CW5

Code

31 October, 2022

Multivariate Conditional Correlation Models

```
# Sample Correlations - all stocks
stocks_corr <- cor(log_returns_demean)
knitr::kable(stocks_corr, digits=2)
```


	SP500	JPM	CITI	APPLE	MSFT	VZ	XOM	GE
SP500	1.00	0.75	0.74	0.70	0.77	0.48	0.60	0.58
JPM	0.75	1.00	0.87	0.42	0.47	0.38	0.58	0.57
CITI	0.74	0.87	1.00	0.42	0.46	0.33	0.60	0.58
APPLE	0.70	0.42	0.42	1.00	0.62	0.24	0.31	0.33
MSFT	0.77	0.47	0.46	0.62	1.00	0.32	0.33	0.33
VZ	0.48	0.38	0.33	0.24	0.32	1.00	0.34	0.29
XOM	0.60	0.58	0.60	0.31	0.33	0.34	1.00	0.51
GE	0.58	0.57	0.58	0.33	0.33	0.29	0.51	1.00

Select two stocks - MSFT and XOM

Note that in the lectures the vector of returns r(t) has dimension *N by 1*. Therefore, for the whole sample of returns r is *N by T*. However, in the code we will preserve the dimension of the xts dataframe, i.e., rows denote dates *T* and columns denote stocks *N*.

Multivariate Conditional Correlation Models

```
log_returns_demean_2 <- log_returns_demean[, c('MSFT', 'XOM')]

head(log_returns_demean_2) # Note that in the lectures the notation
        is transposed. R(t) is
```

```
##                    MSFT          XOM
## 2013-01-03 -0.014441540 -0.001974730
## 2013-01-04 -0.019847716  0.004450369
## 2013-01-07 -0.002826110 -0.011814985
## 2013-01-08 -0.006213824  0.006065923
## 2013-01-09  0.004679307 -0.004019093
## 2013-01-10 -0.009984170  0.010663956
```



```
tail(log_returns_demean_2)
```



```
##                    MSFT          XOM
## 2022-09-22  0.017504985 -0.004356076
## 2022-09-23 -0.013734080 -0.054856183
## 2022-09-26 -0.002932040 -0.021026612
## 2022-09-27 -0.005344088  0.020571490
## 2022-09-28  0.018565146  0.035573434
## 2022-09-29 -0.015874382 -0.002196960
```


Sample Correlations - two stocks

```
stocks_corr <- cor(log_returns_demean_2)
knitr::kable(stocks_corr, digits=2)
```

Multivariate Conditional Correlation Models

	MSFT	XOM
MSFT	1.00	0.33
XOM	0.33	1.00

GARCH(1,1) Univariate conditional volatilities for MSFT and XOM

```
GARCH_1_1 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 0), include.mean = FALSE))

GARCH_1_1_fit_M <- ugarchfit(spec = GARCH_1_1, data = log_returns_demean_2[, 1], solver = 'hybrid')
sigmaGARCH_1_1_M <- GARCH_1_1_fit_M@fit$sigma

GARCH_1_1_fit_X <- ugarchfit(spec = GARCH_1_1, data = log_returns_demean_2[, 2], solver = 'hybrid')
sigma2GARCH_1_1_X <- GARCH_1_1_fit_X@fit$sigma
```



```
# Summarize parameter coefficients
param <- as.table(t(rbind(c(coef(GARCH_1_1_fit_M)["omega"],coef(GARCH_1_1_fit_M)["alpha1"], coef(GARCH_1_1_fit_M)["beta1"]),c(coef(GARCH_1_1_fit_X)["omega"],coef(GARCH_1_1_fit_X)["alpha1"], coef(GARCH_1_1_fit_X)["beta1"]))))
                   colnames(param) = c('MSFT','XOM')
knitr::kable(param, digits=3)
```

