Week 12 Tutorial

ECON90033 - 2023 Semester 2

Josh Copeland

Completed on 22 October 2023

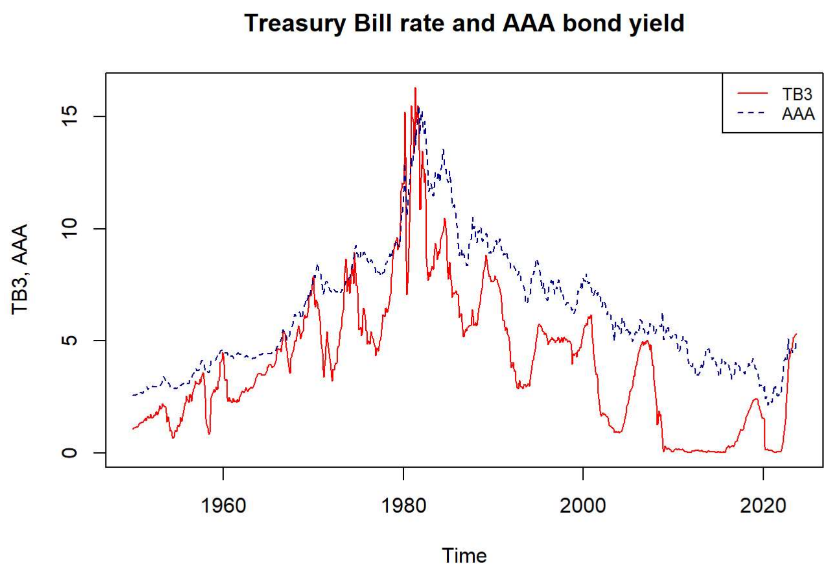
Exercise 1

a) Import the data, graph and interpret it.

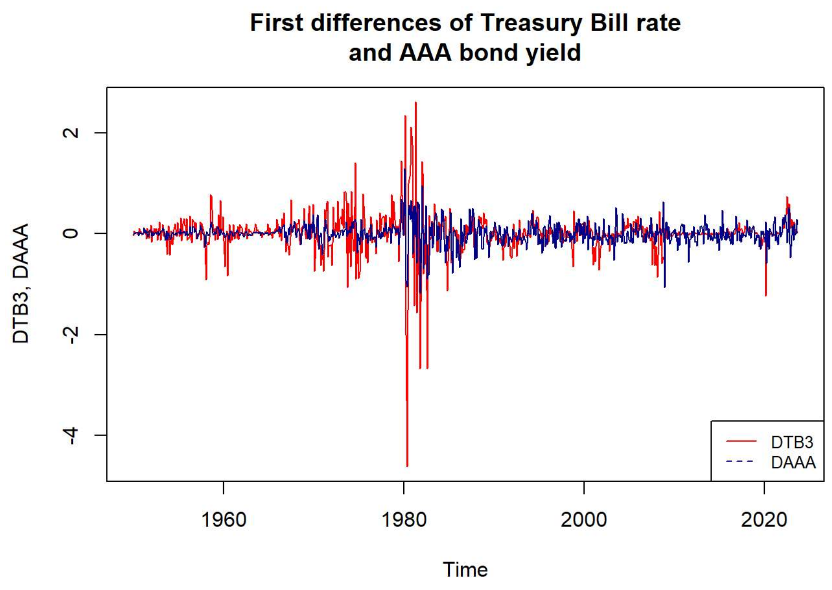
It’s clear both series are non-stationary, but they follow similar time paths and therefore may share a common stochastic trend.

However, the differenced series do not contain trends. Therefore, their means might be stationary.

```
e1 <- read_excel("C:/Users/joshc/Documents/2023S2/ECON90033/Tutorials/Week 12/t12e1.xlsx", sheet
= "Data")

TB3 <- e1 %>%
  pull(TB3) %>%
  ts(start = c(1949, 12), end = c(2023,9), frequency = 12)

DTB3 <- diff(TB3)



AAA <- e1 %>%
  pull(AAA) %>%
  ts(start = c(1949, 12), end = c(2023,9), frequency = 12)

DAAA <- diff(AAA)


plot.ts(TB3, ylab = "TB3, AAA", main = "Treasury Bill rate and AAA bond yield",
col = "red")
lines(AAA, col = "darkblue", lty = 2)
legend("topright", legend = c("TB3", "AAA"),
col = c("red", "darkblue"), lty = 1:2, cex = 0.8)
```



```
plot.ts(DTB3, ylab = "DTB3, DAAA", main = "First differences of Treasury Bill rate
and AAA bond yield", col = "red")
lines(DAAA, col = "darkblue")
legend("bottomright", legend = c("DTB3", "DAAA"),
col = c("red", "darkblue"), lty = 1:2, cex = 0.8)
```



To confirm these observations above, perform the ADF and kPSS tests on the levels and first differences of TB3 and AAA.

Based on the plots, the data generating plots certainly don't have a single linear trend component. It might have a broke nlinear trend components. For this reason you need to conduct tests with both an intercept (Model 2) and both an intercept and a trend (Model 3) for the level series. For the differenced series we use Model 2.

