FE515 2022A Midterm

Yufu Liao

03/22/2023

Question 1: (50 points)

1.1

Download daily equity data of JPM and WFC (2012-01-01 to 2023-01-01)

```
#install.packages('quantmod')
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
```

```
getSymbols(Symbols = "JPM", from = "2012-01-01", to = '2023-01-01')
```

```
## Warning in strptime(xx, ff, tz = "GMT"): unable to identify current timezone 'T':
## please set environment variable 'TZ'
```

```
## [1] "JPM"
```

```
JPM <- data.frame(JPM)
head(JPM)
```

	JPM.Open <dbl>	JPM.High <dbl>	JPM.Low <dbl>	JPM.Close <dbl>	JPM.Volume <dbl>	JPM.Adjusted <dbl>
2012-01-03	34.06	35.19	34.01	34.98	44102800	25.54859
2012-01-04	34.44	35.15	34.33	34.95	36571200	25.71043
2012-01-05	34.71	35.92	34.40	35.68	38381400	26.24745
2012-01-06	35.69	35.77	35.14	35.36	33160600	26.01204
2012-01-09	35.44	35.68	34.99	35.30	23001800	25.96791
2012-01-10	36.07	36.35	35.76	36.05	35972800	26.51963

6 rows

```
getSymbols(Symbols = "WFC", from = "2012-01-01", to = '2023-01-01')
```

```
## [1] "WFC"
```

```
WFC <- data.frame(WFC)
head(WFC)
```

	WFC.Open <dbl>	WFC.High <dbl>	WFC.Low <dbl>	WFC.Close <dbl>	WFC.Volume <dbl>	WFC.Adjusted <dbl>
2012-01-03	27.94	28.52	27.94	28.43	40071200	20.65029
2012-01-04	28.34	28.69	28.04	28.56	27519200	20.74472
2012-01-05	28.50	29.58	28.25	29.02	48435100	21.07884
2012-01-06	28.84	29.08	28.46	28.94	32303500	21.02073
2012-01-09	29.15	29.38	29.00	29.30	25720100	21.28221
2012-01-10	29.74	29.80	29.18	29.41	29860100	21.36211

6 rows

1.2

Calculate both the daily log return and weekly log return for each stock.

