Time Series 413, Assignment 8

Multivariate Volatility Models (TS8)

Reed Ballesteros

Northwestern University SPS, Fall 2022
MSDS-413-DL
Instructor: Dr. Jamie D. Riggs, Ph.D
2022-11-14

The daily prices of Microsoft (MSFT) stock and the S&P index are available for download in the TS8.R file.

- Date: year, month, and day
- MSFT.Adjusted: Microsoft adjusted closing prices
- GSPC.Adjusted: S&P adjusted closing prices

The objective is to explore the time series behavior of these data sets including EDA, modeling, model diagnostics, and interpretation.

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

1. Closing prices (60 points)

Use the daily adjusted closing prices on Microsoft and the S&P 500 over the period 2000-01-03 to 2022-10-31 to compute the continuously compounded returns.

1.1. Perform EDA on the Microsoft and S&P daily prices.

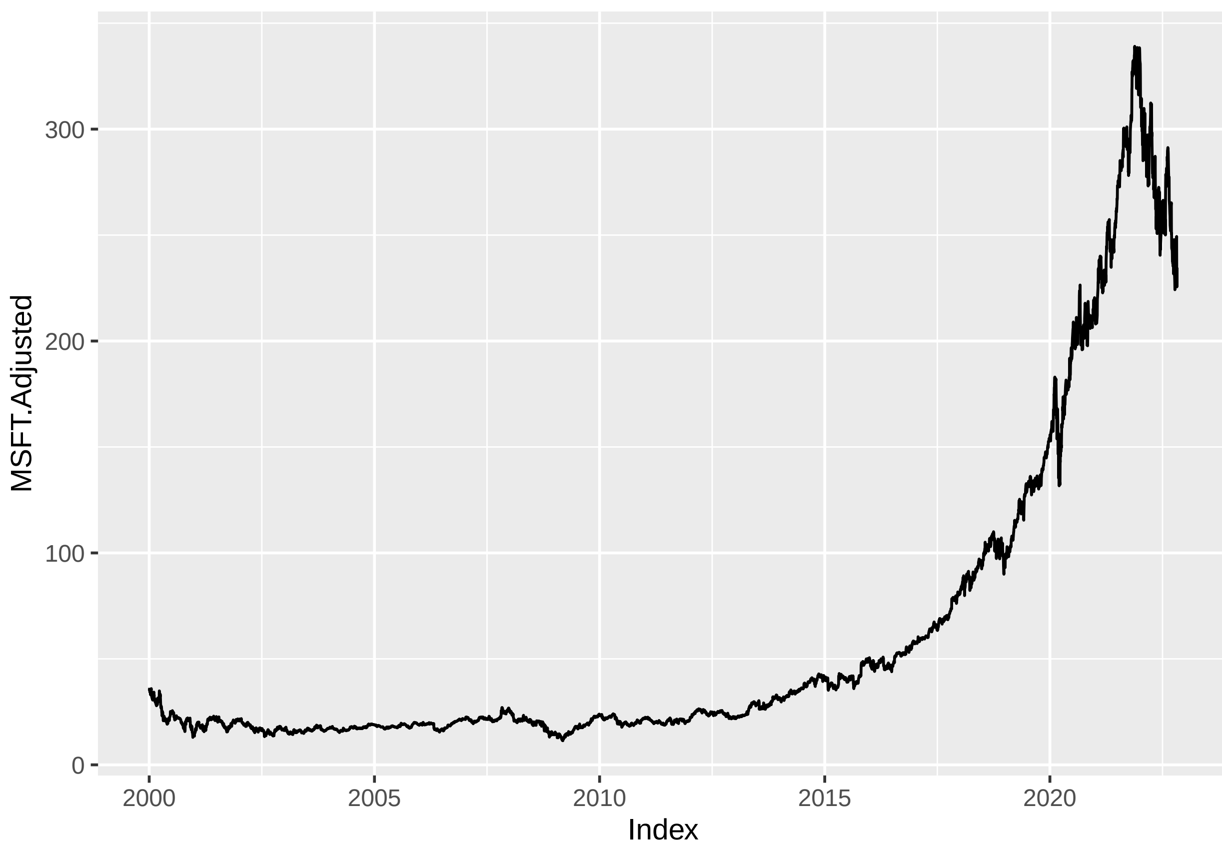
Find an additive time series without a trend.

EDA: MSFT

Plot: MSFT

Let us plot the daily adjust closing prices of Microsoft from 2000-01-03 to 2022-10-31.

Plot: MSFT

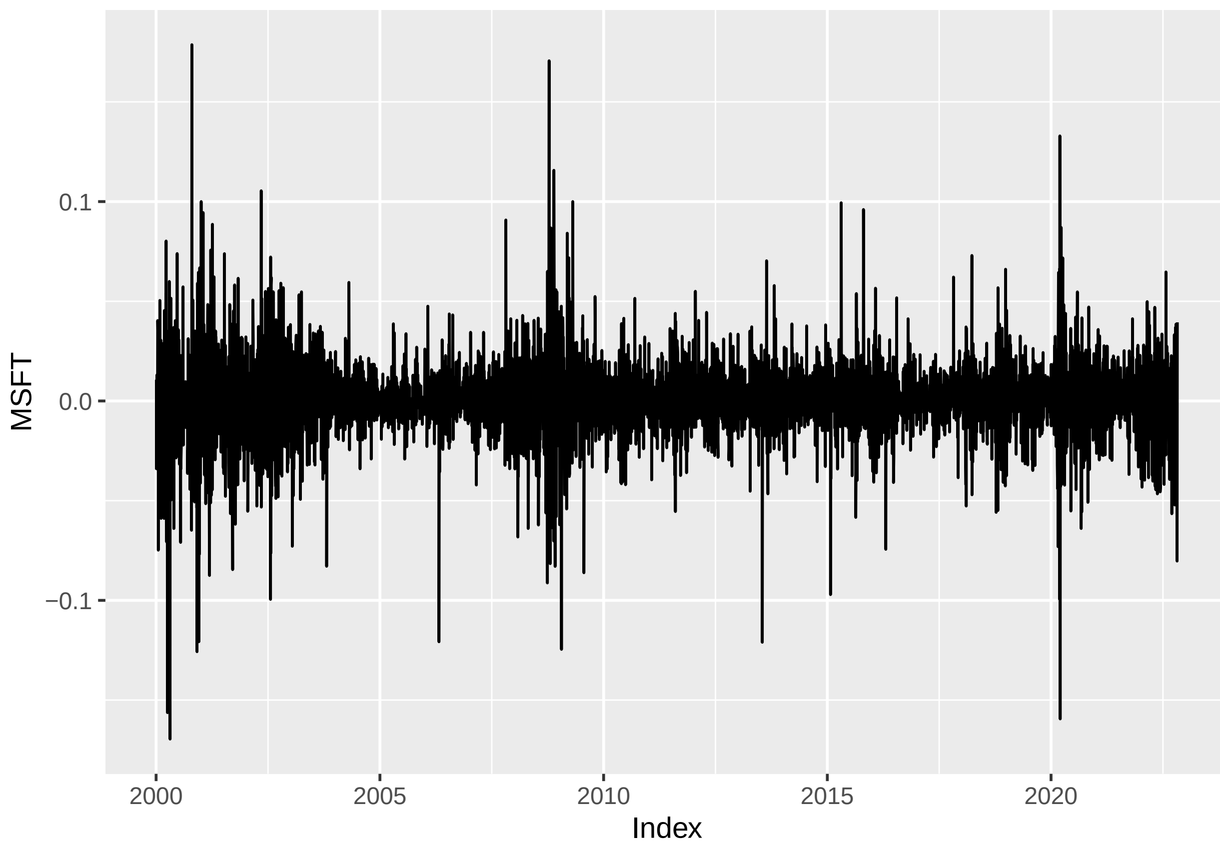


We see a generally curving upward trend up until the end of November 2021, which corresponds to the announcement of the COVID-19 Omicron variant. We and need to transform the data such that we can remove the trend.

We will calculate the compound returns from the prices of Microsoft stock using the PerformanceAnalytics::CalculateReturns(MSFT) method in R and plot it as MSFT.ret:

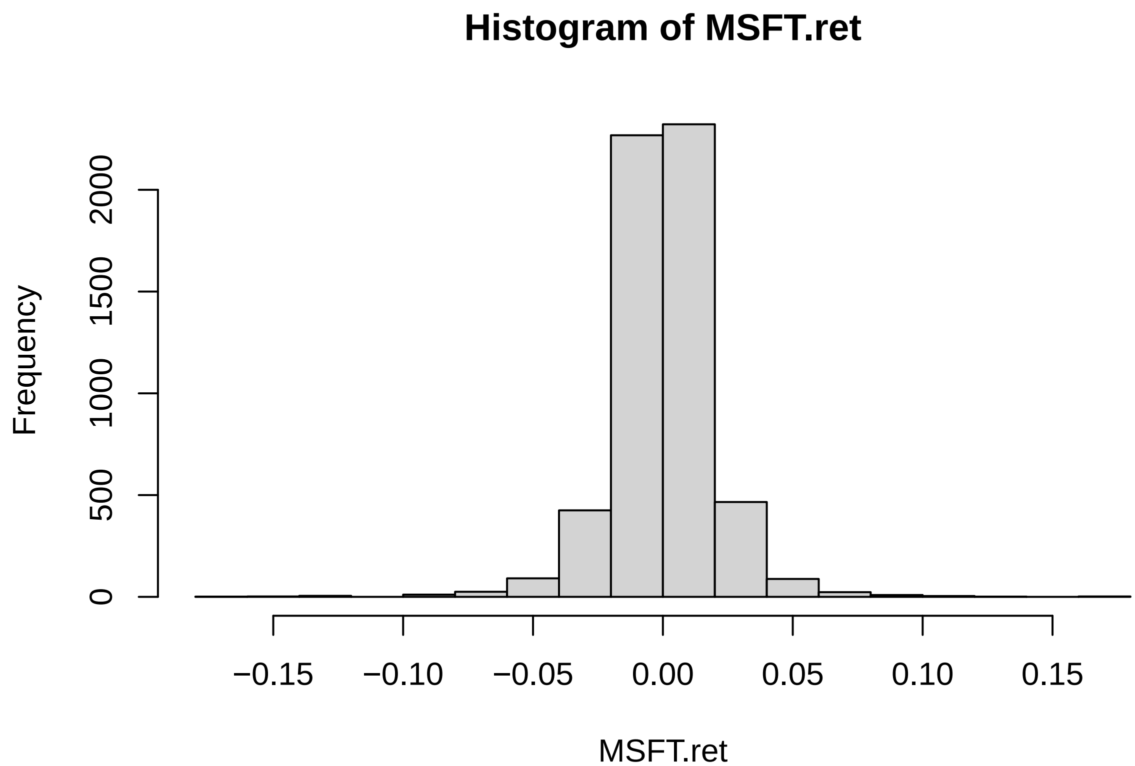
```
# calculate log-returns for GARCH analysis
MSFT.ret <- CalculateReturns(MSFT, method="log")

# remove first NA observation
MSFT.ret <- MSFT.ret[-1,]
colnames(MSFT.ret) <-"MSFT"
```