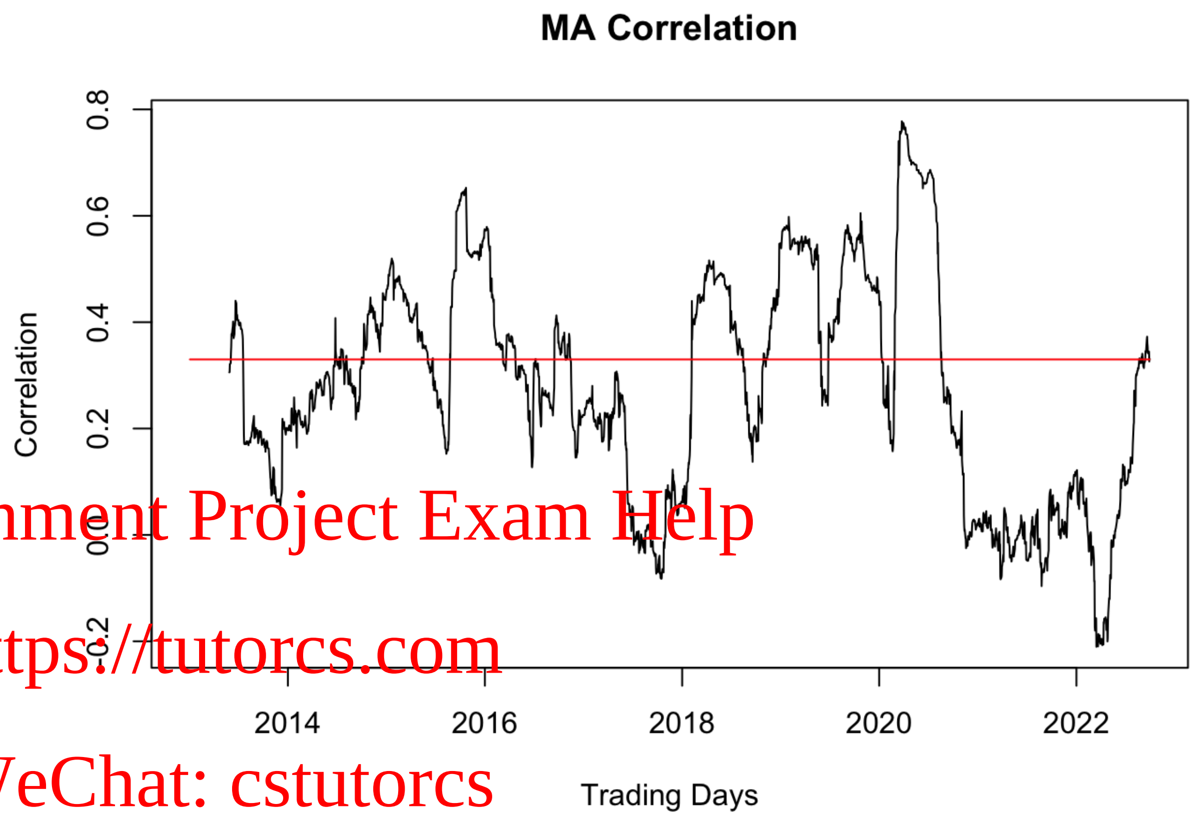
	MSFT	XOM
omega	0.000	0.000

	MSFT	XOM
alpha1	0.214	0.098
beta1	0.700	0.899

Moving Average Conditional Correlation - WE 100

```
corrMA <- rollapply(data = log_returns_demean_2, width = 100, FUN = function(x) cor(x)[2,1], by.column = FALSE)
  # Alternative: roll_cor(x = log_returns_demean_2[, 1], y = log_returns_demean_2[, 2], width = 100)
corrMA <- lag(corrMA, k = 1, na.pad = TRUE) # lagging by 1 to ensure that observations from t = 1 to t = we predict volatility at t = we + 1

par(mfrow=c(1,1))
plot(x = index(corrMA), y = corrMA, type = 'l', main = 'MA Correlation', xlab = 'Trading Days', ylab = 'Correlation')
lines(x = index(corrMA), y = stocks_corr['XOM', 'MSFT'] * rep(x = 1, t = length(index(corrMA))), col = 'red')
```

Multivariate Conditional Correlation Models



Multivariate EWMA

To determine the number of columns in the covariance matrix, we need to figure out how many parameters we are estimating in every time period. For any given time t we will be estimating the conditional variance of each asset, and the conditional covariances for N assets, this is $N+N(N-1)/2$. In our case, it is 3. Column 1 will hold the conditional variance of MSFT, column 2 will hold the conditional covariance between MSFT and XOM, and column 3 will hold the conditional variance for XOM.

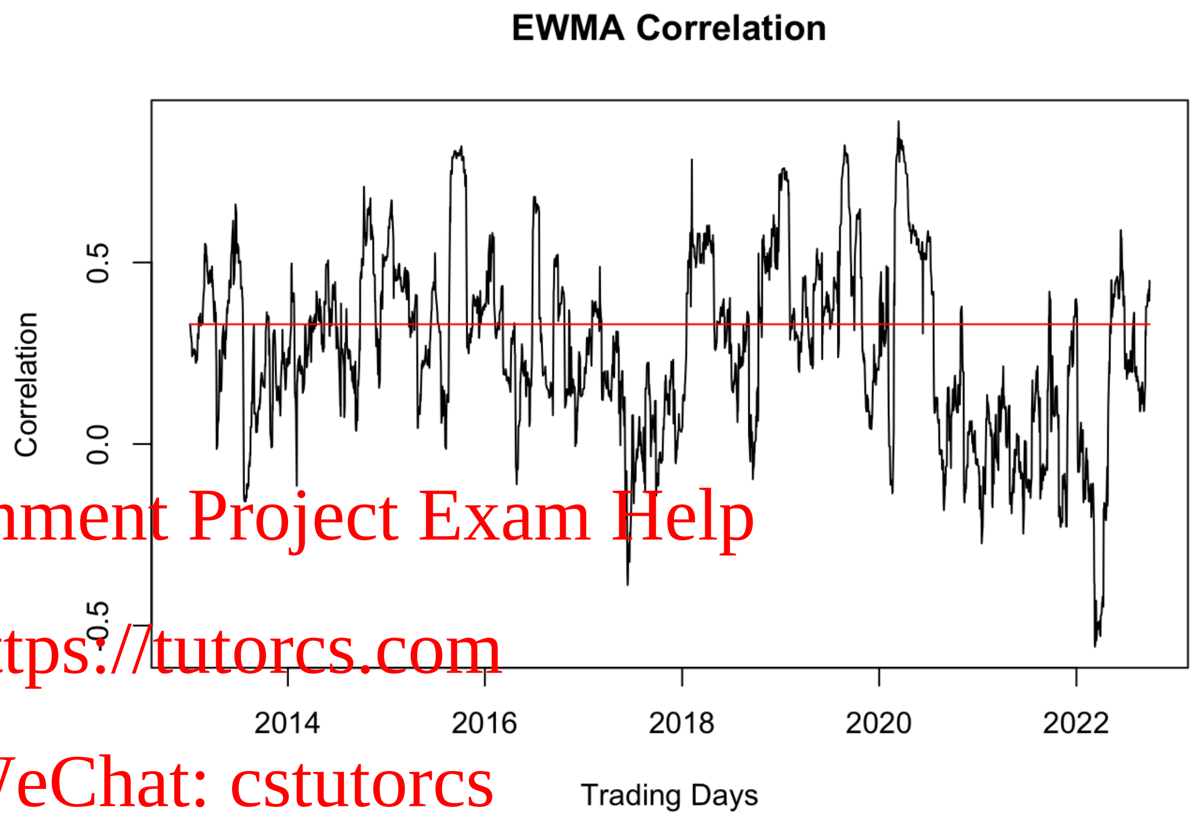
Multivariate Conditional Correlation Models

