MScFE 610 Econometrics (C19-S4)

Group Work Assignment Submission 2 M5

Kwadwo Amo-Addai (kwadwoamoad@gmail.com)

David Sabater Dinter (david.sabater@gmail.com)

Ruben Ohayon (rubensacha.ohayon@gmail.com)

Andrea Chello (chelloandrea@gmail.com)

Pasin Marupanthorn (oporkabbb@hotmail.com)

3.2.1. Volatility Analysis

Forecast Apple daily stock return using a GARCH model.

Source: Yahoo Finance

1. Select GARCH model (ARCH, GARCH-M, IGARCH, EGARCH, TARCH, multivariate GARCH etc). Explain your choice.

We use the historical stock prices of Apple Company from 1/1/2019 to 1/1/2020.. The adjusted close prices are used to calculate a daily return. The time series data is plotted in Figure 1. The autocorrelation is shown in Figure 2. As can be observed from Figure 1 and Figure 2, the process of the return is stationary. There are a few negative shocks in the series the worst around 20th day, so it is sufficient to use the simple model such as standard GRACH. There not exist the strongly sharp spikes in volatility. It contains only some small spikes and then vanish quickly. For the prediction view, we will use the model to predict only one step, so it is a good reason to use the simple model. The more complicated model, higher possibility to be overfitting model. Therefore, the **standard GARCH** is chosen to study in this section.