Plot: MSFT.ret



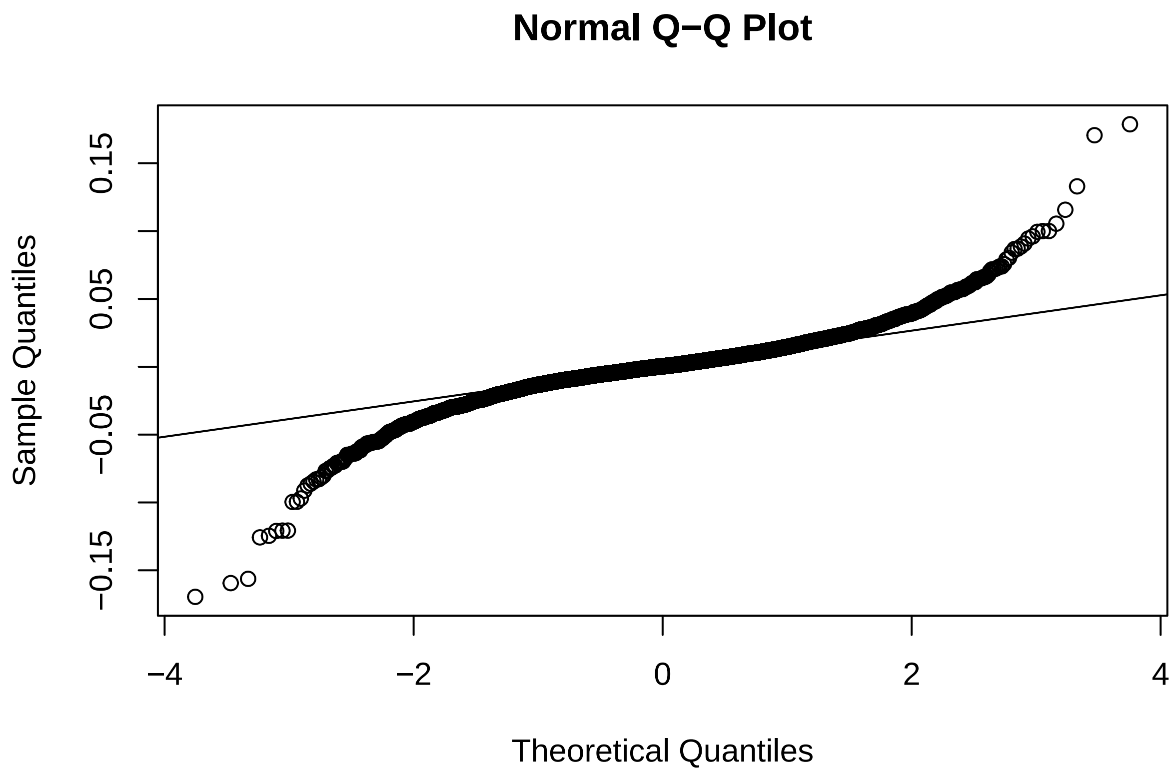
From the PerformanceAnalytics::CalculateReturns(MSFT) transformation plot above we can observe a mean 0 but we will further test the returns below to confirm it. We do not observe constant variance.

Histogram: MSFT.ret



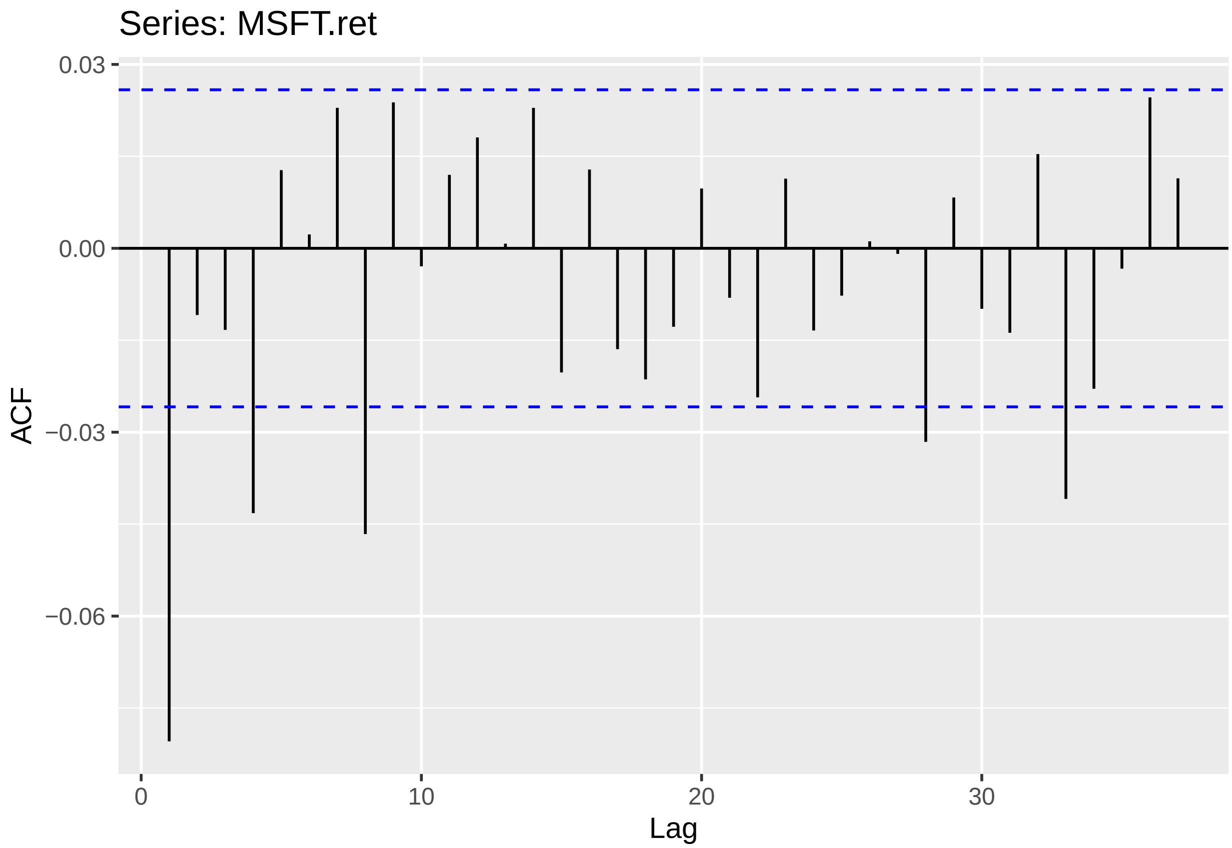
The histogram of MSFT.ret shows tall (excess) Kurtosis and slight left skewness; otherwise it the plot is almost symmetrical, but not normal based on a Gaussian PDF.

Q-Q Plot: MSFT.ret



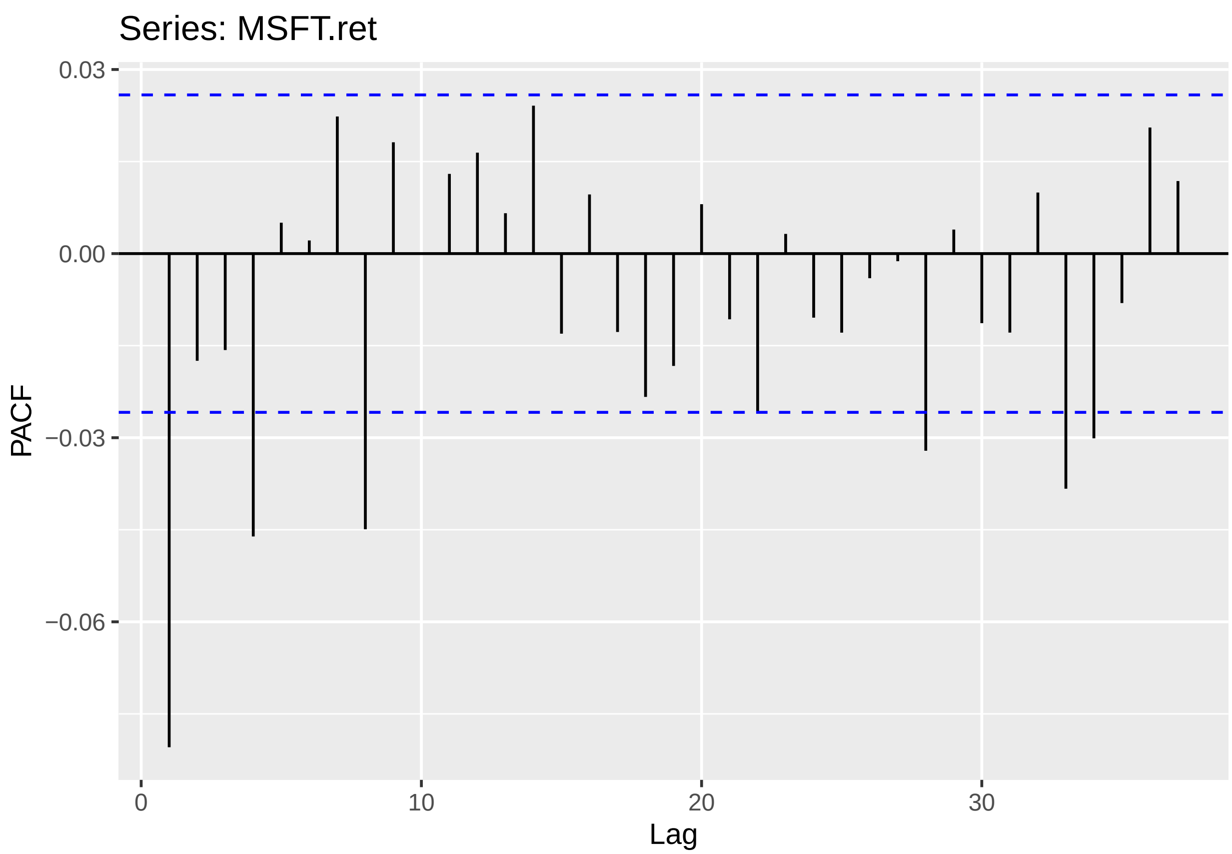
The Q-Q plot shows thick tails indicating tall Kurtosis. The extra trailing outliers on the left indicate the slight left skewness.

ACF Plot: MSFT.ret



The ACF plot shows the MSFT.ret time series data as fairly stationary, with few lags exceeding the 95% confidence interval (CI) threshold range. That said, the exceeding lag values are still very small.

PACF plot: MSFT.ret



The PACF plot is fairly stationary as well.

T-Test for Mean 0: MSFT.ret

```
##
##  One Sample t-test
##
## data:  data
## t = 1.3, df = 5742, p-value = 0.2
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0001762  0.0008255
## sample estimates:
## mean of x
## 0.0003246
##
## T-Test: mean is statistically zero, linear trend *REMOVED* ->
## *FAIL* to reject H0

## [1] TRUE
```

The 95% CI for the T-test includes 0 indicating that the mean for MSFT.ret data is statistically 0, indicating a trend is not present.