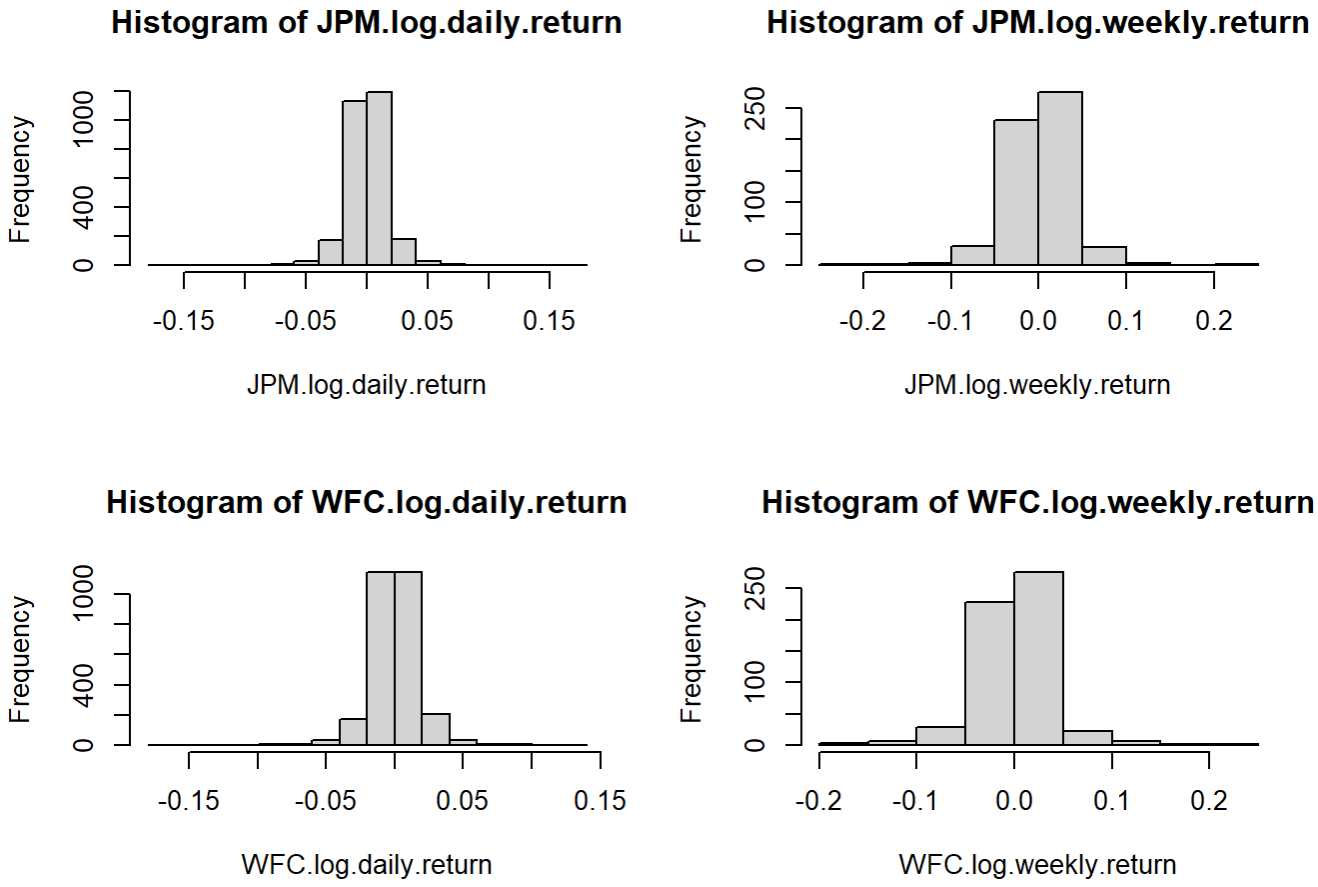
```
JPM.log.daily.return <- diff(log(JPM$JPM.Adjusted))
JPM.log.weekly.return <- periodReturn(JPM, type = 'log', period = 'weekly')

WFC.log.daily.return <- diff(log(WFC$WFC.Adjusted))
WFC.log.weekly.return <- periodReturn(WFC, type = 'log', period = 'weekly')
```

1.3

Visualize the distribution of these log returns using hist() function. Use par() function to put the four histogram together into one single graph, where each histogram is an individual subplot.

```
par(mfrow = c(2, 2))
hist(JPM.log.daily.return)
hist(JPM.log.weekly.return)
hist(WFC.log.daily.return)
hist(WFC.log.weekly.return)
```



1.4

Calculate the first four moments, i.e. mean, variance, skewness and kurtosis, for each stock. Store the calculate result in a data frame and report the result in a table.

```
library('moments')

jpm_moments <- c(mean(JPM.log.daily.return), var(JPM.log.daily.return), moments::skewness(JPM.lo
g.daily.return), moments::kurtosis(JPM.log.daily.return))

# Calculate the first four moments for WFC daily log returns
wfc_moments <- c(mean(WFC.log.daily.return), var(WFC.log.daily.return), moments::skewness(WFC.lo
g.daily.return), moments::kurtosis(WFC.log.daily.return))

# Combine the results into a data frame
moments_df <- data.frame(Statistic = c("Mean", "Variance", "Skewness", "Kurtosis"), JPM = jpm_mo
ments, WFC = wfc_moments)

# Print the data frame
print(moments_df)
```