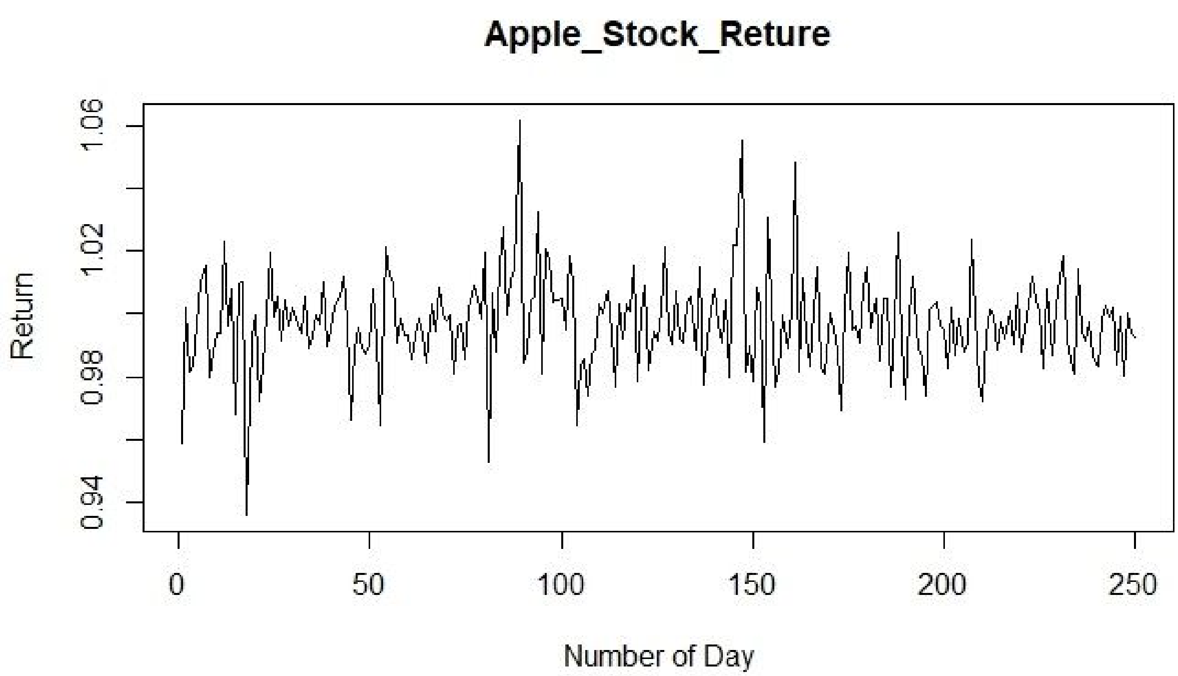


Figure 1. Time Series Data from Apple Company from 1/1/2019 to 1/1/2020

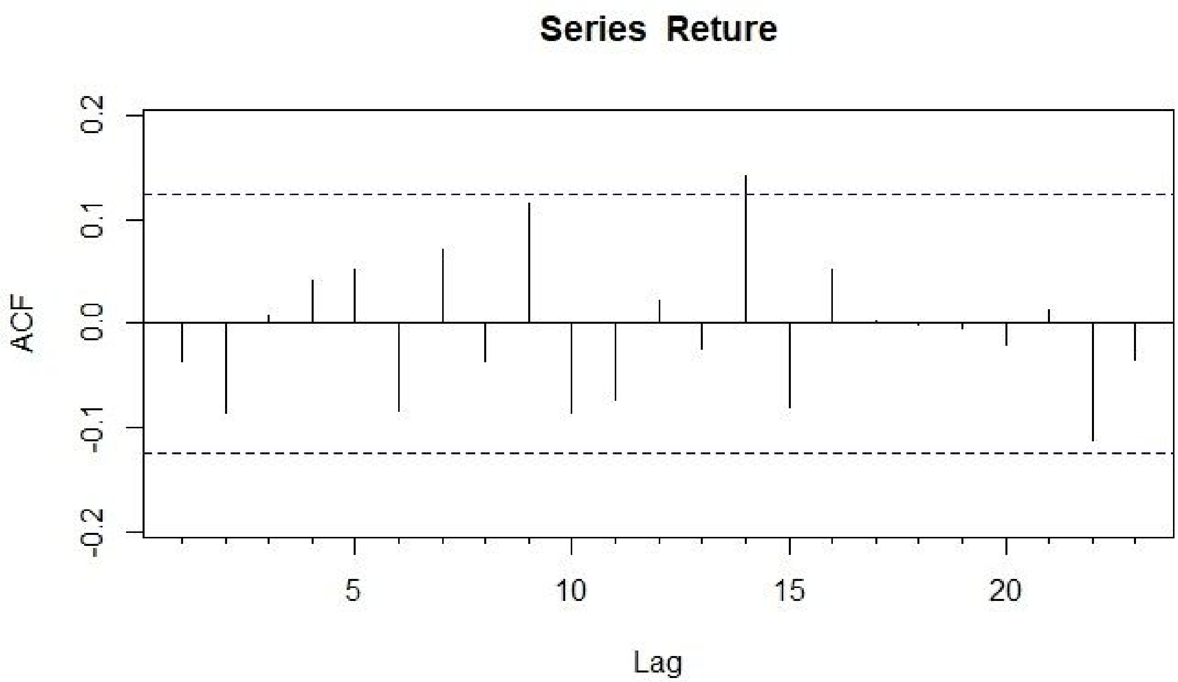


Figure 2. Autocorrelation of the Process

We start with the GARCH(1,1) as a minimum and vary the value of p and q in the model GARCH(1,1), i.e. p, q = 1,2,3,... . We found that GARCH(1,1) provide the significant p value for all parameters otherwise there are some parameters yielding high p value. Using the minimum parameter in model will help us to deduce the complexity and avoid overfitting.

The Analysis of GARCH(1,1) is shown in the table below. All of the residuals tests provide the good results. P value of Jarque-Bera and Shapiro-Wilk Test are significantly low referring to the residuals match to the normal distribution. The results obtained by Ljung-Box Test for R and R^2 refer to autocorrelations of a time series are not different from zero to much. For those reasons, the GARCH(1,1) is chosen to do the prediction in the next section.

Coefficient(s)			
mu	omega	alpha1	beta1
9.9628e-01	2.0498e-05	1.5478e-01	7.6500e-01

Estimate	Std.	Error	t value	Pr(>|t|)
mu	9.2.050e-05	8.234e-04	1209.968	<2e-16 ***
omega	2.050e-05	9.917e-06	2.067	0.0387 *
alpha1	1.548e-01	6.172e-02	2.508	0.0122 *
beta1	7.650e-01	6.552e-02	11.675	<2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Standardised Residuals Tests:

			Statistic	p-Value
Jarque-Bera Test	R	Chi^2	32.72661	7.825414e-08
Shapiro-Wilk Test	R	W	0.9776656	0.0005676868
Ljung-Box Test	R	Q(10)	9.13233	0.5195928
Ljung-Box Test	R	Q(15)	16.42716	0.3542438