Skewness: MSFT.ret

```
##    skew  lwr.ci  upr.ci
```

```
## -0.1742 -0.1911 -0.1550
## Skew: has *LEFT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

The test confirms left skewness for the MSFT.ret data.

(excess) Kurtosis: MSFT.ret

```
##   kurt lwr.ci upr.ci
##  9.383  9.389  9.583
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

The test confirms tall (excess) Kurtosis for the MSFT.ret data.

Breusch-Pagan Test for Constant Variance: MSFT.ret

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 45, df = 1, p-value = 2e-11
##
## Breusch-Pagan: *NON*-constant variance, possible clustering,
## heteroscedastic -> reject H0
```

```
##    BP
## FALSE
```

The Breusch-Pagan test confirms non-costant variance for the MSFT.ret data.

Box-Ljung Test for Lag Dependency/Serial Autocorrelation: MSFT.ret

```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 97, df = 30, p-value = 6e-09
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

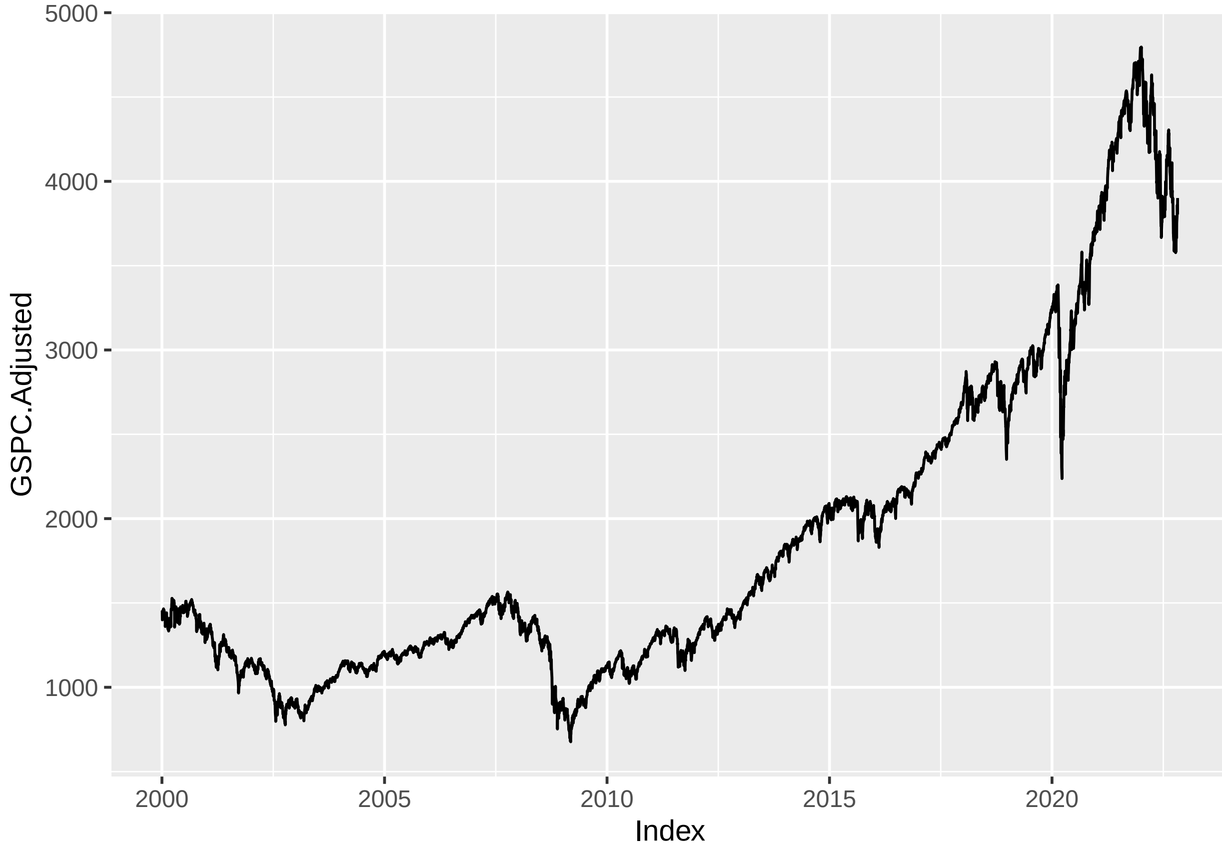
```
## [1] FALSE
```

We observe that the MSFT.ret data contains lag dependency and serial correlation.

EDA: GSPC

Let us plot the daily adjust closing prices of the S$P 500 index from 2000-01-03 to 2022-10-31.

Plot: GSPC

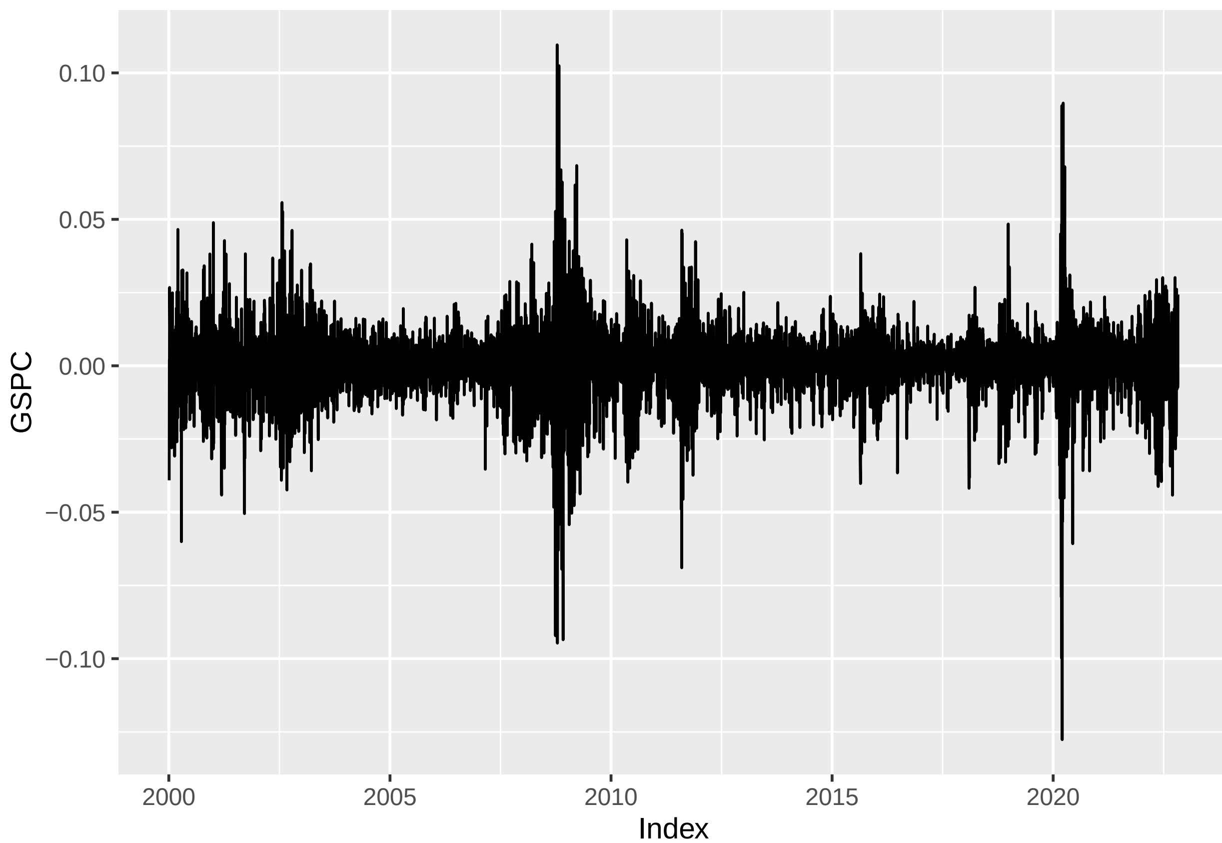


We see a generally curving upward trend of the S&P 500 index with the exception of some dips which coincide with major financial events such as the Dot-Com loss, the 2008 financial crisis, the early 2016 stock market selloff, the 2018 US Government shutdown, the start of the COVID-19 global pandemic, the COVID-19 Omicron variant, and the war in Ukraine.

We will calculate the compound returns from the prices of Microsoft stock using the PerformanceAnalytics::CalculateReturns(GSPC) method in R and plot it as GSPC.ret:

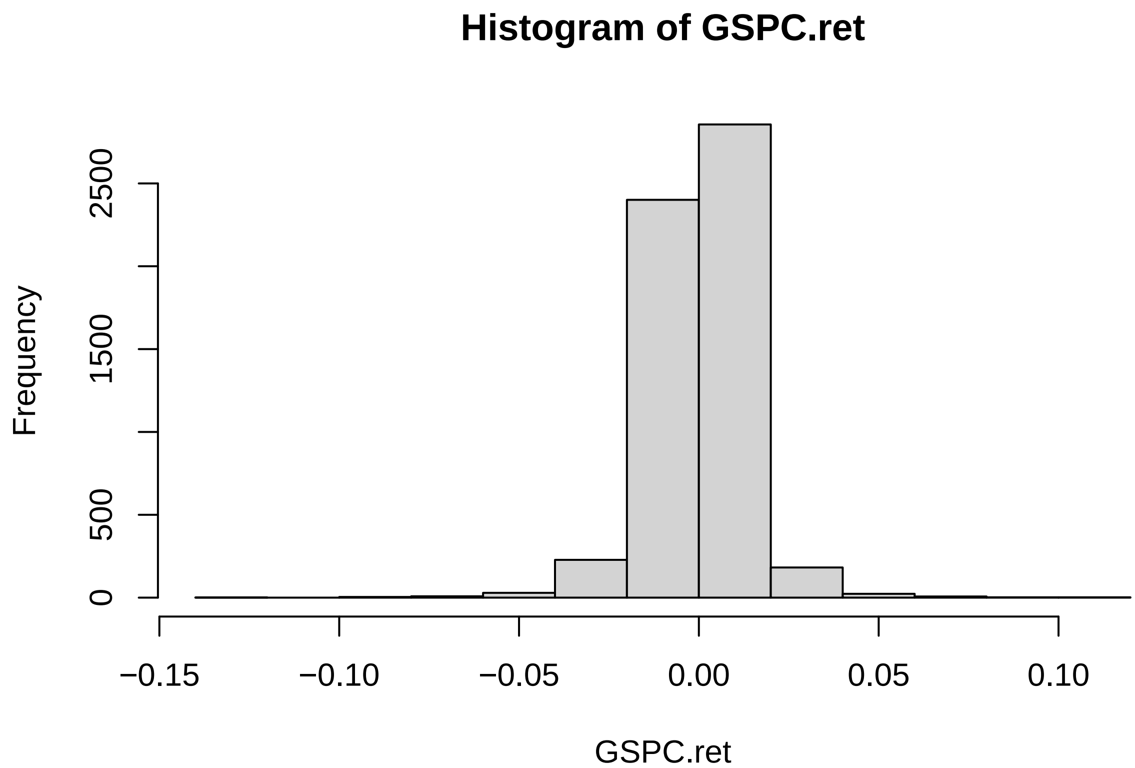
```
# calculate log-returns for GARCH analysis
GSPC.ret <- CalculateReturns(GSPC, method="log")

# remove first NA observation
GSPC.ret <- GSPC.ret[-1,]
colnames(GSPC.ret) <- "GSPC"
```

