daily.cont.excess.return))
```

```
##      SNP500.Return.daily Risk.Free.Log.Compounded.daily
## [1,]        -0.002181907                   0.0002221975
## [2,]         0.008578272                   0.0001944255
## [3,]        -0.006275577                   0.0001944255
## [4,]         0.000560481                   0.0001944255
## [5,]         0.001955781                   0.0001944255
## [6,]        -0.014959996                   0.0001944255
##      SNP500.daily.cont.excess.return
## [1,]                   -0.0024041045
## [2,]                    0.0083838465
## [3,]                   -0.0064700025
## [4,]                    0.0003660555
```

```
## [5,]                       0.0017613555
## [6,]                      -0.0151544215
```

```
plot(SNP500.daily.cont.excess.return, type="l", ylab="Cont. Excess Daily Returns of SP500", xlab = "Dat
axis(1, at=5*(1:50), labels=SNP500[5*(1:50),1])
```

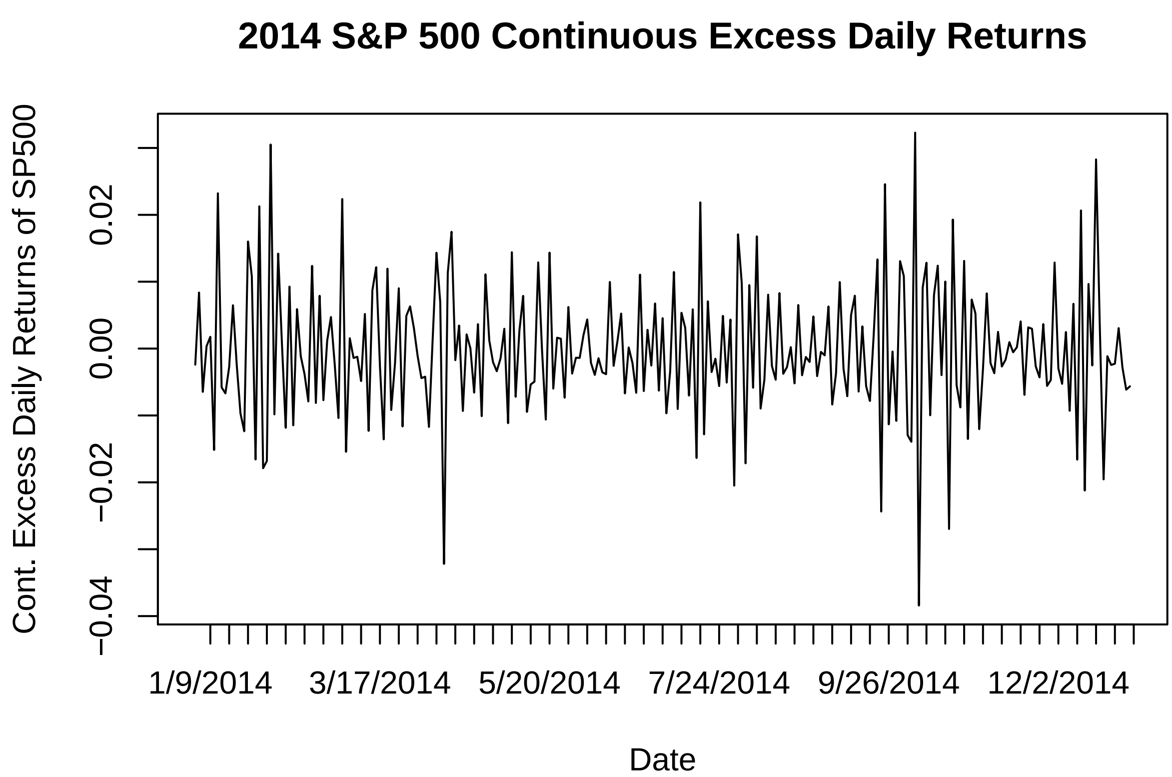



```
#Calculate the continuous daily excess return using cumulative product method

RIFSPFF.cont.return.1<-c()
for(i in 1:length(SNP500$SP500Returns)){

  RIFSPFF.cont.return.1[i]<-prod(1+SNP500$RIFSPFF_N.B[1:i]/360)-1 # RFrate Cont. Compounded Return

}

SNP500.daily.cont.excess.return.1<- diff(SNP500$SP500Returns)-diff(RIFSPFF.cont.return.1)
head(cbind(SNP500.Return.daily=diff(SNP500$SP500Returns),
  Risk.Free.Log.Compouned.daily=diff(RIFSPFF.cont.return.1),
  SNP500.daily.cont.excess.return=SNP500.daily.cont.excess.return.1))
```

```
##      SNP500.Return.daily Risk.Free.Log.Compouned.daily
## [1,]        -0.002181907                  0.0002222716
## [2,]         0.008578272                  0.0001945309
## [3,]        -0.006275577                  0.0001945687
## [4,]         0.000560481                  0.0001946065
```

```
## [5,]         0.001955781                  0.0001946444
## [6,]        -0.014959996                  0.0001946822
##       SNP500.daily.cont.excess.return
## [1,]                    -0.0024041786
## [2,]                     0.0083837411
## [3,]                    -0.0064701457
## [4,]                     0.0003658745
## [5,]                     0.0017611366
## [6,]                    -0.0151546782
```

