Assignment1

Manish Kumar

2023-01-11

1. Read Chapter 1 of the book

2. Download and analyze adjusted excess returns of S&P 500. In case if package `quantmod` is not working on your system, download the data directly from the web site https://finance.yahoo.com

2.1 Calculate continuous daily excess returns of SP500 ("^GSPC") for the period from 1/1/2022 until 12/22/2022 using overnight Fed Funds rates as risk-free rates

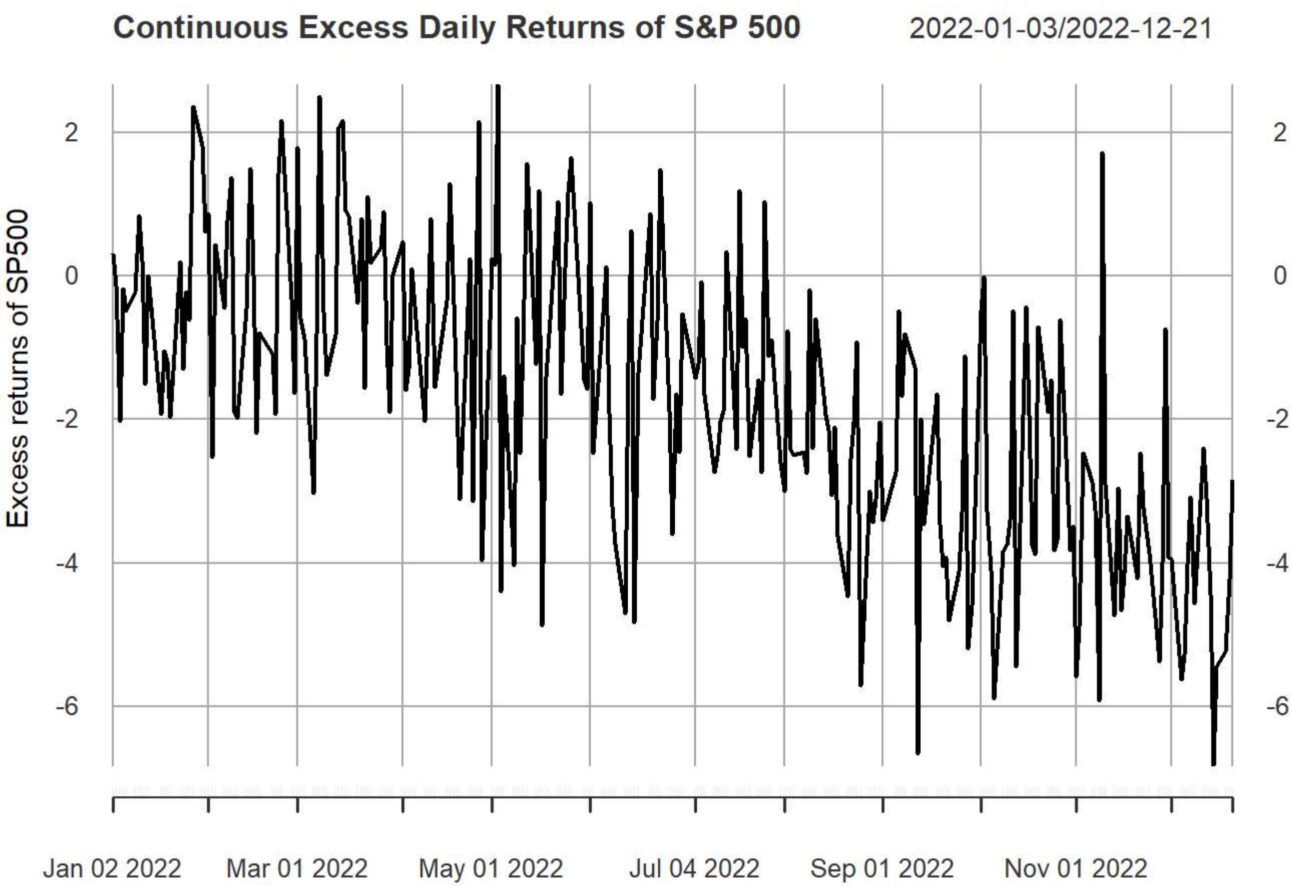
```
# Load the required library
suppressMessages(library(quantmod))

# Get the S&P500 data and Fed Funds Rates
getSymbols("^GSPC", src="yahoo", from=as.Date("2022-01-01"), to=as.Date("2022-12-22"), periodicity
="daily")
getSymbols("DFF", src="FRED", from=as.Date("2022-01-01"), to=as.Date("2022-12-22"), periodicity="d
aily")
```

```
## [1] "^GSPC"
## [1] "DFF"
```

```
# Calculate daily returns (dailyReturns automatically takes care of the Adjusted closing price)
SP500 <- dailyReturn(GSPC)

# Filter and order by the date range in S&P500 data
fed_funds_rate = DFF[rownames(data.frame(SP500))] / 100

# Calculate excess returns
daily_excess_returns = (SP500 - fed_funds_rate) * 100

# Show and plot the data
head(daily_excess_returns)
```

```
##            daily.returns
## 2022-01-03     0.3055040
## 2022-01-04    -0.1429622
## 2022-01-05    -2.0192758
## 2022-01-06    -0.1763769
## 2022-01-07    -0.4850217
## 2022-01-10    -0.2241031
```

```
plot(daily_excess_returns, ylab="Excess returns of SP500", main="Continuous Excess Daily Returns o
f S&P 500")
```