Plot: GSPC.ret



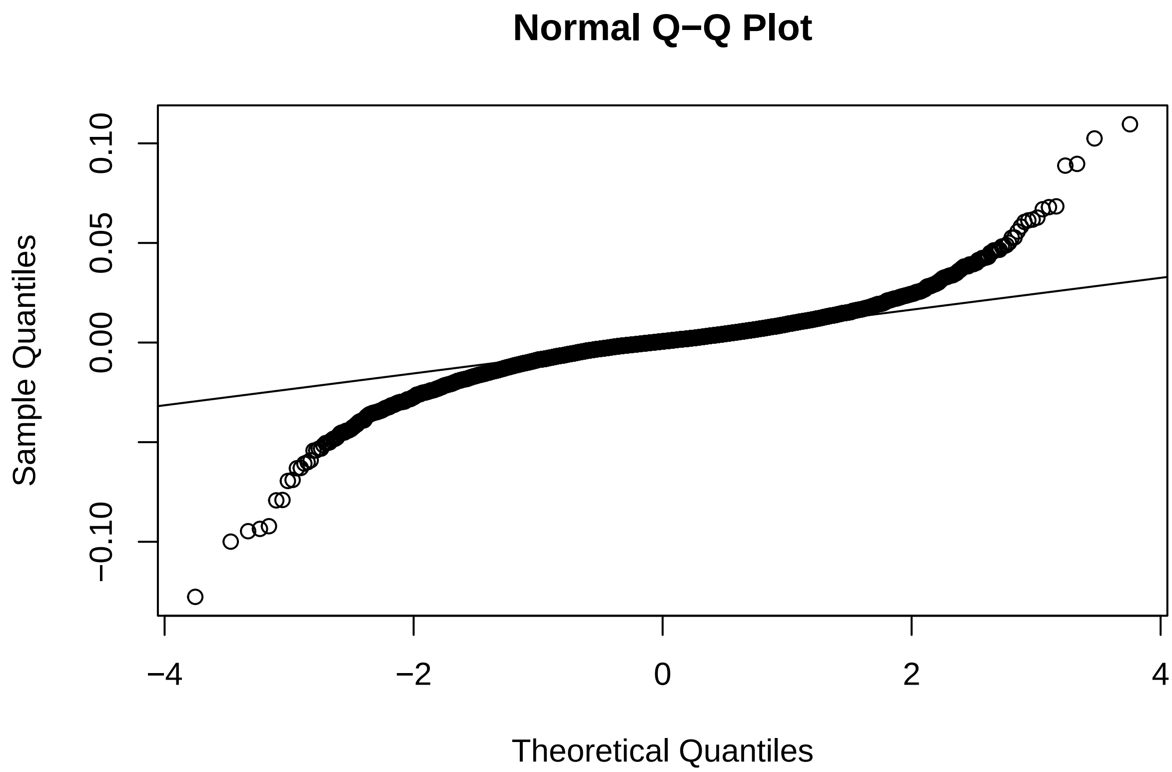
From the PerformanceAnalytics::CalculateReturns(GSPC) transformation plot above we can observe a mean 0 but we will further test the returns below to confirm it. We do not observe constant variance.

Histogram: GSPC.ret



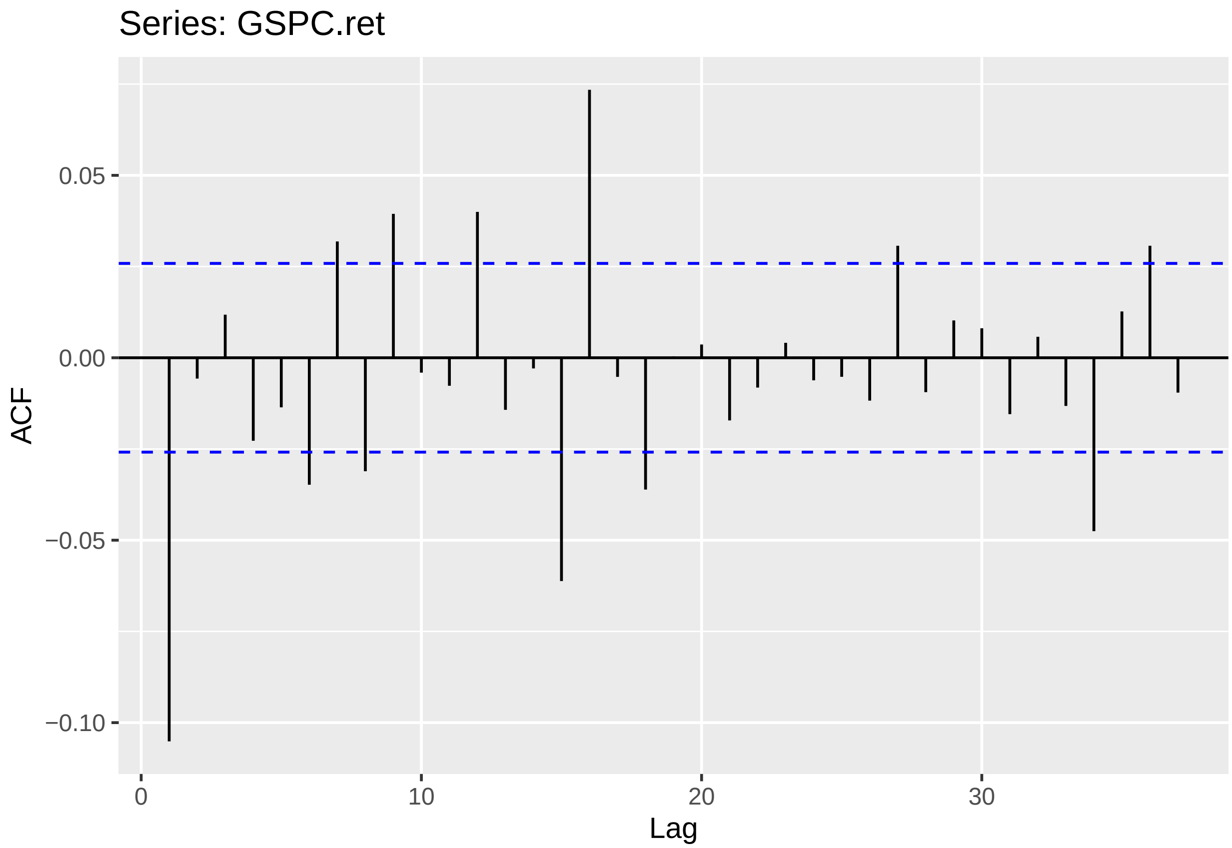
The histogram of GSPC.ret shows tall (excess) Kurtosis and slight left skewness; otherwise it the plot is almost symmetrical, but not normal based on a Gaussian PDF.

Q-Q Plot: GSPC.ret



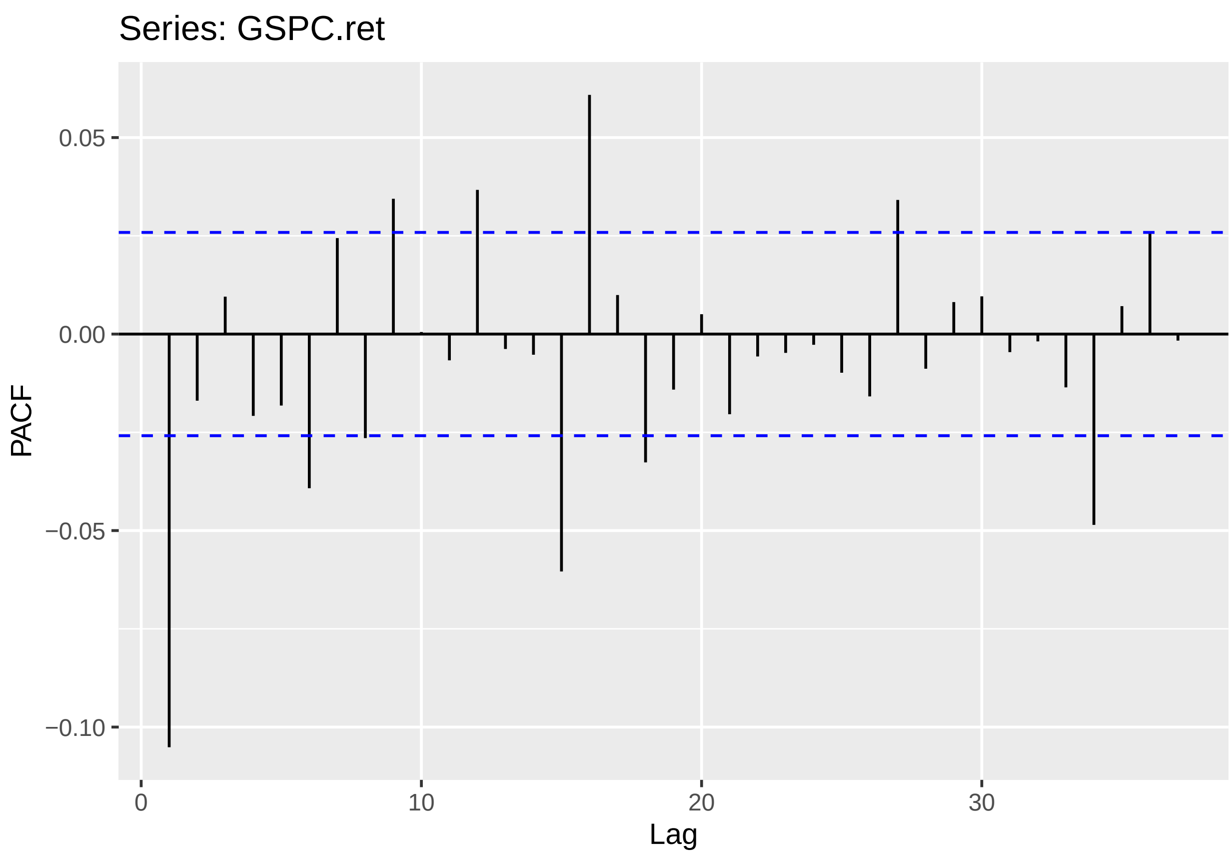
The Q-Q plot shows thick tails indicating tall Kurtosis. The extra trailing outliers on the left indicate the slight left skewness.

ACF Plot: GSPC.ret



The ACF plot shows the GSPC.ret time series data as fairly stationary, with few lags exceeding the 95% confidence interval (CI) threshold range. That said, the exceeding lag values are still very small.

PACF plot: GSPC.ret



The PACF plot is fairly stationary as well.

T-Test for Mean 0: GSPC.ret

```
##
##  One Sample t-test
##
## data:  data
## t = 1, df = 5742, p-value = 0.3
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0001520  0.0004954
## sample estimates:
## mean of x
## 0.0001717
##
## T-Test: mean is statistically zero, linear trend *REMOVED* ->
## *FAIL* to reject H0
```

```
## [1] TRUE
```

The 95% CI for the T-test includes 0 indicating that the mean for GSPC.ret data is statistically 0, indicating a trend is not present.