Ljung-Box Test	R	Q(20)	17.59302	0.6141985
Ljung-Box Test	R^2	Q(10)	8.992109	0.5328526
Ljung-Box Test	R^2	Q(15)	11.60851	0.7083875
Ljung-Box Test	R^2	Q(20)	15.06495	0.7726791
LM Arch Test	R	TR^2	10.45102	0.5764565

2. Forecast next period daily return (t+1) using the chosen model. Select the timeframe in the analysis. Provide charts and comments.

```
# only needed in case you have not yet installed these packages

install.packages(c("quantmod","rugarch","rmgarch"))
library(quantmod); library(rugarch); library(rmgarch)

getSymbols("AAPL", from='2019-01-01', to='2020-01-01', src='yahoo')
chartSeries(AAPL, type="line", subset='2018', theme=chartTheme('white'))
```


```
df <- data.frame(AAPL)
head(df)
daily_returns <- dailyReturn(AAPL)

# UNIVARIATE GARCH MODEL

ug_spec = ugarchspec(); ug_spec
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model                     : sGARCH(1,1)
Variance Targeting      : FALSE

Conditional Mean Dynamics
-----------------------------------
Mean Model              : ARFIMA(1,0,1)
Include Mean            : TRUE
GARCH-in-Mean                   : FALSE
```

Conditional Distribution

Distribution : norm

Includes Skew: FALSE

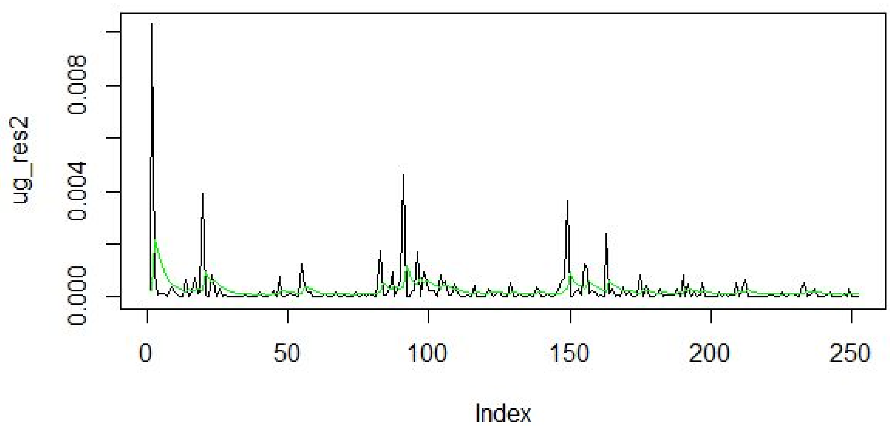
Includes Shape : FALSE

Includes Lambda : FALSE

MODEL ESTIMATION - Now that we have specified a model to estimate we need to find the best parameters

i.e. we need to estimate the model. This step is achieved by the ugarchfit function.

```
ugfit = ugarchfit(spec = ug_spec, data = daily_returns); ugfit
ug_var <- ugfit@fit$var   # save the estimated conditional variances
ug_res2 <- (ugfit@fit$residuals)^2   # save the estimated squared residuals
plot(ug_res2, type = "l")
lines(ug_var, col = "green")
```



MODEL FORECASTING

ugfore <- ugarchforecast(ugfit, n.ahead = 10); ugfore

Output:

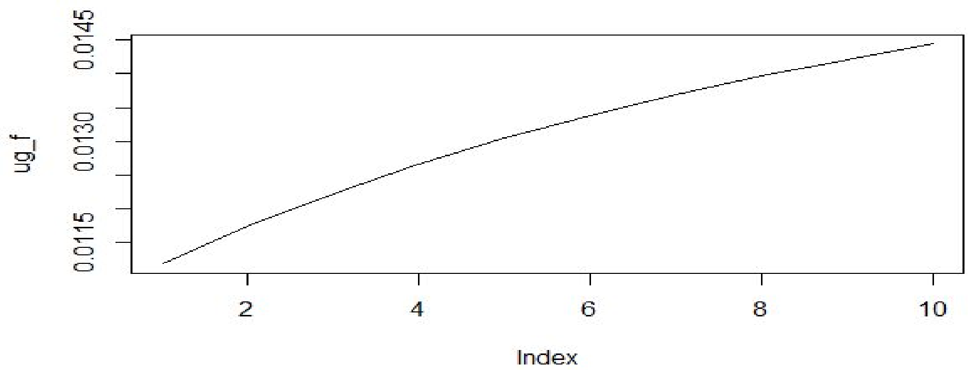
```
*------------------------------------*
*       GARCH Model Forecast         *
*------------------------------------*
Model: sGARCH
Horizon: 10
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=2019-12-31]:
       Series   Sigma
T+1  0.001641 0.01117
T+2  0.002067 0.01173
T+3  0.002395 0.01222
T+4  0.002646 0.01266
T+5  0.002839 0.01304
T+6  0.002987 0.01338
T+7  0.003101 0.01369
T+8  0.003189 0.01397
T+9  0.003256 0.01422
T+10 0.003307 0.01445
```