It is necessary to determine how to estimate the conditional covariances of the first period. For this, we will use the unconditional covariance of the sample and “burn” the first few observations. The effect of a given conditional covariance from a past period quickly dies out as time passes, so the effect of initializing the EWMA matrix with the unconditional sample covariance will not be a problem after a few time periods.

Multivariate Conditional Correlation Models

```
lambda = 0.94

TS <- dim(log_returns_demean_2)[1] # Number of days in sample
N <- dim(log_returns_demean_2)[2] # Number of stocks
X <- N+N*(N-1)/2 # Number of variance/covariances to estimate for each day

EWMA <- xts(matrix(nrow = TS, ncol = X), order.by = index(log_returns_demean_2)) # Vectorizing covariance matrix.
S <- cov(log_returns_demean_2) # Initialize matrix using sample covariance
EWMA[1, ] <- S[upper.tri(S, diag = TRUE)] # Using the fact that covariance matrix is symmetric, use upper triangle and include the diagonal. Alternative code to include diagonal S[!upper.tri(S)].

for (i in 2:length(index(log_returns_demean_2))) {
  S <- lambda * S + (1-lambda) * t(log_returns_demean_2[i-1,]) %*% log_returns_demean_2[i-1,]
  EWMA[i, ] <- S[!upper.tri(S)]
}

# Matrix multiplication %*%

sigmaEWMA1 <- sqrt(EWMA[, 1])
sigmaEWMA2 <- sqrt(EWMA[, 3])
corrEWMA <- EWMA[, 2] / (sigmaEWMA1 * sigmaEWMA2)
vcvEWMA <- cbind(sigmaEWMA1, corrEWMA, sigmaEWMA2)

plot(x = index(corrEWMA), y = corrEWMA, type = 'l', main = 'EWMA Correlation', xlab = 'Trading Days', ylab = 'Correlation')
lines(x = index(corrEWMA), y = stocks_corr['XOM', 'MSFT'] * rep(x = 1, t = length(index(corrEWMA))), col = 'red')
```

Multivariate Conditional Correlation Models



BEKK estimation

The BEKK model specification is

H(t) = CC'+A'r(t-1)r(t-1)'A + G'H(t-1)G

Multivariate Conditional Correlation Models

```
BEKK_model <- bekk_spec(model = list(type = "bekk", asymmetric = FA
        LSE),
                               init_values = NULL, signs = NULL, N = NULL)
BEKK_model_fit <- bekk_fit(spec = BEKK_model, data = log_returns_de
        mean_2,
                                  QML_t_ratios = FALSE, max_iter = 50, cri
        t = 1e-09)
vcvBEKK <- BEKK_model_fit$sigma_t
corrBEKK <- vcvBEKK[, 2]
summary(BEKK_model_fit)
```

Multivariate Conditional Correlation Models

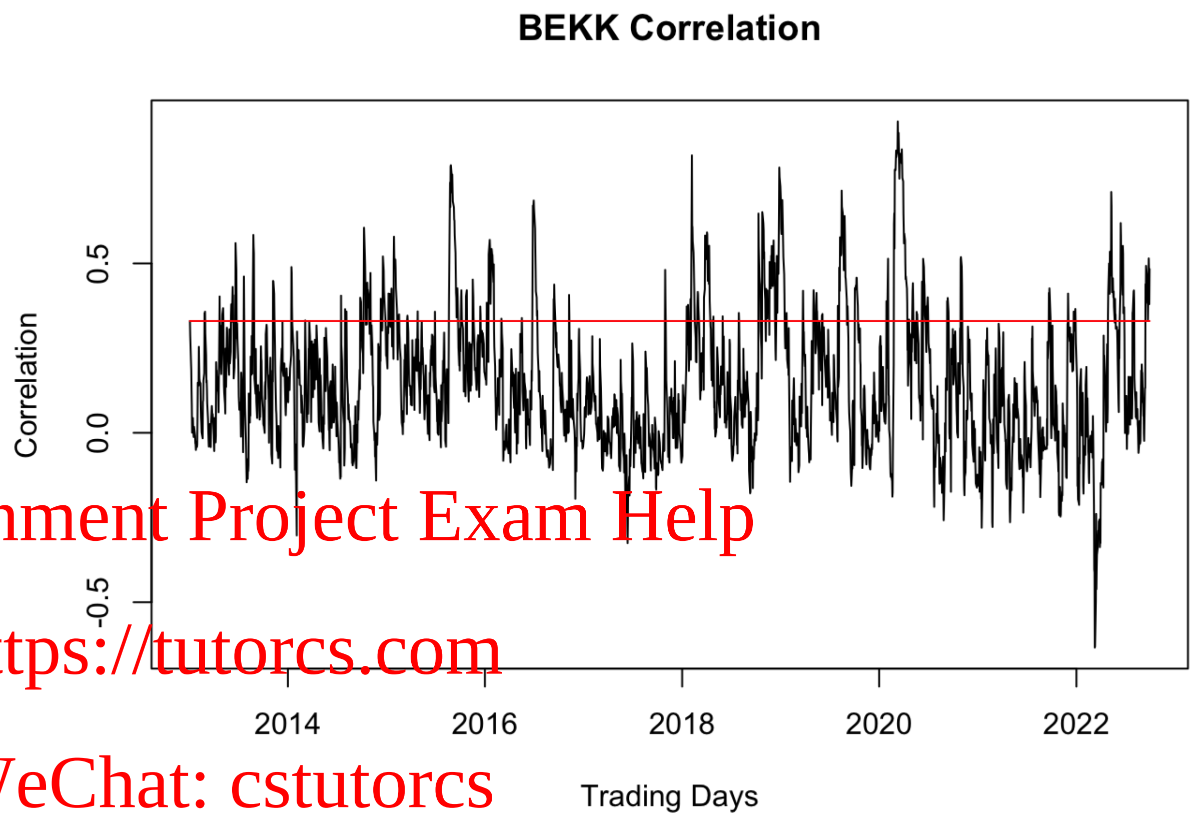
```
##
## BEKK estimation results
## -----------------------
## Log-likelihood: 14040.77
## BEKK model stationary: TRUE
## Number of BHHH iterations: 27
## AIC: -28062.53
## BIC: -28054.12
## Estimated paramater matrices:
##
## C
##              [,1]        [,2]
## [1,] 0.0057900648 0.000000000
## [2,] 0.0007465491 0.001144361
##
## A
##            [,1]       [,2]
## [1,] 0.41640423 0.05575069
## [2,] 0.05361679 0.25666507
##
## G
##              [,1]        [,2]
## [1,]  0.836270485 -0.02185699
## [2,] -0.009659171  0.96416645
##
## t-values of paramater matrices:
##
## C
##           [,1]     [,2]
## [1,] 21.271180 0.000000
## [2,]  3.643374 9.446941
##
## A
```