Skewness: GSPC.ret

```
##    skew  lwr.ci  upr.ci
```

```
## -0.3937 -0.4229 -0.3859
## Skew: has *LEFT* skewness,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

The test confirms left skewness for the GSPC.ret data.

(excess) Kurtosis: GSPC.ret

```
##   kurt lwr.ci upr.ci
##  10.26  10.32  10.56
## Kurt: has *TALL thick-tailed* (excess) kurtosis,
## property does *NOT* conform to normality and Gaussian PDF
```

```
## [1] FALSE
```

The test confirms tall (excess) Kurtosis for the GSPC.ret data.

Breusch-Pagan Test for Constant Variance: MSFT.ret

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(data ~ seq(1, length(data)))
## BP = 1.5, df = 1, p-value = 0.2
##
## Breusch-Pagan: constant variance, homoscedastic ->
## *FAIL* to reject H0
```

```
##   BP
## TRUE
```

The Breusch-Pagan test confirms non-costant variance for the GSPC.ret data.

Box-Ljung Test for Lag Dependency/Serial Autocorrelation: GSPC.ret

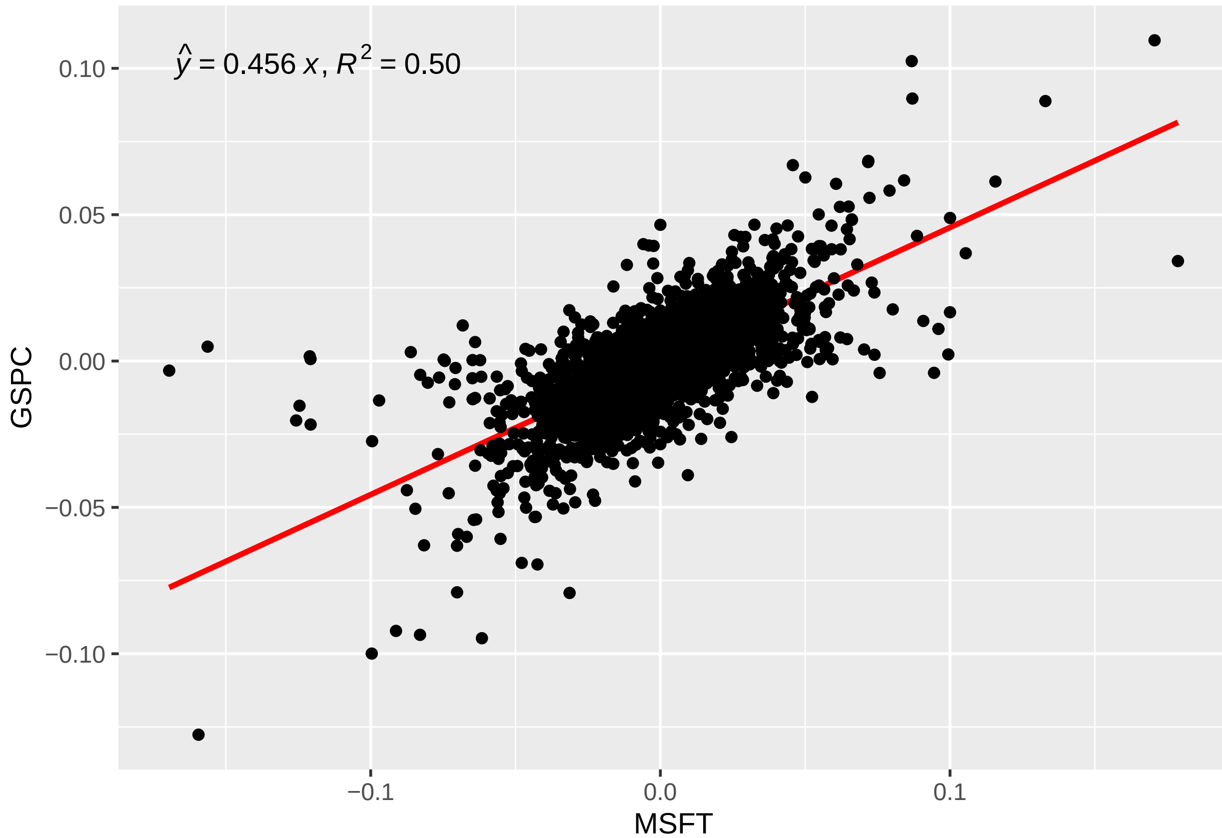
```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 177, df = 30, p-value <2e-16
##
## Box-Ljung: implies dependency present over 30 lags,
## autocorrelation present -> reject H0
```

```
## [1] FALSE
```

We observe that the GSPC.ret data contains lag dependency and serial correlation.

```
# create combined data series
MSFT.GSPC.ret <- merge(MSFT.ret,GSPC.ret)
```

Let us create a scatterplot between the transformed returns of Microsoft and the S&P 500 index:



We see somewhat of an upward linear relationship between both returns but overall mostly hovering near zero. The R-squared is 0.5, indicating that 50% of the variance of the transformed returns of the S&P 500 Index is explained by the transformed returns of Microsoft stock.

We will compute the cross-correlation matrix between the transformed returns:

```
##        MSFT   GSPC
## MSFT 1.0000 0.7061
## GSPC 0.7061 1.0000
```

```
## [1] 0.4986
```

Again we see an R-Squared value near 0.5, indicating that 50% of the variance of the transformed returns of the S&P 500 Index is explained by the transformed returns of Microsoft stock.

Let us test multivariate normality for MSFT.ret and GSPC.ret:

```
## $multivariateNormality
##              Test        Statistic               p value Result
## 1 Mardia Skewness 786.523759555481 6.37265573927001e-169     NO
## 2 Mardia Kurtosis 285.046293005668                     0     NO
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
```

```
## 1 Anderson-Darling   MSFT        98.82  <0.001      NO
## 2 Anderson-Darling   GSPC       114.39  <0.001      NO
##
## $Descriptives
##         n      Mean Std.Dev    Median     Min    Max      25th     75th    Skew
## MSFT 5743 0.0003246 0.01936 0.0003511 -0.1696 0.1787 -0.008254 0.009323 -0.1742
## GSPC 5743 0.0001717 0.01251 0.0005885 -0.1277 0.1096 -0.004888 0.005897 -0.3937
##      Kurtosis
## MSFT    9.383
## GSPC   10.263
```

There is no multivariate normality between the transformed returns of Microsoft stock and the S&P 500 index, indicated by the skewness and Kurtosis are not near 0.

1.2. First, estimate an ARCH(5) model for each series.

What is the sum of the ARCH coefficients? What does the sum tell you?

Let's perform a univariate test for ARCH effects for MSFT.ret:

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  MSFT.ret
## Chi-squared = 440, df = 5, p-value <2e-16
```

We find that there are ARCH effects for up to 5 lags.

We create the following ARCH(5) model for MSFT.ret:

```
m <- tseries::garch(MSFT.ret, order=c(0,5), trace = F)
```

```
##
## Call:
## tseries::garch(x = MSFT.ret, order = c(0, 5), trace = F)
##
## Model:
## GARCH(0,5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.036  -0.497   0.021   0.571   6.603
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0  1.11e-04    1.77e-06     62.5   <2e-16 ***
## a1  1.92e-01    1.33e-02     14.4   <2e-16 ***
## a2  1.71e-01    1.28e-02     13.3   <2e-16 ***
## a3  1.27e-01    7.67e-03     16.6   <2e-16 ***
## a4  1.48e-01    1.13e-02     13.1   <2e-16 ***
## a5  1.38e-01    1.21e-02     11.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 11597, df = 2, p-value <2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 1.7, df = 1, p-value = 0.2
```

The MSFT.ret residuals are not symmetric at the Min/Max and the Jaques-Bera test for Gaussian residuals fails. All the coefficients a0 to a5 are statistically significant. The Box-Ljung test suggests there are no ARCH effects on the squared residuals (p-value = 0.2 > 0.05).

The sum of all coefficients for the MSFT.ret ARCH(5) model:

```
sum(coef(m))
```

```
## [1] 0.7764
```

The sum of 0.7764 suggests model volatility forecasts are somewhat quick in mean-reverting but not very persistent.

Let's perform a univariate test for ARCH effects for GSPC.ret:

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  GSPC.ret
## Chi-squared = 1556, df = 5, p-value <2e-16
```

We find that there are ARCH effects for up to 5 lags.

We create the following ARCH(5) model for GSPC.ret:

```
m <- tseries::garch(GSPC.ret, order=c(0,5), trace = F)
```

```
##
## Call:
## tseries::garch(x = GSPC.ret, order = c(0, 5), trace = F)
##
## Model:
## GARCH(0,5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1316 -0.4987  0.0618  0.6223  4.2351
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
```

```
## a0  2.87e-05    1.01e-06     28.5   <2e-16 ***
## a1  1.09e-01    1.02e-02     10.7   <2e-16 ***
## a2  2.08e-01    1.57e-02     13.2   <2e-16 ***
## a3  1.86e-01    1.50e-02     12.4   <2e-16 ***
## a4  1.93e-01    1.45e-02     13.2   <2e-16 ***
## a5  1.46e-01    1.33e-02     10.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 879, df = 2, p-value <2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.13, df = 1, p-value = 0.7
```

The GSPC.ret residuals are somewhat symmetric at the Min/Max and the Jaques-Bera test for Gaussian residuals fails. The coefficients are statistically significant. The Box-Ljung test suggests there are no ARCH effects on the squared residuals (p-value = 0.7 > 0.05).

The sum of all coefficients for the GSPC.ret ARCH(5) model:

```r
sum(coef(m))
```

```
## [1] 0.8417
```

The sum of 0.8417 suggests model volatility forecasts are not too quick with mean-reverting, as well as not very persistent.

1.3. Next, estimate a GARCH(1,1) model for each series.

What is the sum of the GARCH coefficients? Interpret and compare with the ARCH sum.

We create the following GARCH(1,1) model for MSFT.ret:

```r
m <- fGarch::garchFit(MSFT.ret ~ garch(1, 1), data = MSFT.ret, trace = F)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  fGarch::garchFit(formula = MSFT.ret ~ garch(1, 1), data = MSFT.ret,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x0000025413e64dc8>
```

```
##  [data = MSFT.ret]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 7.2401e-04  7.0499e-06  8.4779e-02  8.9723e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     7.240e-04   1.919e-04    3.773 0.000161 ***
## omega  7.050e-06   1.221e-06    5.775 7.69e-09 ***
## alpha1 8.478e-02   1.035e-02    8.189 2.22e-16 ***
## beta1  8.972e-01   1.236e-02   72.567  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  15410    normalized:  2.683
##
## Description:
##  Tue Mar 28 19:58:48 2023 by user: Reed
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  17172     0
##  Shapiro-Wilk Test  R    W      NA        NA
##  Ljung-Box Test     R    Q(10)  22.18     0.0142
##  Ljung-Box Test     R    Q(15)  28.28     0.0199
##  Ljung-Box Test     R    Q(20)  31.94     0.04394
##  Ljung-Box Test     R^2  Q(10)  2.243     0.9941
##  Ljung-Box Test     R^2  Q(15)  4.896     0.993
##  Ljung-Box Test     R^2  Q(20)  6.101     0.9988
##  LM Arch Test       R    TR^2   2.723     0.9972
##
## Information Criterion Statistics:
##    AIC    BIC    SIC   HQIC
## -5.365 -5.360 -5.365 -5.363
```

The Jaques-Bera test for Gaussian residuals fails (p-value $= 0 < 0.05$). The coefficients are statistically significant. The LM test suggests there are no ARCH effects on the squared residuals (p-value $= 0.9972 > 0.05$).