As you can see we have produced forecasts for the next ten days, both for the expected returns (Series) and for the conditional volatility (square root of the conditional variance).

Similar to the object created for model fitting, ugfore contains two slots (@model and @forecast) and you can use names(ugfore@forecast) to figure out under which names the elements are saved. For instance you can extract the conditional volatility forecast as follows:

ug_f <- ugfore@forecast$sigmaFor

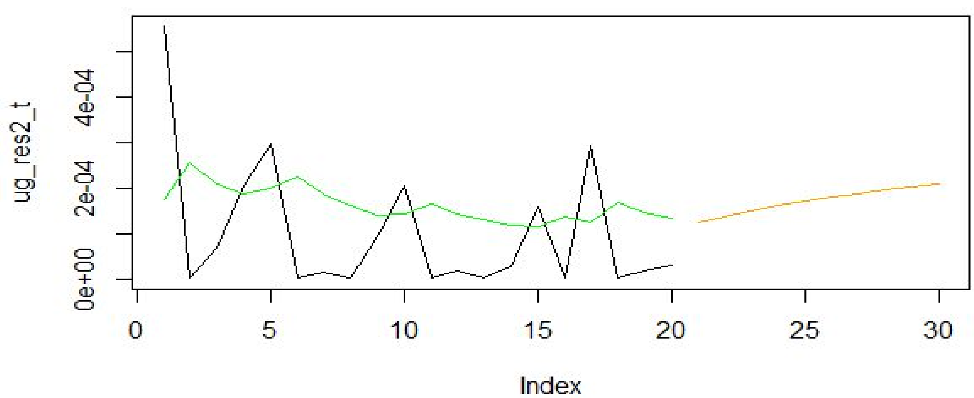
plot(ug_f, type = "l")



Note that the volatility is the square root of the conditional variance.

To put these forecasts into context let's display them together with the last 50 observations used in the estimation.

```
ug_var_t <- c(tail(ug_var,20),rep(NA,10))  # gets the last 20 observations
ug_res2_t <- c(tail(ug_res2,20),rep(NA,10))  # gets the last 20 observations
ug_f <- c(rep(NA,20),(ug_f)^2)

plot(ug_res2_t, type = "l")
lines(ug_f, col = "orange")
lines(ug_var_t, col = "green")
```



3.2.2. Multivariate Analysis

You can choose any currency pair you want.

You calculate the equilibrium FX for your local currency.

1. Indicate economic theories and models for calculating equilibrium FX.
2. Indicate macroeconomic variables used for calculating equilibrium FX.

ANSWER 1 and 2: There are several theories and models to calculate the equilibrium FX, in the context of this submission, we will focus on the models and theories described in (MacDonald, July 2000).

First is important to describe theoretically what we mean by equilibrium, as the value that helps us determine whether the exchanged currencies are correctly priced, or they represent some misalignment/shocks which will revert to the mean eventually.

The first approach to describe the equilibrium is by using the **Purchasing Parity Power (PPP)** of the countries for the currencies exchanged as the variables for the model. This basic model is represented below:

$$s_t = p_t + p^*{}_t$$

Where s_t represents a general of the equilibrium exchange rate, as the log of the spot exchange rate (home currency price of a unit of foreign currency), p_t is the log of the domestic price level, and asterisk denotes a foreign variable.

For the PPP model to work there shouldn't be impediments to international trade so the only factors affecting the prices would be the offer and demand for each of the currencies exchanged, clearly this is not the case, whoever this model is typically used as a baseline to compare against other models to represent the equilibrium more accurately. The **Monetary Extension of PPP** tries to include other factors and not just the offer and demand of the currencies exchanged, determining the prices of each country by imposing continuous money market clearing.