Multivariate Conditional Correlation Models

```
##              [,1]       [,2]
## [1,] 21.520093  3.545351
## [2,]  3.839503 22.979069
##
## G
##              [,1]        [,2]
## [1,] 63.379866   2.447653
## [2,]  1.809282 316.451287
```

```
plot(x = index(corrBEKK), y = corrBEKK, type = 'l', main = 'BEKK Co
        rrelation', xlab = 'Trading Days', ylab = 'Correlation')
lines(x = index(corrBEKK), y = stocks_corr['XOM', 'MSFT'] * rep(x =
        1, t = length(index(corrBEKK))), col = 'red')
```

Multivariate Conditional Correlation Models



The BEKK scalar model specification is

H(t) = CC'+ar(t-1)r(t-1)' + gH(t-1)

Multivariate Conditional Correlation Models

```
BEKK_models <- bekk_spec(model = list(type = "sbekk", asymmetric =
        FALSE),
                              init_values = NULL, signs = NULL, N = NULL)
BEKK_model_fits <- bekk_fit(spec = BEKK_models, data = log_returns_
        demean_2,
                                 QML_t_ratios = FALSE, max_iter = 50, cri
        t = 1e-09)

summary(BEKK_model_fits)
```

Multivariate Conditional Correlation Models

```
##
## Scalar BEKK estimation results
## -------------------------------
## Log-likelihood: 13999.58
## Scalar BEKK model stationary: TRUE
## Number of BHHH iterations: 37
## AIC: -27992.17
## BIC: -27973.75
## Estimated paramater matrices:
##
## C
##                  [,1]        [,2]
## [1,] 0.0027877905 0.000000000
## [2,] 0.0002826643 0.001869056
##
## a
## [1] 0.08284876
##
## g
## [1] 0.898595
##
## t-values of paramater matrices:
##
## C
##             [,1]     [,2]
## [1,] 24.170418  0.00000
## [2,]  2.974531 17.39567
##
## a
## [1] 17.86764
##
```

```
## g
## [1] 161.2876
```

DCC estimation

```
# Specify the default univariate GARCH model with no mean
xspec = ugarchspec(mean.model = list(armaOrder = c(0, 0), include.m
        ean = FALSE))
# Replicate it into a multispec() element
uspec = multispec(replicate(2, xspec))
# Define the specification for the DCC model
spec = dccspec(
  # GARCH specification
  uspec = uspec,
  # DCC specification
  dccOrder = c(1, 1),
  # Distribution, here multivariate normal
  distribution = 'mvnorm')

# Fit the specification to the data
res <- dccfit(spec, data = log_returns_demean_2)
# In sample conditional covariance
H <- res@mfit$H
#Output
res
```