2.2 Calculate average daily excess return, actual return of S&P 500 in 2022 per day, and average FedFund rate of return per day

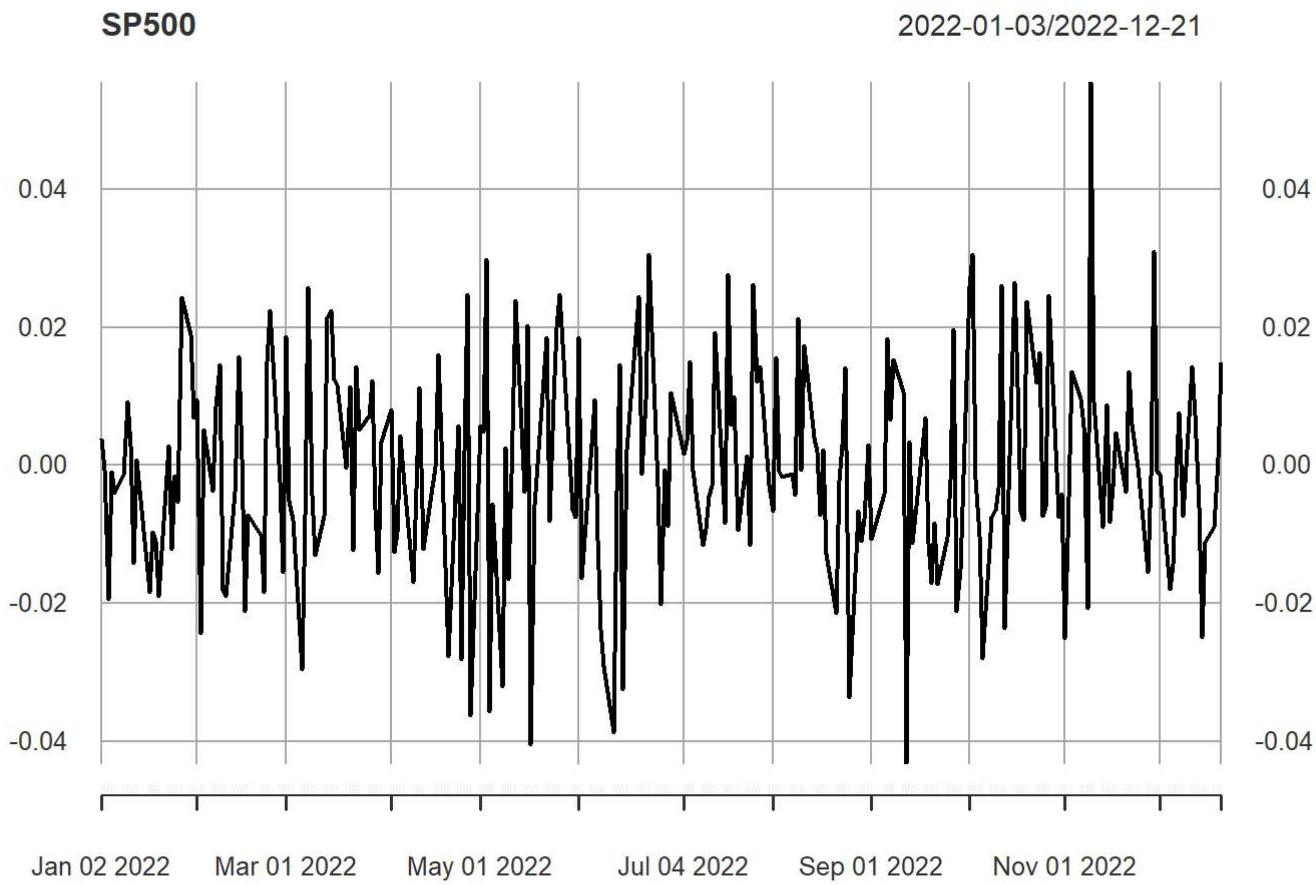
```
# Average daily excess return
cat("Average Daily Excess Return:", mean(daily_excess_returns), "\n")
```

```
## Average Daily Excess Return: -1.704405
```

```
# Actual return for S&P500
head(SP500)
```

```
##            daily.returns
## 2022-01-03  0.0038550401
## 2022-01-04 -0.0006296221
## 2022-01-05 -0.0193927578
## 2022-01-06 -0.0009637689
## 2022-01-07 -0.0040502168
## 2022-01-10 -0.0014410312
```

```
plot(SP500)
```




```
# Average FedFund rate of return
cat("Average FedFund rate of Return:", mean(fed_funds_rate), "\n")
```

```
## Average FedFund rate of Return: 0.0163098
```

3. Download and analyze exchange rates. In case if package `quantmod` is not working on your system, download the data directly from the web site https://finance.yahoo.com.