The next theory and model analysed is the **Capital Enhanced Equilibrium Exchange Rate (CHEER)**, which combines the extended PPP model together with the Uncovered Interest Rate Parity (UIP). As per (Investopedia), the UIP theory states that the difference in interest rates between two countries will equal the relative change in currency foreign exchange rates over the same period.

The Formula for Uncovered Interest Rate Parity (UIP) is:

$F_0 = S_0 \frac{1+i_c}{1+i_b}$

Where:

F_0 = Forward rate

S_0 = Spot Rate

i_c = Interest rate of country c

i_b = Interest rate of country b

The last model analysed is **Behavioural Equilibrium Exchange Rates (BEER)**, this approach advocates that both current and capital account items of the balance of payments are important determinants of the evolution of the real exchange rate.

Since the BEER model uses real exchange rates as variables, it could be constructed using a variety of estimators. The particular estimator used by Clark and MacDonald (MacDonald, July 2000) is the Vector Error Correction Mechanism (VECM).

3. Explain the connection between linear regression and Vector Error Correction (VEC).

The connection between linear regression and Vector Error Correction stems from the problem of spurious regressions when working with non-stationary time-series. Spurious regression arises when we regress a unit root process on an independent unit root process, and this causes significant results and a well-fitted model even if there is no real-world relationship between the variables of interest. The one and only case in which it is valid to regress one unit-root process onto another is if there exists a linear combination of the two processes such that the resulting *cointegrating relationship* is integrated to an order of 0, i.e. is stationary. This process is known as *Cointegration*, and is used to uncover the long run dynamics of a non-stationary time-series.

Given two non-stationary variables Y_t, X_t such that:

$$Y_t, X_t \sim I(1)$$

Regressing them onto each other yields two relationships:

1. **Levels:** which makes sense only if the linear combination of the two variables yields a cointegrating relationship. In such a case we can think of this linear regression as the equilibrium level of the variable Y_t. Otherwise, this would yield a spurious regression.

$$Y_t^E = \alpha + \beta X_t + \varepsilon_t$$

The cointegrating relationships are given by the vector β such that, for $k = 2$:

$$Z_T = [Y_t X_t]' \sim I(1) \qquad \beta' = [1 \ -\beta_2]$$

$$Z_T\beta' = \left[1 - \beta_2\right] \left[Y_t X_t\right]' = Y_t - \beta_2 X_t \sim I(0)$$

2. **Differences:** which yields the short-run dynamics of how the time-series variables interact with each other and is always stationary.

$$\Delta Y_t = \delta \Delta X_t + u_t$$

If there exists no cointegrating relationship between the two variables, i.e. no linear combination of Y_t, X_t such that the resulting relationship is stationary, the causal directionality between the regression in levels and the one in differences is unilateral. This means that if there is a long-term trend in levels then this is also likely to be present in the differences, but not vice-versa. Therefore, it is important to look for cointegrating relationships in order to prove there is a long-term relationship between the two non-stationary variables in both levels and differences.

This is achieved by using the cointegration approach developed by Engle and Granger which consists of modelling an error correction model which accounts for both the short-run dynamics (embedded in the differences relationship) and the long-run dynamics (embedded in the levels relationship at equilibrium). This approach however has several limitations, namely that is only considers one cointegrating relationship at a time. Therefore, it is useful instead to use the Vector Error Correction Model (VECM) to make sense of the short-run and long-run dynamics of non-stationary variables across several cointegrating relationships, i.e. linear combinations. This is why we first start with a VAR process and manipulate the terms algebraically to arrive at the VECM process.

VAR(P): $$X_t = A_0 + \sum_{i=1}^{p} A_i X_{t-i} + \varepsilon_t$$

VECM(P): $$\Delta X_t = A_0 + \Pi X_{t-1} + \sum_{i=1}^{p} C_i \Delta X_{t-i} + \varepsilon_t$$

Using the derivation of the VECM is useful when uncovering the cointegrating relationships between variables in a multivariate setting across time which allows us to look at the interactions of the short-run and long-run dynamics of the cointegrated processes. The number of cointegrating relationships is determined by the rank of the matrix Π, which has to be between the number of variables n and 0, i.e. 0 < r < n. Once the number of cointegrating relationships has been determined, we can decompose the matrix Π into a product of two matrices, **AB**. This is useful in so much as it allows us to treat the matrix as the relationship we illustrated above with $Z_T\beta'$.

$$\Pi = A * B$$

B = Represents r linearly independent rows that when multiplied with X_t yields r stationary long-term relationships.