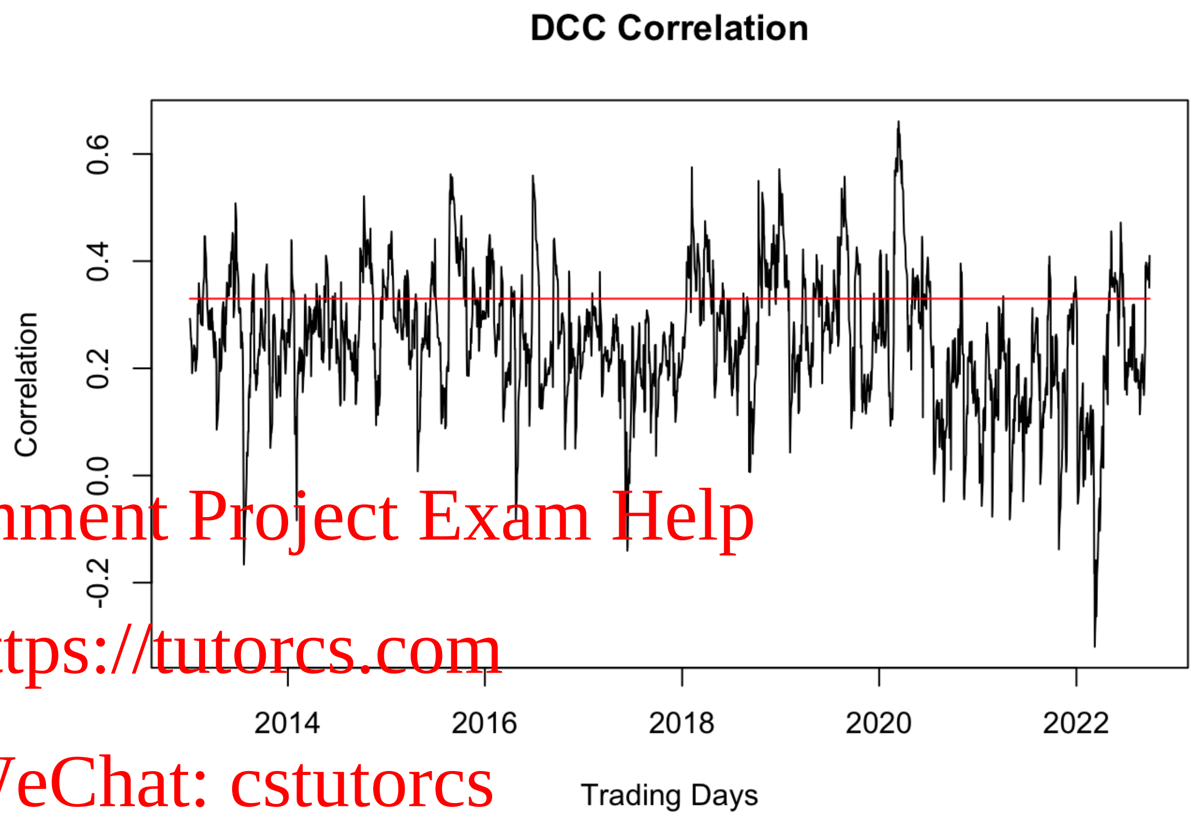
Multivariate Conditional Correlation Models

```
##
## *---------------------------------*
## *          DCC GARCH Fit          *
## *---------------------------------*
##
## Distribution          :  mvnorm
## Model                 :  DCC(1,1)
## No. Parameters        :  9
## [VAR GARCH DCC UncQ] : [0+6+2+1]
## No. Series            :  2
## No. Obs.              :  2453
## Log-Likelihood        :  14043.26
## Av.Log-Likelihood     :  5.72
##
## Optimal Parameters
## -----------------------------------
##                  Estimate  Std. Error  t value Pr(>|t|)
## [MSFT].omega     0.000028    0.000008  3.38122 0.000722
## [MSFT].alpha1    0.214350    0.050634  4.23332 0.000023
## [MSFT].beta1     0.700199    0.054944 12.74389 0.000000
## [XOM].omega      0.000002    0.000003  0.75729 0.448878
## [XOM].alpha1     0.097647    0.035504  2.75030 0.005954
## [XOM].beta1      0.898655    0.034189 26.28477 0.000000
## [Joint]dcca1     0.055343    0.016020  3.45454 0.000551
## [Joint]dccb1     0.858046    0.039232 21.87106 0.000000
##
## Information Criteria
## ---------------------
##
## Akaike       -11.443
## Bayes        -11.421
## Shibata      -11.443
## Hannan-Quinn -11.435
```

```
##
##
## Elapsed time : 4.07681
```

```
# In sample conditional correlations

DCCrho=xts(vector(length=dim(log_returns_demean_2)[1]), order.by = index(log_returns_demean_2))
for(i in 1:dim(log_returns_demean_2)[1]){
  DCCrho[i] =  H[1,2,i]/sqrt(H[1,1,i]*H[2,2,i])
}

Y <- dim(log_returns_demean_2)[1]
X <- dim(log_returns_demean_2)[2] + dim(log_returns_demean_2)[2] * (dim(log_returns_demean_2)[2] - 1) / 2
vcvDCC <- xts(matrix(nrow = Y, ncol = X), order.by = index(log_returns_demean_2))
vcvDCC[, 1] <- sqrt(H[ 1, 1,])
vcvDCC[, 2] <- DCCrho
vcvDCC[, 3] <- sqrt(H[ 2, 2,])

plot(x = index(DCCrho), y = DCCrho, type = 'l', main = 'DCC Correlation', xlab = 'Trading Days', ylab = 'Correlation')
lines(x = index(DCCrho), y = stocks_corr['XOM', 'MSFT'] * rep(x = 1, t = length(index(DCCrho))), col = 'red')
```