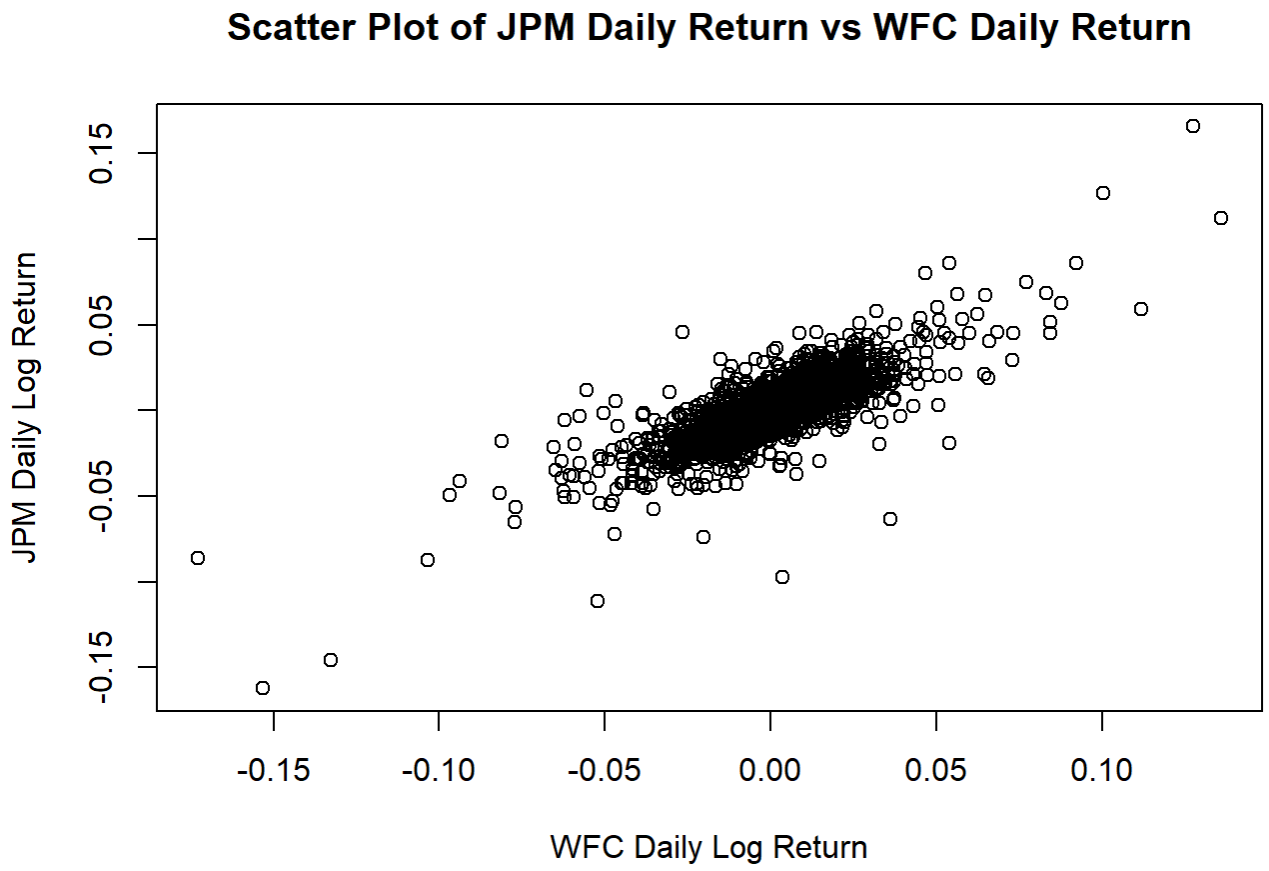
```
##   Statistic           JPM           WFC
## 1      Mean  0.0005965465  0.0002481144
## 2  Variance  0.0002875020  0.0003325385
## 3  Skewness -0.1100144471 -0.3475342967
## 4  Kurtosis 15.7443253866 14.1951100693
```

1.5

Draw a scatter plot of JPM daily return against WFC daily return. (i.e. WFC return on x-axis and JPM return on y-axis)

```
returns_df <- data.frame(JPM = JPM.log.daily.return, WFC = WFC.log.daily.return)

plot(JPM ~ WFC, data = returns_df, xlab = "WFC Daily Log Return", ylab = "JPM Daily Log Return",
     main = "Scatter Plot of JPM Daily Return vs WFC Daily Return")
```



1.6

Build a simple linear regression model using the WFC daily return as explanatory variable and the JPM daily return as response variable. Report the fitted model using summary( ) function.

```
model <- lm(JPM ~ WFC, data = returns_df)

summary(model)
```