```
plot(SNP500.daily.cont.excess.return.1, type="l", ylab="Cont. Excess Daily Returns of SP500", xlab = "Da
axis(1, at=5*(1:50), labels=SNP500[5*(1:50),1])
```




```
head(SNP500.daily.cont.excess.return)
```

```
## [1] -0.0024041045  0.0083838465 -0.0064700025  0.0003660555  0.0017613555
## [6] -0.0151544215
```

```
head(SNP500.daily.cont.excess.return.1)
```

```
## [1] -0.0024041786  0.0083837411 -0.0064701457  0.0003658745  0.0017611366
## [6] -0.0151546782
```

The SNP500 daily excess returns calculated using log and cumulative Formula are very similar as demonstrated above

3. Download and analyze exchange rates

Answer the same questions as in Exercise 5 on page 37 as a refresher of statistical analysis skills. Try to do it without using R demo code from the book

1. Download GBP/USD and USD/JPY exchange rates from Oanda using quantmode

```
#install.packages("quantmod")
suppressWarnings(library(quantmod))
```

```
## 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.
```

```
#Oanda only provides historical data for the past 180 days.
getSymbols("GBP/USD",src="oanda", from= "2019-12-30", to= Sys.Date())
```

```
## '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.

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

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

```
getSymbols("USD/JPY",src="oanda", from= "2019-12-30", to= Sys.Date())
```

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

```
## [1] "USD/JPY"
```

2. Calculate daily log returns of both exchange rates

```r
# Since Oanda only provides historical data for the past 180 days, I am going to use the data from the 
GBPUSD.daily.log.return <- diff(log(GBPUSD$GBP.USD))
GBPUSD.daily.log.return <- GBPUSD.daily.log.return [-1,]
USDJPY.daily.log.return <- diff(log(USDJPY$USD.JPY))
USDJPY.daily.log.return <- USDJPY.daily.log.return [-1,]
head(cbind(GBPUSD.daily.log.return=unname(GBPUSD.daily.log.return),
USDJPY.daily.log.return=unname(USDJPY.daily.log.return)))
```

```
##            GBPUSD.daily.log.return USDJPY.daily.log.return
## 2020-01-10           -0.0005799262            1.549204e-03
## 2020-01-11           -0.0002786065           -7.323218e-04
## 2020-01-12           -0.0001408636            5.884517e-05
## 2020-01-13           -0.0045204519            2.893470e-03
## 2020-01-14           -0.0002753706            2.138563e-03
## 2020-01-15            0.0018615052           -1.135079e-03
```

3. Calculate min, mean, sd, skewness, Kurtosis, and max of log returns for both exchange rates

```r
#install.packages("moments")
library(moments)
c(GBPUSD.min = min(GBPUSD.daily.log.return),
  GBPUSD.mean = mean(GBPUSD.daily.log.return),
  GBPUSD.sd = sd(GBPUSD.daily.log.return),
  GBPUSD.skew =unname(skewness(GBPUSD.daily.log.return)),
  GBPUSD.kurt = unname(kurtosis(GBPUSD.daily.log.return)),
  GBPUSD.max = max(GBPUSD.daily.log.return))
```

```
##    GBPUSD.min   GBPUSD.mean     GBPUSD.sd   GBPUSD.skew   GBPUSD.kurt 
## -0.0277770382 -0.0002540042  0.0057183820 -0.5948938180  8.8803129475 
##    GBPUSD.max 
##  0.0253022194
```

```r
c(USDJPY.min = min(USDJPY.daily.log.return),
  USDJPY.mean = mean(USDJPY.daily.log.return),
  USDJPY.sd = sd(USDJPY.daily.log.return),
  USDJPY.skew = unname(skewness(USDJPY.daily.log.return)),
  USDJPY.kurt = unname(kurtosis(USDJPY.daily.log.return)),
  USDJPY.max = max(USDJPY.daily.log.return))
```

```
##    USDJPY.min   USDJPY.mean     USDJPY.sd   USDJPY.skew   USDJPY.kurt 
## -2.622212e-02 -9.547692e-05  4.840386e-03 -1.173991e-01  1.181558e+01 
##    USDJPY.max 
##  2.133925e-02
```

4. Test hypothesis H0: u=0 against alternative H0: u!=0

```r
suppressWarnings(t.test(GBPUSD.daily.log.return))
```

```
##
##  One Sample t-test
##
## data:  GBPUSD.daily.log.return
## t = -0.59428, df = 178, p-value = 0.5531
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0010974507  0.0005894424
## sample estimates:
##     mean of x
## -0.0002540042
```

```r
suppressWarnings(t.test(USDJPY.daily.log.return))
```

```
##
##  One Sample t-test
##
## data:  USDJPY.daily.log.return
## t = -0.2639, df = 178, p-value = 0.7922
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0008094213  0.0006184675
## sample estimates:
##     mean of x
## -9.547692e-05
```

If we set our alpha to be 0.05, in both cases the p- values are not significant. So for both cases we cannot reject the null hypothesis that mean of the daily log returns is zero.