The sum of all coefficients for the MSFT.ret GARCH(1,1) model:

```
sum(coef(m))
```

```
## [1] 0.9827
```

The sum of 0.9827 means the GARCH(1,1) MSFT.ret model exhibits slow mean reverting, much slower than the 0.7764 for the ARCH(0,5) model, but much higher persistence as well, almost to 1.

We create the following GARCH(1,1) model for GSPC.ret:

```
m <- fGarch::garchFit(GSPC.ret ~ garch(1, 1), data = GSPC.ret, trace = F)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  fGarch::garchFit(formula = GSPC.ret ~ garch(1, 1), data = GSPC.ret,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x00000254142bb290>
##  [data = GSPC.ret]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 5.8967e-04  2.3459e-06  1.2542e-01  8.5893e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     5.897e-04   1.055e-04    5.588 2.30e-08 ***
## omega  2.346e-06   2.963e-07    7.916 2.44e-15 ***
## alpha1 1.254e-01   9.553e-03   13.129  < 2e-16 ***
## beta1  8.589e-01   9.733e-03   88.252  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  18473    normalized:  3.217
##
## Description:
##  Tue Mar 28 19:58:48 2023 by user: Reed
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  957.5     0
##  Shapiro-Wilk Test  R    W      NA        NA
##  Ljung-Box Test     R    Q(10)  20.35     0.02615
##  Ljung-Box Test     R    Q(15)  28.66     0.01781
##  Ljung-Box Test     R    Q(20)  34.58     0.02246
##  Ljung-Box Test     R^2  Q(10)  15.86     0.1039
```

```
##  Ljung-Box Test     R^2  Q(15)  18.53     0.236
##  Ljung-Box Test     R^2  Q(20)  19.48     0.4909
##  LM Arch Test       R    TR^2   16.6      0.1653
##
## Information Criterion Statistics:
##    AIC    BIC    SIC   HQIC
## -6.432 -6.427 -6.432 -6.430
```

The Jaques-Bera test for Gaussian residuals fails (p-value = $0 < 0.05$). All Greek coefficients are statistically significant. The LM test suggests there are no ARCH effects on the squared residuals (p-value = $0.1653 > 0.05$).

The sum of all coefficients for the GSPC.ret GARCH(1,1) model:

```
## [1] 0.9849
```

The sum of 0.9827 means the GARCH(1,1) GSPC.ret model exhibits slow mean reverting, much slower than the 0.8417 for the ARCH(0,5) model, but much higher persistence as well, almost to 1.

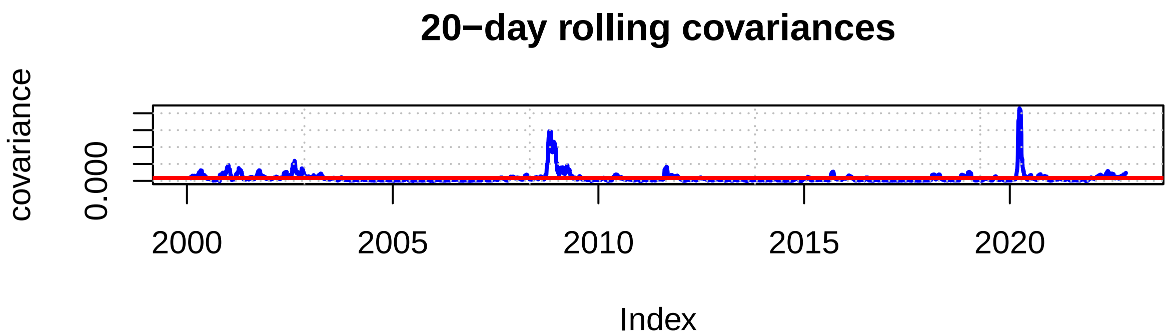
1.4. Using a 20-day moving window, compute and plot rolling covariances and correlations.

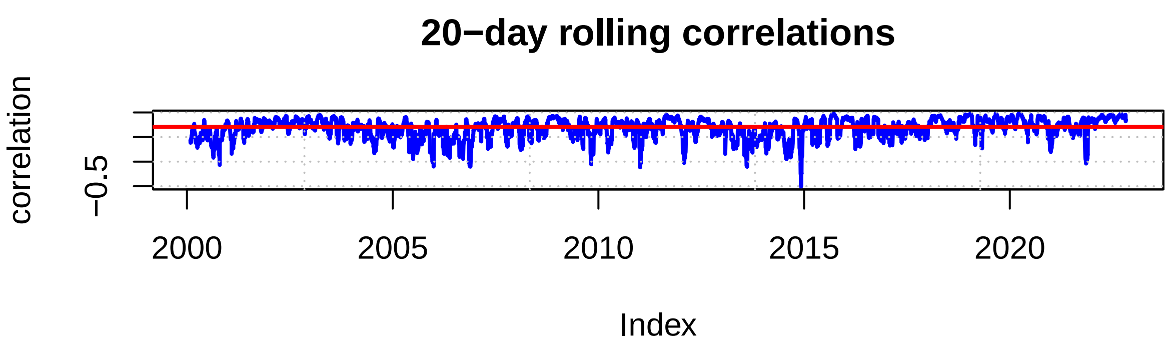
Briefly comment on what you see.

Computing the rolling covariances and correlations:

```
cor.fun <- function(x){
  cor(x)[1,2]
}

cov.fun <- function(x){
  cov(x)[1,2]
}

roll.cov <- rollapply(as.zoo(MSFT.GSPC.ret), FUN=cov.fun, width=20, by.column=FALSE, align="right")
roll.cor <- rollapply(as.zoo(MSFT.GSPC.ret), FUN=cor.fun, width=20, by.column=FALSE, align="right")
```

Plot of the rolling covariances and correlations:





The rolling covariance for a 20-day window above shows almost 0 covariance between MSFT.ret and GSPC.ret, which means most of the time MSFT.ret does not influence GSPC.RET in variance, or vice versa. That being said, we do notice some major spikes of volatility in the plot, coinciding with major financial events during the given time period, such as the .com losses in the early 2000s, the housing market crash of 2008, and the beginning of the COVID-19 pendemic in March of 2020. While the spikes go up as far as just over 0.004, indicating that MSFT.ret is leading over GSPC.ret at those times, we are not aware how much this really impacts the relationship between the two datasets as rolling covariance is scale dependent and we are not aware of the scale in context. While the spike of above 0.004 at the beginning of the COVID-19 pandemic seems like fairly small number, without the lower and upper bounds in scale we cannot fully determine how influential these leading spikes can be unless we have the proper context.

The rolling correlation for a 20-day window shows a different story. The plot shows that MSFT.ret typically leads over GSPC.ret in correlation. As correlation is standardized, we can see that the average rolling correlation between them is 0.7061, which in a 0 to 1 scale is an indication of MSFT.ret having a strong influential correlation lead.

1.5. Let $r_t = (r_M SFT, t, r_G SPC, T)^T$.

Using the **dccfit()** function from the **rmgarch** package, estimate the normal-DCC(1,1) model. Briefly comment on the estimated coefficients and the fit of the model.

Let us create a normal GARCH(1,1) mean model specification:

```
# univariate normal GARCH(1,1) for each series
garch11.spec <- ugarchspec(mean.model = list(armaOrder = c(0,0)), variance.model = list(garchOrder = c(
```

From the normal GARCH(1,1) model we will create the DCC model specification:

```r
# dcc specification - GARCH(1,1) for conditional correlations
dcc.garch11.spec <- dccspec(uspec = multispec( replicate(2, garch11.spec) ), dccOrder = c(1,1), distribu
```

We will fit the normal-DCC(1,1) model to the joint MSFT.ret and GSPC.ret data using the **rmgarch::dccfit()** function:

```r
dcc.fit <- dccfit(dcc.garch11.spec, data = MSFT.GSPC.ret)
```

```
##
## *---------------------------------*
## *          DCC GARCH Fit          *
## *---------------------------------*
##
## Distribution         :  mvnorm
## Model                :  DCC(1,1)
## No. Parameters       :  11
## [VAR GARCH DCC UncQ] : [0+8+2+1]
## No. Series           :  2
## No. Obs.             :  5743
## Log-Likelihood       :  35593
## Av.Log-Likelihood    :  6.2
##
## Optimal Parameters
## -----------------------------------
##                Estimate  Std. Error  t value Pr(>|t|)
## [MSFT].mu      0.000724    0.000192  3.76327 0.000168
## [MSFT].omega   0.000007    0.000010  0.67519 0.499556
## [MSFT].alpha1  0.084361    0.015805  5.33744 0.000000
## [MSFT].beta1   0.897780    0.040756 22.02822 0.000000
## [GSPC].mu      0.000589    0.000108  5.44221 0.000000
## [GSPC].omega   0.000002    0.000001  1.81443 0.069611
## [GSPC].alpha1  0.125050    0.014459  8.64870 0.000000
## [GSPC].beta1   0.859423    0.015532 55.33106 0.000000
## [Joint]dcca1   0.033419    0.030289  1.10334 0.269879
## [Joint]dccb1   0.948638    0.062494 15.17966 0.000000
##
## Information Criteria
## ---------------------
##
## Akaike       -12.392
## Bayes        -12.379
## Shibata      -12.392
## Hannan-Quinn -12.387
##
##
## Elapsed time : 4.79
```

The sum of coefficients for MSFT.ret:

```
## [1] 0.9821
```

0.9821 is close to 1 which makes the MSFT.ret univariate GARCH(1,1) model volatility highly persistent. The sum is close to the sum of the GARCH(1,1) model 0.9827 we created in section 1.3.

The sum of coefficients for GSPC.ret:

```
## [1] 0.9845
```

0.9845 is close to 1 which makes the GSPC.ret univariate GARCH(1,1) model volatility highly persistent. The sum is close to the sum of the GARCH(1,1) model 0.9849 we created in section 1.3.