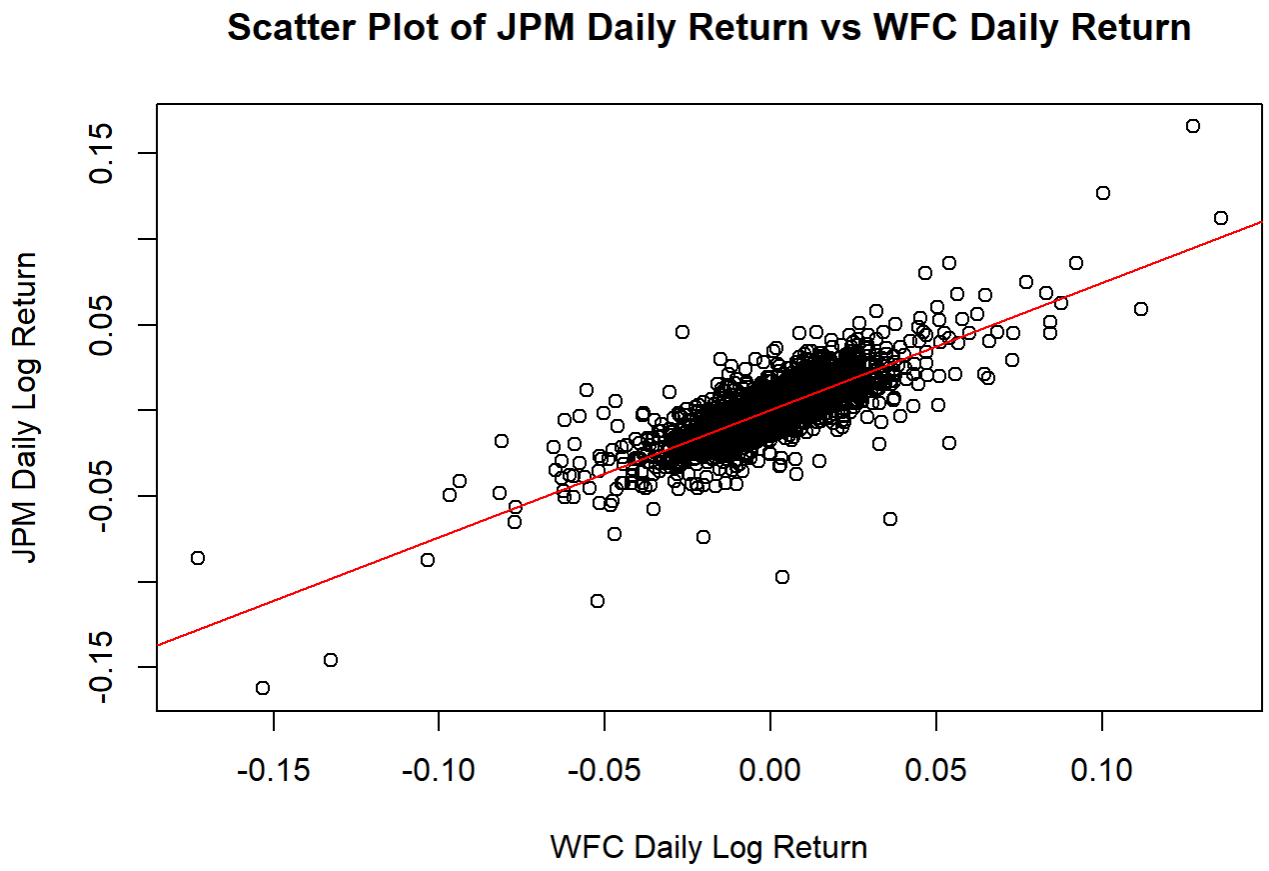
```
##
## Call:
## lm(formula = JPM ~ WFC, data = returns_df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.100468 -0.005141 -0.000134  0.005129  0.070680
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0004122  0.0001938   2.127   0.0335 *
## WFC         0.7430601  0.0106299  69.903   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01019 on 2765 degrees of freedom
## Multiple R-squared:  0.6386, Adjusted R-squared:  0.6385
## F-statistic:  4886 on 1 and 2765 DF,  p-value: < 2.2e-16
```