Multivariate Conditional Correlation Models

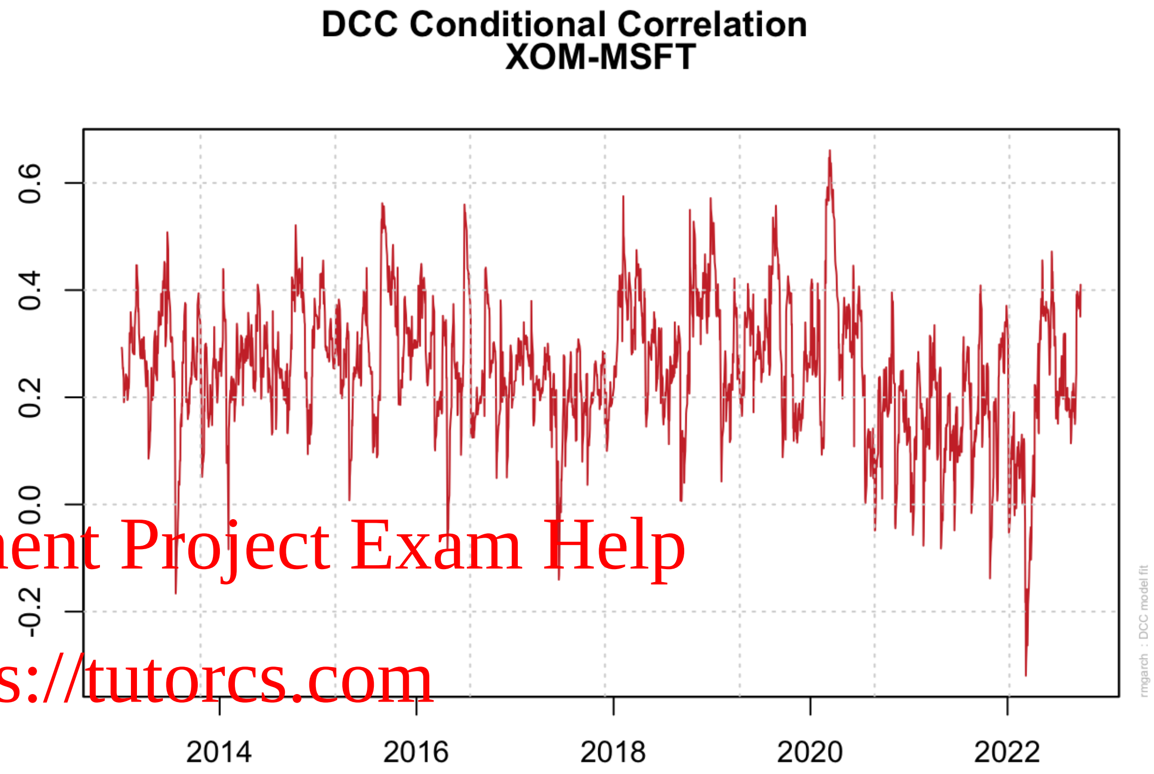


Calling the function plot() on a DCC object gives us a menu of options: Make a plot selection (or 0 to exit):

1. Conditional Mean (vs Realized Returns)
2. Conditional Sigma (vs Realized Absolute Returns)
3. Conditional Covariance
4. Conditional Correlation
5. EW Portfolio Plot with conditional density VaR limits

```
plot(res, which=4)
```

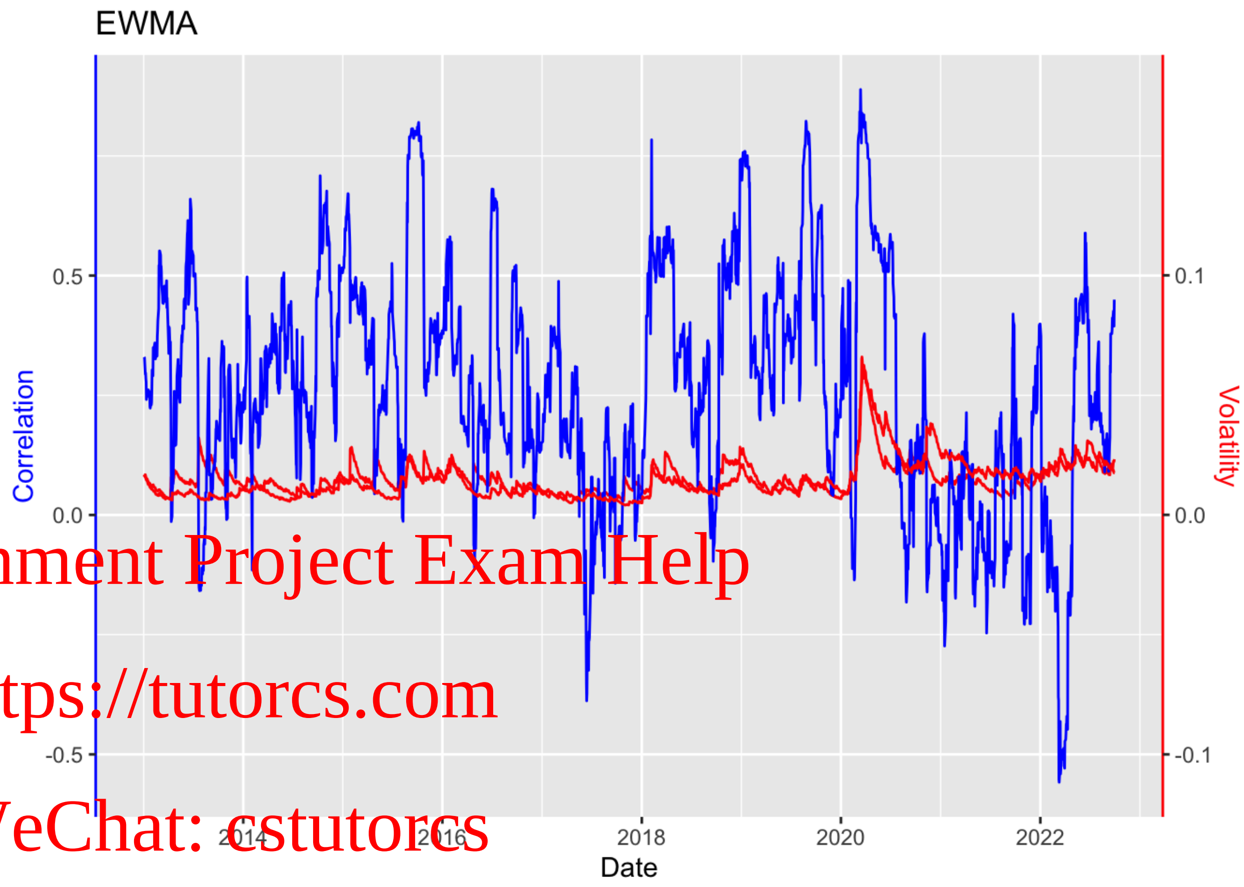
Multivariate Conditional Correlation Models