The joint/interactive sum of coefficients:

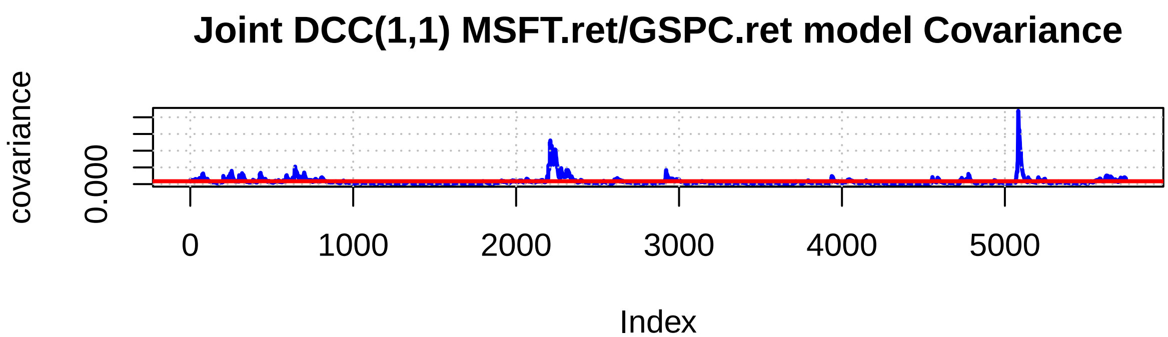
```
## [1] 0.9821
```

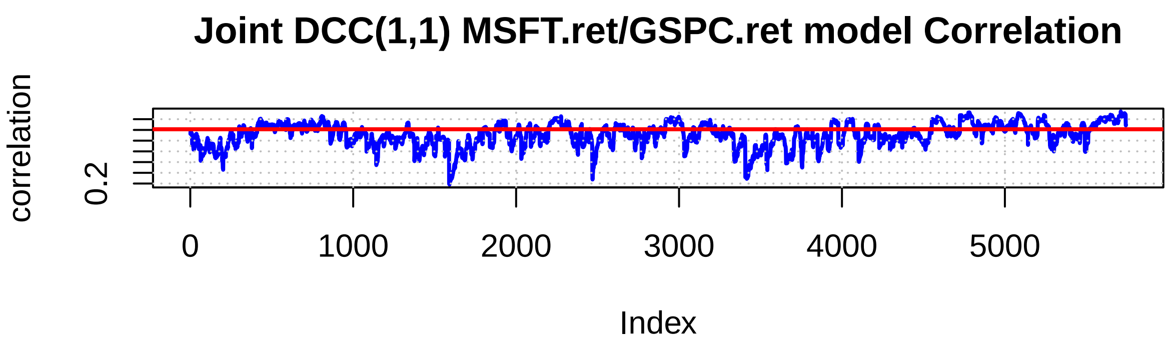
0.9821 is close to 1 which makes the joint DCC(1,1) model volatility highly persistent. Due to this high persistence, we quantitatively see that the two transformed datasets highly influence each other.

1.6. Plot the estimated in-sample conditional covariances and correlations of the DCC model.

Compare with the EWMA and rolling estimates.

Let's plot the rolling covariance and correlation of the joint DCC(1,1) model:



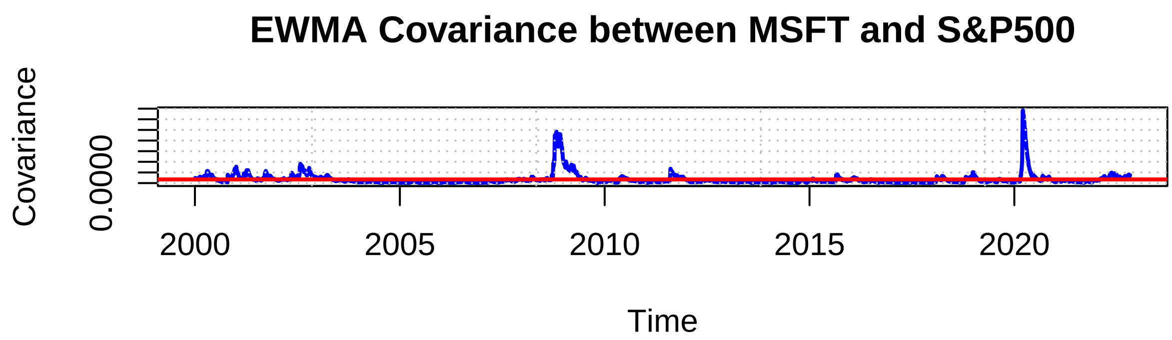


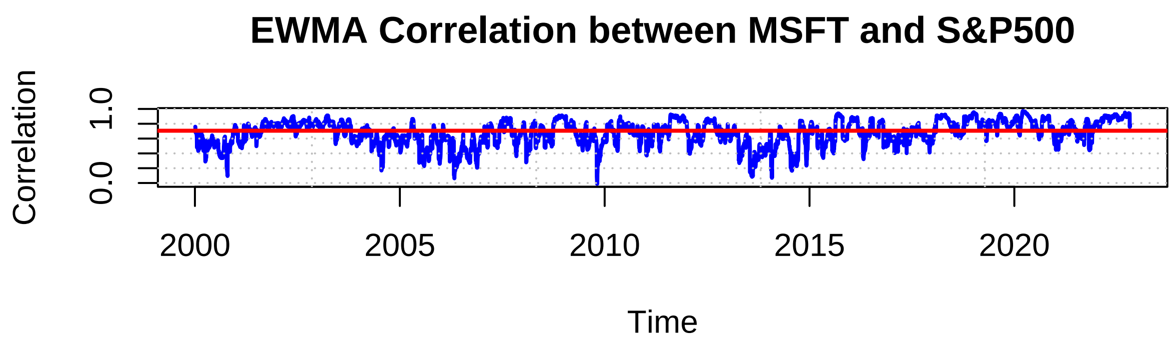
The mean of the rolling covariance in the plot above seems to be just over 0 at 0.00016. Since covariance is scale-dependent, we cannot fully determine if the spikes, while they seem small, represent strong or weak covariance without the context of the upper and lower bounds. This is similar to the 20-day rolling covariance plots we've seen in section 1.4, with the spikes coinciding with major financial events during the given time period.

The rolling correlation of of the joint DCC(1,1) model also seems to be similar to the 20-day rolling correlation plot in section 1.4, with MSFT.ret leading over GSPC.ret in correlation, with an average of 0.6531.

Let's create the EWMA covariance and correlation of the transformed joint MSFT.ret and GSPC.ret data:

The EWMA covariance and correlation plots:





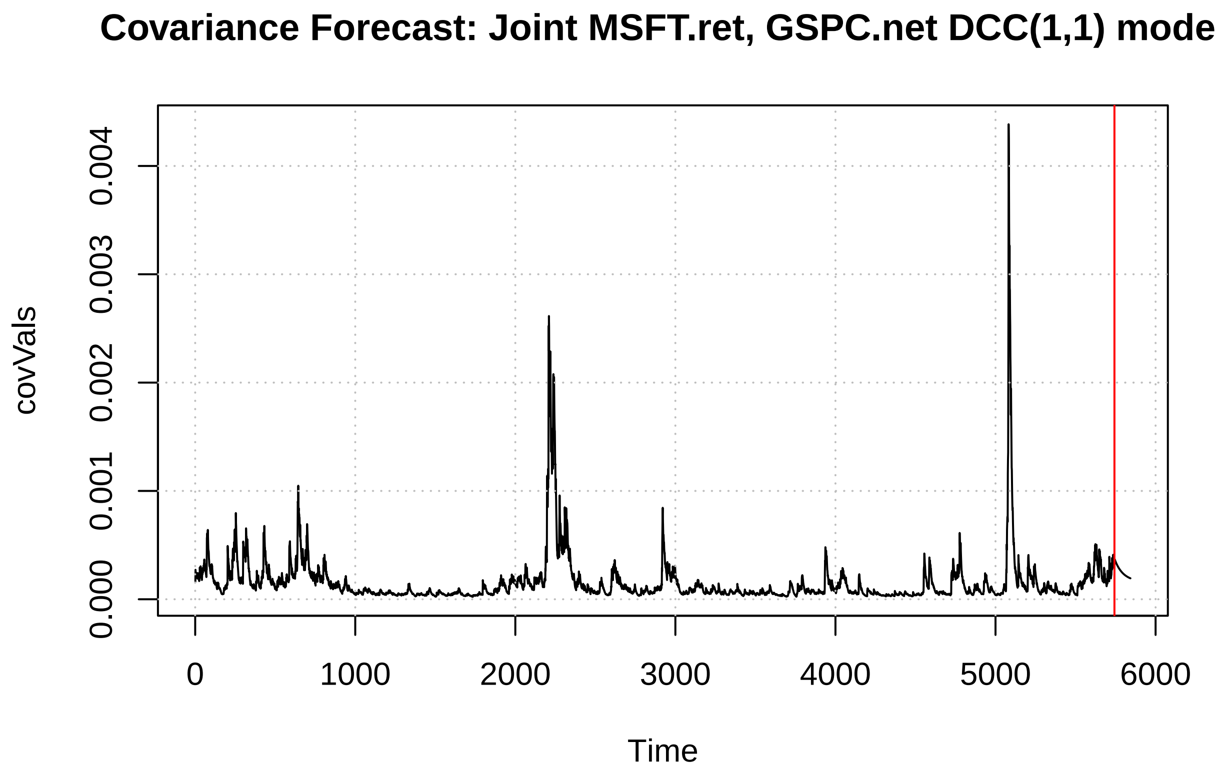
The EWMA plots are also very similar to both the 20-day and DCC(1,1) model covariance and correlation plots.

1.7. Using the Estimated DCC(1,1) Model

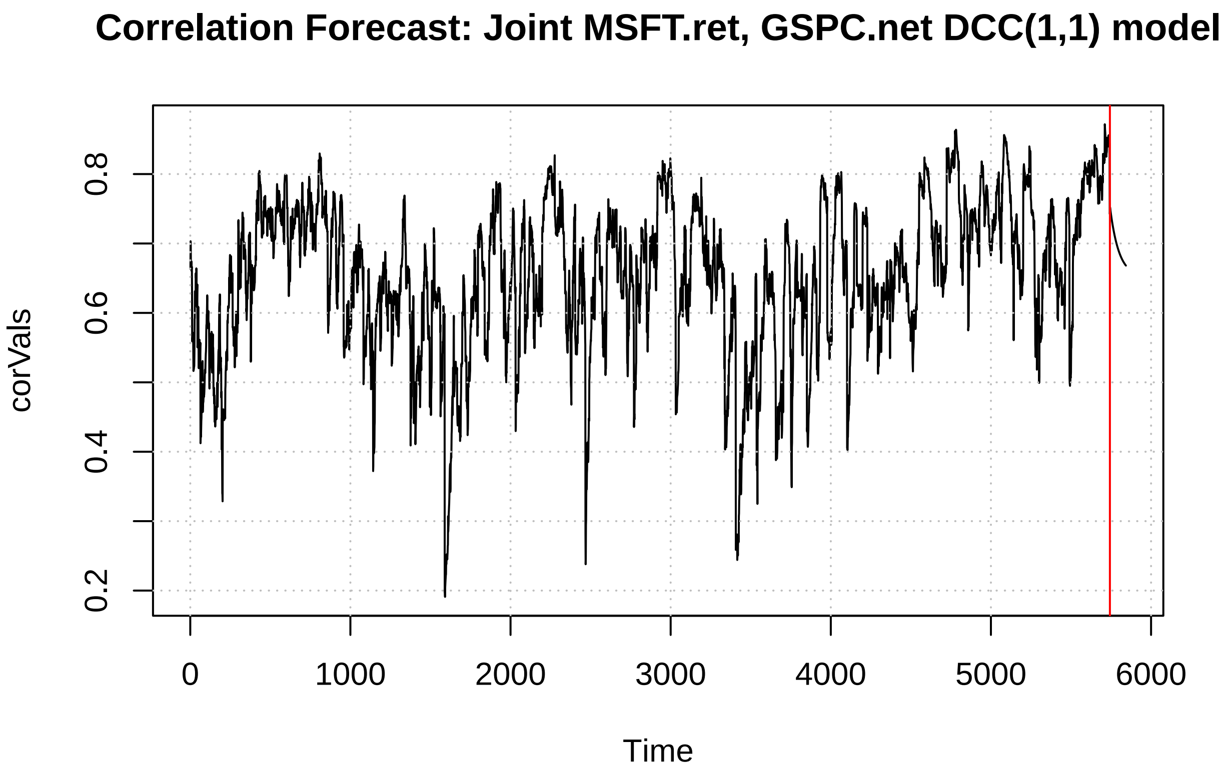
Compute (using **dccforecast()** function) and plot the first 100 h-step ahead forecasts of conditional covariance and correlation.