1.7

Draw a regression line on the scatter plot using the fitted model above. Make sure use a different color to draw the regression line.

```
plot(JPM ~ WFC, data = returns_df, xlab = "WFC Daily Log Return", ylab = "JPM Daily Log Return",
     main = "Scatter Plot of JPM Daily Return vs WFC Daily Return")

abline(model, col = "red")
```



Question 2

2.1

Without using packages, create a function of 2 variables “x” and “adjusted” that calculates the sample skewness of “x” using the formulas on Lecture 6 page 20 and page 21. When ”adjusted” = TRUE, it returns the adjusted skewness of “x”, and FALSE returns the unadjusted one.

```
fun <- function(x, adjusted) {

  if(length(x) <= 2)
    stop("please give more than 3 elements to calculate skewness")

  m3 <- sum((x - mean(x))^3) / length(x)
  m2 <- sum((x - mean(x))^2) / length(x)
  result <- m3 / (m2^(3/2))
  if(adjusted) {
    result <- result * sqrt((length(x) * (length(x) - 1))) / (length(x) - 2)
  }
  return(result)
}
```

2.2

Without using packages, create a function of 2 variables “x” and ”adjusted” that calculates the sample kurtosis of “x” using the formulas on Lecture 6 page 20 and page 23. When “adjusted” = TRUE, it returns the adjusted kurtosis of “x”, and FALSE returns the unadjusted one.

```
fun2 <- function(x, adjusted) {
  n <- length(x)
  if(n <= 3) {
    stop("please give more than 4 elements to calculate kurtosis")
  }
  m4 <- sum((x - mean(x))^4) / n
  m2 <- sum((x - mean(x))^2) / n
  kurtosis <- (m4 / m2^2)
  if(adjusted) {
    kurtosis <- ((n-1)  * ( (n+1) * kurtosis - 3 * (n-1) ) / ((n-2) * (n-3))) + 3
  }
  return(kurtosis)
}

#fun2(c(1,4,9, 11), TRUE)
#fun2(c(1,4,9, 11), FALSE)
```

2.3

Download historical price for ticker ”SPY” for the whole 2012 and 2013 years with quantmod package, use its adjusted close price to calculate daily log return (Note the adjusted close price is different from the “adjusted” for sample moments).

```
getSymbols(Symbols = "SPY", from = "2012-01-01", to = '2014-01-01')
```

```
## [1] "SPY"
```

```
SPY <- data.frame(SPY)
head(SPY)
```

	SPY.Open <dbl>	SPY.High <dbl>	SPY.Low <dbl>	SPY.Close <dbl>	SPY.Volume <dbl>	SPY.Adjusted <dbl>
2012-01-03	127.76	128.38	127.43	127.50	193697900	103.2023
2012-01-04	127.20	127.81	126.71	127.70	127186500	103.3642
2012-01-05	127.01	128.23	126.43	128.04	173895000	103.6394
2012-01-06	128.20	128.22	127.29	127.71	148050000	103.3723
2012-01-09	128.00	128.18	127.41	128.02	99530200	103.6232
2012-01-10	129.39	129.65	128.95	129.13	115282000	104.5217

6 rows

```
SPY.log.daily.return <- diff(log(SPY$SPY.Adjusted))
```

2.4

Calculate the adjusted and unadjusted skewness for the daily log return in 2.3 using the function you defined. (both numbers should be close to -0.15)

```
adj_skewness <- fun(SPY.log.daily.return, TRUE)
adj_skewness
```

```
## [1] -0.1602064
```

```
unadj_skewness <-fun(SPY.log.daily.return, FALSE)
unadj_skewness
```

```
## [1] -0.1597263
```

2.5

Calculate the adjusted and unadjusted kurtosis for the daily log return in 2.3 using the function you defined. (both numbers should be close to 4.1)

```
adj_kurtosis <- fun2(SPY.log.daily.return, TRUE)
adj_kurtosis
```

```
## [1] 4.135157
```

```
unadj_kurtosis <-fun2(SPY.log.daily.return, FALSE)
unadj_kurtosis
```

```
## [1] 4.111907
```