Comparing conditional volatility vs conditional correlation

```
ggplot(data = vcvEWMA, mapping = aes(x = index(vcvEWMA))) +
  xlab('Date') +
  ggtitle('EWMA') +
  geom_line(aes(y = vcvEWMA[, 1] * 5), color = 'red') +
  geom_line(aes(y = vcvEWMA[, 2]), color = 'blue') +
  geom_line(aes(y = vcvEWMA[, 3] * 5), color = 'red') +
  scale_y_continuous(name = 'Correlation', sec.axis = sec_axis(~./
        5, name = 'Volatility')) +
  theme(axis.line.y.left=element_line(color="blue")) +
  theme(axis.line.y.right=element_line(color="red")) +
  theme(axis.title.y.left = element_text(color = 'blue')) +
  theme(axis.title.y.right = element_text(color = 'red'))
```

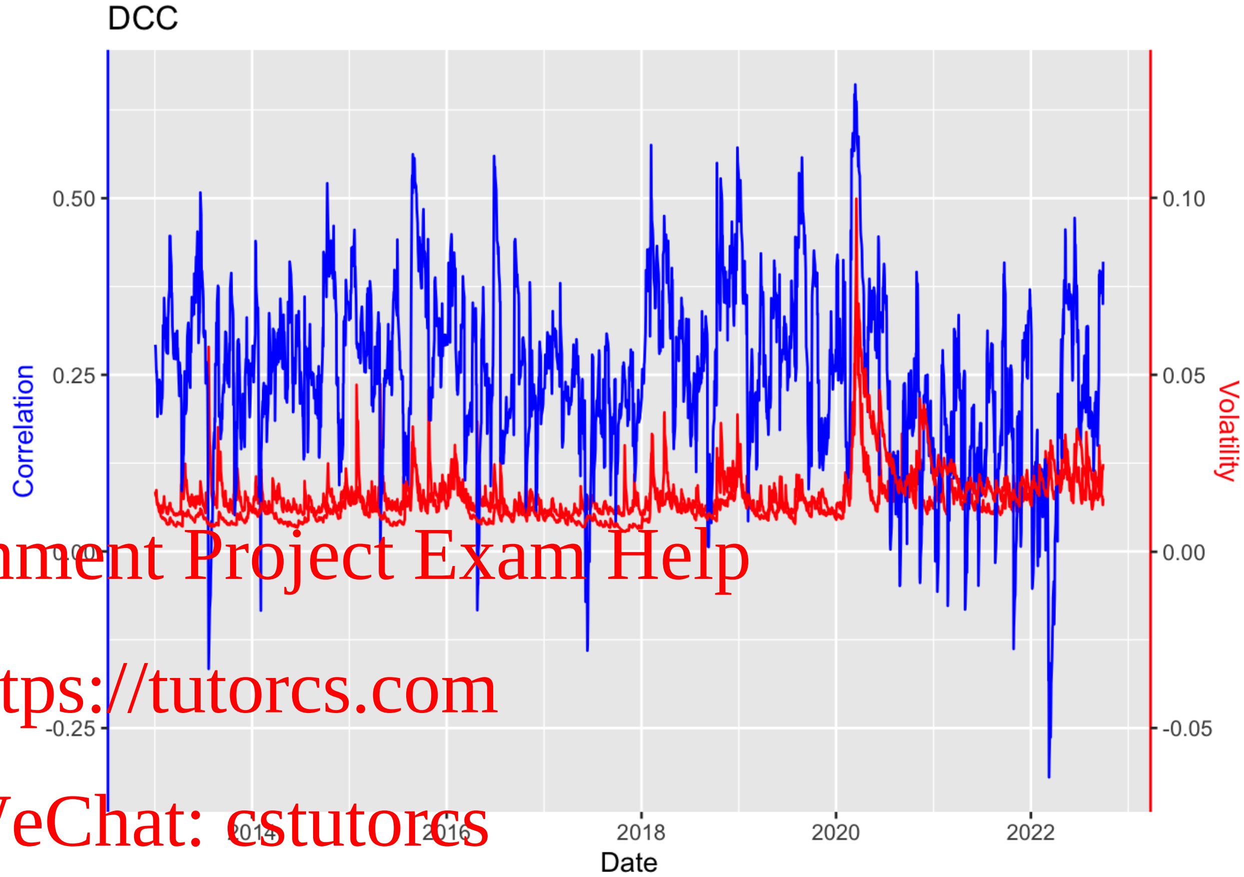
Multivariate Conditional Correlation Models