Computing forecasts with **dccforecast()**:

```
dcc.fcst <- dccforecast(dcc.fit, n.ahead=100)
```



The forecast of the joint MSFT.ret and GSPC.ret covariance is indicated by the right side fo the red vertical line. The forecast seems to trend downward towards the mean of almost 0 but not completely, showing the MSFT.ret data having less of an influence over GSPC.ret over the 100-day forecast period. This a slow reversion to the mean during the time period, indicated by the high persistence calculated by the sum of the model coefficients. Given the current climate of a rocky stock market, inflation, a post-pandemic era, and the war in Ukraine we are currently in a volatile time as shown towards the end of the data right before the forecast. That being said, as covariance is scale dependent and despite these small values, we cannot fully determine how much of an influence MSFT.ret has over GSPC.ret without the proper context of upper and lower bounds.



The forecast of the joint MSFT.ret and GSPC.ret correlation is indicated by the right side fo the red vertical line. Similar to the covariance forecast above, we're also seeing a downward trend where the MSFT.ret data would have an average influence of about 65% over the GSPC.ret data in terms of correlation. The reversion looks to be slow over the 100-day forecast period, indicated by the high persistence calculated from the sum of the model coefficients.

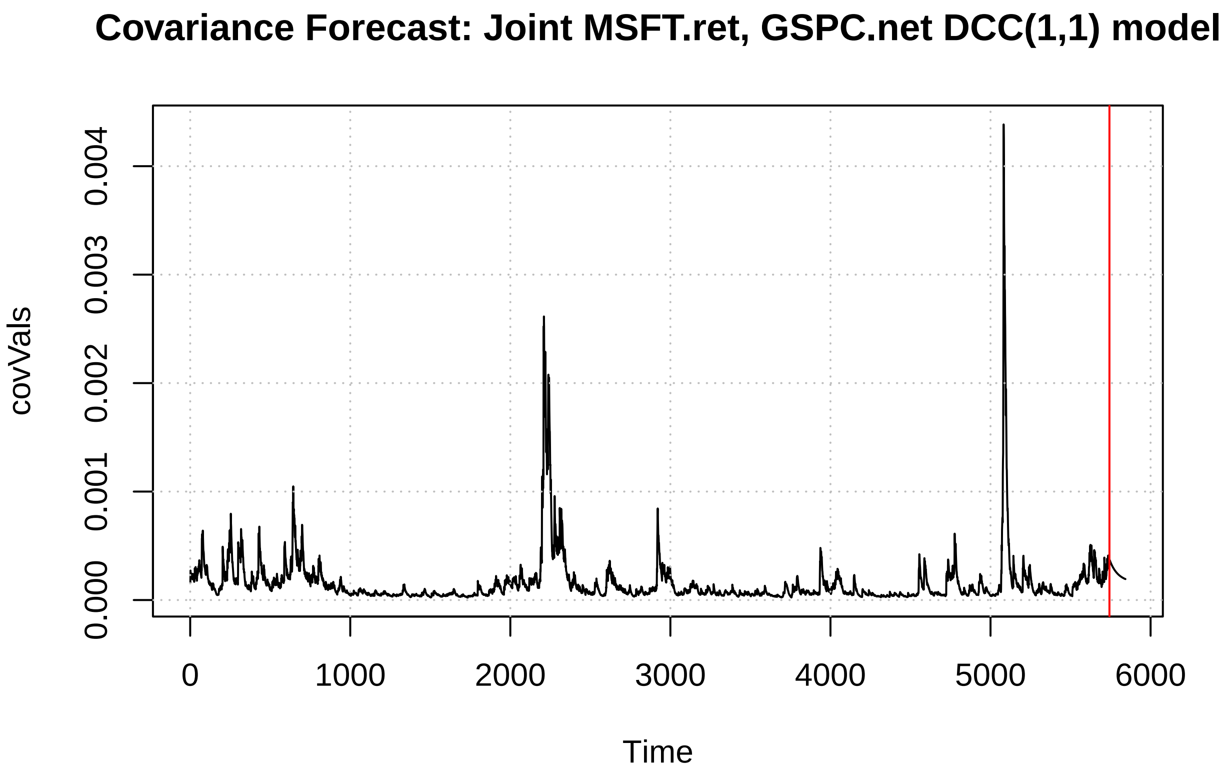
1.8. Choose which model you think is best and justify your choice.

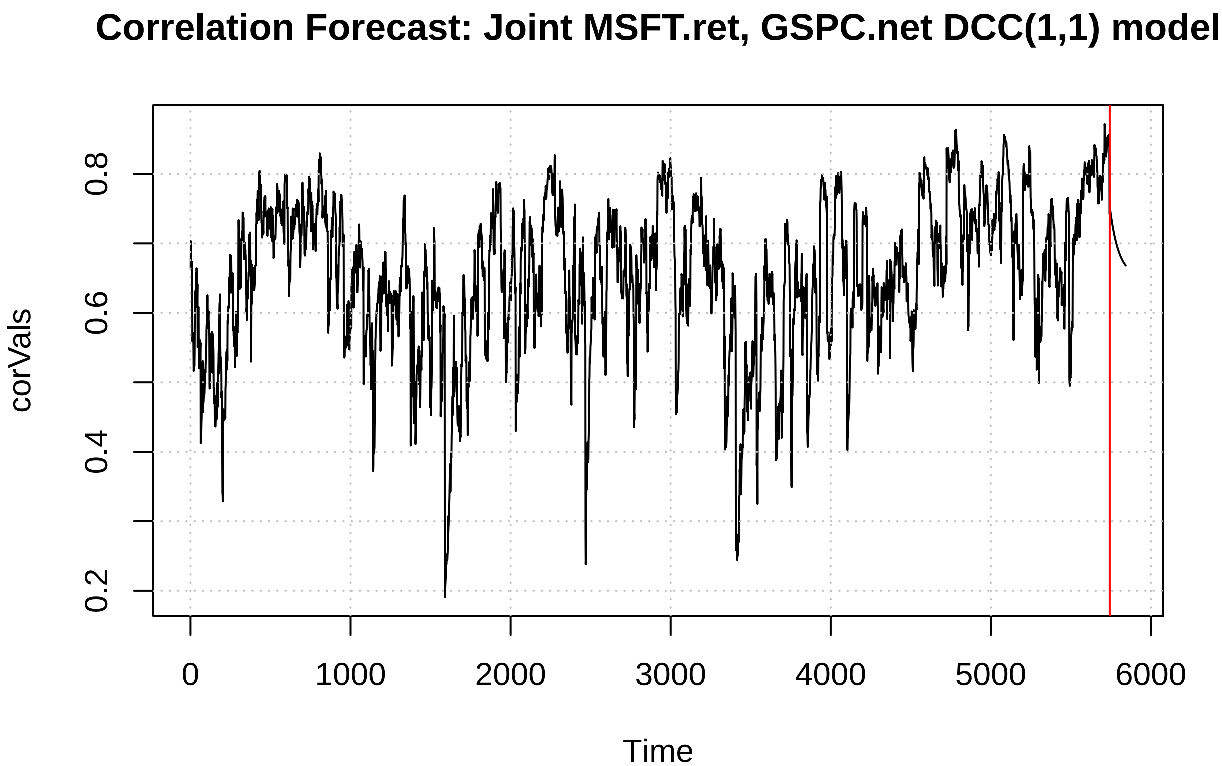
The univariate ARCH(5) models are not the best choice due to that their calculated persistence is lower compared to the other models (0.7764 for the MSFT.ret ARCH(5) model and 0.8477 for the GSPC.ret ARCH(5) model). While the GARCH(1,1) and joint normal-DCC(1,1) models all have high persistence calculated at just over 0.98, we would give the edge to the joint normal-DCC(1,1) model as the best one in that it has the lowest AIC rating (-12.392) over the univariate GARCH(1,1) models (-5.365 for the MSFT.ret GARCH(1,1) model and -6.432 for the GSPC.ret GARCH(1,1) model).

2. Report (20 points)

Write an executive summary of your Microsoft and S&P analysis outcomes.

(We will use the joint normal-DCC(1,1) model and its forecasts for the executive summary.)


Covariance Forecast: Joint MSFT.ret, GSPC.net DCC(1,1) model
covVals
0.000
0.001
0.002
0.003
0.004
0
1000
2000
3000
4000
5000
6000
Time



This forecast models attempts to predict how much of an influence Microsoft stock returns have over the S&P 500 Index returns. We look at the influence in terms of variance (or volatility) and correlation. Our forecast covers a 100-day period and is based on Microsoft stock and S&P 500 Index data retrieved from Yahoo! Finance from 2000-01-03 to 2022-10-31. From the plots above, the respective forecasts are shown at the right side of the vertical red line.

The average variance calculated is almost 0, which means on average Microsoft stock returns usually does not influence S&P 500 returns during times of non-volatility. That being said, we do see spikes of volatility where Microsoft returns can influence S&P 500 returns during major financial events, such as the .com tech loss of the early 2000s, the housing market crash in 2008, the start of the COVID-19 pandemic in March of 2020, and the current time of this writing dealing with inflation, stock market instability, a post-pandemic era, and the war in Ukraine. The 100-day forecast shows a downward trend towards the 0 mean but not a fast reversion as the model is found to have high persistence.

The average correlation calculated is about 65% which means Microsoft stock returns have a fairly strong influence over S&P 500 returns in terms of correlation, and at the time of this writing seem to have a stronger than average correlation due to the volatile times we are currently in that have an impact in the financial market. That being said, the 100-day forecast has a somewhat slow downward trend towards the average correlation between the two assets.

Our takes from the plots and forecasts above show how influential Microsoft returns have over S&P 500 returns, especially during times of volatility which can have a major impact on the financial market. Given that Microsoft is currently the second-highest-weighted stock in the S&P 500 at about 6%, major fluctuations to Microsoft shares will impact the S&P 500. Our forecast attempts to show how much of an impact in terms of returns. That being said, our forecasts do not anticipate any future major events happening such that they slowly revert back towards the respective means in variance and correlation.

We will update this model as we collect more financial data over time and as the market changes, as well as continue to improve it as our domain expands.