A = Represents the response of the changes in each variable given by deviations from the long-term relationships in BX_t.

More succinctly, and in the case $r = 1$, i.e. there is only one cointegrating relationship:

$$\Delta X_{it} = \alpha_i U_{t-1} + \sum_{i=1}^{p} C_{i1} \Delta X_{t-i} + \varepsilon_{it}$$

U_{t-1} = Deviation from the long-run equilibrium to which there is an error correcting mechanism that pushes these deviations toward the long-run equilibrium.

α_i = Speed of adjustment of short-run shocks to the process to the long-term equilibrium.

$C_{i1} \Delta X_{t-i}$ **=** Captures the short-run dynamics of the process.

Therefore, we can see that the connection between linear regression and the VECM process is one of investigating both the short-run and long-run dynamics of non-stationary variables that have cointegrating relationships.

4. Calculate equilibrium FX using VEC. You can use the Behavioural Equilibrium Exchange Rate (BEER) approach. Comment results.

```
library(readxl)
library(urca)
library(tsDyn)

euro_dollar_rates = "/Users/Ruben/Desktop/DEXUSEU.xls"
d_one_month_rate <- euro_dollar_rates$WED1
d_three_month_rate <- euro_dollar_rates$WED3
d_six_month_rate <- euro_dollar_rates$WED6

d_rates <- cbind(d_one_month_rate, d_three_month_rate, d_six_month_rate)

jotest1 <- ca.jo(d_rates, type ="eigen", K=9, ecdet="none", spec = "longrun")
summary(jotest1)

jotest2 <- ca.jo(d_rates, type ="trace", K=9, ecdet="none", spec = "longrun")
summary(jotest2)
```

```
########################
# Johansen-Procedure #
########################

Test type: maximal eigenvalue statistic (lambda max) , with linear

Eigenvalues (lambda):
[1] 0.146156486 0.065201628 0.006132301

Values of teststatistic and critical values of test:

           test 10pct  5pct  1pct
r <= 2 |   5.41  6.50  8.18 11.65
r <= 1 |  59.27 12.91 14.90 19.19
r = 0  | 138.89 18.90 21.07 25.75

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                     d_one_month_rate.l9 d_three_month_rate.l9
d_one_month_rate.l9            1.0000000              1.000000
d_three_month_rate.l9         -1.6015884              2.904978
d_six_month_rate.l9            0.6080626             -3.886571
                     d_six_month_rate.l9
d_one_month_rate.l9             1.000000
d_three_month_rate.l9          -1.257257
d_six_month_rate.l9             3.767824

Weights W:
(This is the loading matrix)

                    d_one_month_rate.l9 d_three_month_rate.l9
d_one_month_rate.d           -0.5234407           -0.03698967
d_three_month_rate.d         -0.1128012           -0.03317527
d_six_month_rate.d           -0.1715670           -0.01719750
                    d_six_month_rate.l9
d_one_month_rate.d        -0.0002448355
d_three_month_rate.d      -0.0005654808
d_six_month_rate.d        -0.0008244080
```

```
#######################
# Johansen-Procedure #
#######################

Test type: trace statistic , with linear trend

Eigenvalues (lambda):
[1] 0.30849636 0.10483943 0.04482237

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 2 |  40.26  6.50  8.18 11.65
r <= 1 | 137.50 15.66 17.95 23.52
r = 0  | 461.39 28.71 31.52 37.22

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                      d_one_month_rate.l9 d_three_month_rate.l9
d_one_month_rate.l9             1.0000000              1.000000
d_three_month_rate.l9          -1.3010037              7.099073
d_six_month_rate.l9             0.3439403             -7.898395
                      d_six_month_rate.l9
d_one_month_rate.l9              1.000000
d_three_month_rate.l9            7.873342
d_six_month_rate.l9              2.328653

Weights W:
(This is the loading matrix)

                     d_one_month_rate.l9 d_three_month_rate.l9
d_one_month_rate.d            -2.8898995           -0.08126170
d_three_month_rate.d          -0.6583075           -0.07096291
d_six_month_rate.d            -0.6545841            0.03952592
                     d_six_month_rate.l9
d_one_month_rate.d           -0.02271754
d_three_month_rate.d         -0.03119485
d_six_month_rate.d           -0.03447150
```

The Johansen tests we conducted show strong cointegration.