Answer the following 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.

3.1 Download from `Oanda` using method `getFX()` from library `quantmode` for the period from July 1, 2022 to December 22, 2022:

- GBP/USD exchange rate, i.e. price of 1 British pound in US dollars
- USD/JPY exchange rate, i.e. price of 1 US dollar in Japanese yen

```r
# Download exchange rates
suppressWarnings(getFX("GBP/USD", src="oanda", from=as.Date("2022-07-01"), to=as.Date("2022-12-2
2")))
suppressWarnings(getFX("USD/JPY", src="oanda", from=as.Date("2022-07-01"), to=as.Date("2022-12-2
2")))
```

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

3.2 Calculate daily log returns of both exchange rates

```r
daily_log_ret_gbp <- diff(log(GBPUSD))
daily_log_ret_jpy <- diff(log(USDJPY))

head(daily_log_ret_gbp)
```

```
##                  GBP.USD
## 2022-07-16            NA
## 2022-07-17  3.285817e-05
## 2022-07-18  6.358335e-03
## 2022-07-19  3.875245e-03
## 2022-07-20  2.801597e-04
## 2022-07-21 -2.318682e-03
```

```r
head(daily_log_ret_jpy)
```

```
##                  USD.JPY
## 2022-07-16            NA
## 2022-07-17 -0.0000425051
## 2022-07-18 -0.0024752093
## 2022-07-19 -0.0015511176
## 2022-07-20  0.0015068046
## 2022-07-21 -0.0003146622
```

3.3 Calculate sample min, mean, sd, skewness, kurtosis, max of log returns for both exchange rates

```r
suppressMessages(library(fBasics))

cat("Stats for GBP", "\n")

# Min
cat("Min: ", min(daily_log_ret_gbp, na.rm=TRUE), "\n")

# Mean
cat("Mean: ", mean(daily_log_ret_gbp, na.rm=TRUE), "\n")

# Standard Deviation
cat("Sd: ", sd(daily_log_ret_gbp, na.rm=TRUE), "\n")

# Skewness
cat("Skewness: ", skewness(daily_log_ret_gbp, na.rm=TRUE), "\n")

# Kurtosis
cat("Kurtosis: ", kurtosis(daily_log_ret_gbp, na.rm=TRUE), "\n")

# Max
cat("Max: ", max(daily_log_ret_gbp, na.rm=TRUE), "\n")
```

```
## Stats for GBP
## Min:  -0.01892142
## Mean:  0.0001054084
## Sd:  0.006120471
## Skewness:  0.2354114
## Kurtosis:  2.170017
## Max:  0.02060692
```

```r
cat("Stats for JPY", "\n")

# Min
cat("Min: ", min(daily_log_ret_jpy, na.rm=TRUE), "\n")

# Mean
cat("Mean: ", mean(daily_log_ret_jpy, na.rm=TRUE), "\n")

# Standard Deviation
cat("Sd: ", sd(daily_log_ret_jpy, na.rm=TRUE), "\n")

# Skewness
cat("Skewness: ", skewness(daily_log_ret_jpy, na.rm=TRUE), "\n")

# Kurtosis
cat("Kurtosis: ", kurtosis(daily_log_ret_jpy, na.rm=TRUE), "\n")

# Max
cat("Max: ", max(daily_log_ret_jpy, na.rm=TRUE), "\n")
```

```
## Stats for JPY
## Min:  -0.02962302
## Mean:  -0.0002968692
## Sd:  0.006041583
## Skewness:  -1.387603
## Kurtosis:  5.096345
## Max:  0.01500384
```

3.4 Test hypothesis H_0: μ=0 against alternative H_0: μ≠0

```
suppressWarnings(t.test(daily_log_ret_gbp))
```

```
##
##  One Sample t-test
##
## data:  daily_log_ret_gbp
## t = 0.21716, df = 158, p-value = 0.8284
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0008532715  0.0010640883
## sample estimates:
##    mean of x
## 0.0001054084
```

```
suppressWarnings(t.test(daily_log_ret_jpy))
```

```
##
##  One Sample t-test
##
## data:  daily_log_ret_jpy
## t = -0.6196, df = 158, p-value = 0.5364
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0012431925  0.0006494542
## sample estimates:
##     mean of x
## -0.0002968692
```

In both cases, Since this p-value is greater than 0.05, we fail to reject the null hypothesis.