Multivariate Conditional Correlation Models

```
ggplot(data = vcvDCC, mapping = aes(x = index(vcvDCC))) +
  xlab('Date') +
  ggtitle('DCC') +
  geom_line(aes(y = vcvDCC[, 1] * 5), color = 'red') +
  geom_line(aes(y = vcvDCC[, 2]), color = 'blue') +
  geom_line(aes(y = vcvDCC[, 3] * 5), color = 'red') +
  scale_y_continuous(name = 'Correlation', sec.axis = sec_axis(~./
        5, name = 'Volatility')) +
  theme(axis.line.y.left=element_line(color="blue")) +
  theme(axis.line.y.right=element_line(color="red")) +
  theme(axis.title.y.left = element_text(color = 'blue')) +
  theme(axis.title.y.right = element_text(color = 'red'))
```

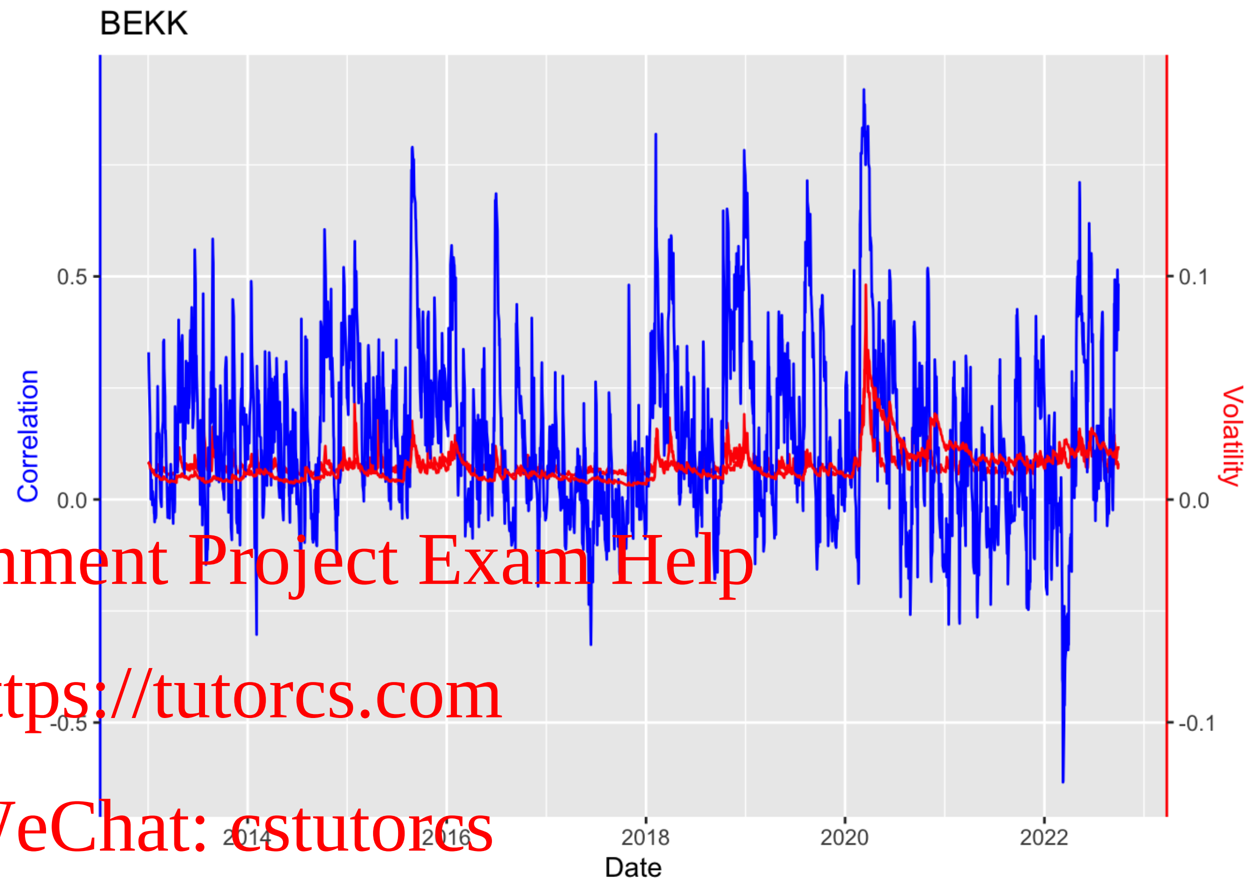
Multivariate Conditional Correlation Models



Multivariate Conditional Correlation Models

```
ggplot(data = vcvBEKK, mapping = aes(x = index(vcvBEKK))) +
  xlab('Date') +
  ggtitle('BEKK') +
  geom_line(aes(y = vcvBEKK[, 1] * 5), color = 'red') +
  geom_line(aes(y = vcvBEKK[, 2]), color = 'blue') +
  geom_line(aes(y = vcvBEKK[, 3] * 5), color = 'red') +
  scale_y_continuous(name = 'Correlation', sec.axis = sec_axis(~./
        5, name = 'Volatility')) +
  theme(axis.line.y.left=element_line(color="blue")) +
  theme(axis.line.y.right=element_line(color="red")) +
  theme(axis.title.y.left = element_text(color = 'blue')) +
  theme(axis.title.y.right = element_text(color = 'red'))
```

Multivariate Conditional Correlation Models




```
plot(x = index(DCCrho), y = vcvBEKK[,2], type = 'l', main = 'DCC/BEKK/EWMA Correlations', xlab = 'Trading Days', ylab = 'Correlation')
lines(x = index(DCCrho), y = vcvDCC[,2], col = 'red')
lines(x = index(DCCrho), y = vcvEWMA[,2], col = 'blue')
legend('topleft', legend = c('BEKK','DCC','EWMA'), col = c('black','red', 'blue'), lty=1)
```

Multivariate Conditional Correlation Models