VECM <- VECM(d_rates, 1, r = 2, include = "const", estim="ML",LRinclude = "none")
summary(VECM)

```
#############
###Model VECM
#############
Full sample size: 888   End sample size: 886
Number of variables: 3  Number of estimated slope parameters 18
AIC -15291.93   BIC -15196.19   SSR 20.12427
Cointegrating vector (estimated by ML):
   d_one_month_rate d_three_month_rate d_six_month_rate
r1     1.000000e+00                  0       -0.9827959
r2     5.829457e-17                  1       -0.9930257


                            ECT1                ECT2
Equation d_one_month_rate   -0.3478(0.0325)*** 0.3937(0.0585)***
Equation d_three_month_rate -0.1019(0.0242)*** -0.0007(0.0436)
Equation d_six_month_rate   -0.0965(0.0256)*** 0.0389(0.0461)
                            Intercept           d_one_month_rate -1
Equation d_one_month_rate   -0.0112(0.0039)**  0.0278(0.0517)
Equation d_three_month_rate -0.0127(0.0029)*** -0.1679(0.0385)***
Equation d_six_month_rate   -0.0094(0.0031)**  -0.1466(0.0408)***
                            d_three_month_rate -1 d_six_month_rate -1
Equation d_one_month_rate   0.0414(0.1243)        -0.0079(0.0986)
Equation d_three_month_rate 0.3699(0.0926)***     -0.0485(0.0734)
Equation d_six_month_rate   0.2265(0.0979)*       0.1039(0.0777)
```

References

- (MacDonald, July 2000) - Concepts to Calculate Equilibrium Exchange Rates: An Overview, Discussion paper 3/00 Economic Research Group of the Deutsche Bundesbank. https://www.bundesbank.de/resource/blob/703012/5176ff771c87ec257f78abd1db0c3f9e/mL/2000-07-03-dkp-03-data.pdf
- (UIP, Investopedia) - https://www.investopedia.com/terms/u/uncoveredinterestrateparity.asp
- https://cran.r-project.org/web/packages/rmgarch/vignettes/The_rmgarch_models.pdf
- L. Bauwens, S. Laurent, and J.V.K. Rombouts. Multivariate garch models: A survey. Journal of Applied Econometrics, 21(1):79–109, 2006.

- P.H. Boswijk and R. van der Weide. Method of moments estimation of go-garch models. Journal of Econometrics, 163(1):118–126, 2011.
- T. E. Conine and M. J. Tamarkin. On diversification given asymmetry in returns. Journal of Finance, 36(5):1143–1155, 1981.
- Hsing Fang and Lai Tsong-Yue. Co-kurtosis and capital asset pricing. Financial Review, 32(2): 293–307, 1997.
- C. Harvey and A. Siddique. Conditional skewness in asset pricing tests. Journal of Finance, 55 (3):1263–1295, 2000.
- H. Joe. Multivariate models and dependence concepts, volume 73. Chapman & Hall/CRC, 1997.
- E. Jondeau and M. Rockinger. The copula-garch model of conditional dependencies: An international stock market application. Journal of International Money and Finance, 25(5):827–853, 2006.
- M. Junker and A. May. Measurement of aggregate risk with copulas. Econometrics Journal, 8 (3):428–454, 2005.
- DX Li. On default correlation: A copula function approach. Journal of Fixed Income, 9(4): 43–54, 2000.
- Y. Malevergne and D. Sornette. Testing the gaussian copula hypothesis for financial assets dependences. Quantitative Finance, 3(4):231–250, 2003.
- R.S. Sears and K.C.J. Wei. Asset pricing, higher moments, and the market risk premium: A note. Journal of Finance, 40:1251–1253, 1985.Y.K. Tse and A.K.C. Tsui. A multivariate generalized autoregressive conditional heteroscedasticity model with time-varying correlations. Journal of Business and Economic Statistics, 20 (3):351–362, 2002.
- R. van der Weide. Wake me up before you go-garch. In Computing in Economics and Finance. Society for Computational Economics, 2004.
- K. Zhang and L. Chan. Efficient factor garch models and factor-dcc models. Quantitative Finance, 9(1):71–91, 2009.