We don't discuss the output in detail, but they show that both variables are integrated of order 1 (i.e. I(1)) no matter of the significance level is 10% or 5%.

```
# DF test
#Library(urca)
adf.TB3 = ur.df(TB3, type = "drift", selectlags = "BIC")
summary(adf.TB3)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8227 -0.0830 -0.0155  0.1005  2.6016
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.049106   0.020539   2.391  0.01702 *
## z.lag.1     -0.011365   0.004047  -2.808  0.00509 **
## z.diff.lag   0.348167   0.031567  11.029  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3694 on 881 degrees of freedom
## Multiple R-squared:  0.1253, Adjusted R-squared:  0.1233
## F-statistic: 63.12 on 2 and 881 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.8082 3.9758
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```
adf.DTB3 = ur.df(diff(TB3), type = "none", selectlags = "BIC")
summary(adf.DTB3)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3414 -0.0808  0.0097  0.1082  2.1945
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.78587    0.03787 -20.749  < 2e-16 ***
## z.diff.lag  0.19594    0.03304   5.931 4.33e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3639 on 881 degrees of freedom
## Multiple R-squared:  0.3543, Adjusted R-squared:  0.3529
## F-statistic: 241.7 on 2 and 881 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -20.7494
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
adf.AAA = ur.df(AAA, type = "drift", selectlags = "BIC")
summary(adf.AAA)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11444 -0.08028 -0.00553  0.07935  1.19397
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.028929   0.016274   1.778   0.0758 .
## z.lag.1     -0.004232   0.002343  -1.806   0.0712 .
## z.diff.lag   0.325805   0.031835  10.234   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1949 on 881 degrees of freedom
## Multiple R-squared:  0.1084, Adjusted R-squared:  0.1064
## F-statistic: 53.58 on 2 and 881 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -1.8062 1.6789
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

```
adf.DAAA = ur.df(diff(AAA), type = "none", selectlags = "BIC")
summary(adf.DAAA)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01082 -0.07914  0.00542  0.08295  1.14681
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.82609    0.03823 -21.606  < 2e-16 ***
## z.diff.lag  0.22250    0.03289   6.765 2.43e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1904 on 881 degrees of freedom
## Multiple R-squared:  0.3702, Adjusted R-squared:  0.3688
## F-statistic:   259 on 2 and 881 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -21.6057
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
kpss.TB3 = ur.kpss(TB3, type = "mu")
summary(kpss.TB3)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 2.9097
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
kpss.DTB3 = ur.kpss(diff(TB3), type = "mu")
summary(kpss.DTB3)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0689
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
kpss.AAA = ur.kpss(AAA, type = "mu")
summary(kpss.AAA)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 2.7442
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
kpss.DAAA = ur.kpss(diff(AAA), type = "mu")
summary(kpss.DAAA)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.2718
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Since TB3 and AAA are both I(1) they might be cointegrated. Check this possibility by performing cointegration tests.

There are several tests to consider:

- Engle-Granger (EG)
    - Basically an ADF-type test on the residuals from a simple linear regression of TB3 on AAA (or vice versa) but with different critical values.
    - EG tests do this and test under the null hypothesis of two or more series being not cointegrated.
    - Since the results can be sensitive to normalisation (i.e. choice of Y), best practice is to run the test twice.
- Interpreting EG test output:
    - Each printout has three panels, corresponding to the three possibl especifications of the ADF test regressions: Model 1, Model 2 and Model 3. We are using model 1 in this instance.
    - Comparing the models to eachother you can see the normalistaion does matter as the t-value vary significantly between printouts - which also leads to different p-values. However, in both cases the null hypothesis of no cointegration can be rejected at the 2% significance level, implying that they are CI(1,1).

```
# EG test

# library(aTSA)

coint.test(TB3,AAA)
```

```
## Response: TB3
## Input: AAA
## Number of inputs: 1
## Model: y ~ X + 1
## -------------------------------
## Engle-Granger Cointegration Test
## alternative: cointegrated
##
## Type 1: no trend
##     lag      EG p.value
##    6.00   -3.88    0.01
## -----
##  Type 2: linear trend
##     lag      EG p.value
##   6.000  -0.633   0.100
## -----
##  Type 3: quadratic trend
##     lag      EG p.value
##  6.0000  0.0773  0.1000
## -----------
## Note: p.value = 0.01 means p.value <= 0.01
##     : p.value = 0.10 means p.value >= 0.10
```

```
coint.test(AAA,TB3)
```

```
## Response: AAA
## Input: TB3
## Number of inputs: 1
## Model: y ~ X + 1
## -------------------------------
## Engle-Granger Cointegration Test
## alternative: cointegrated
##
## Type 1: no trend
##     lag      EG p.value
##   6.000  -3.337   0.017
## -----
##  Type 2: linear trend
##     lag      EG p.value
##     6.0     0.3     0.1
## -----
##  Type 3: quadratic trend
##     lag      EG p.value
##    6.00    1.14    0.10
## -----------
## Note: p.value = 0.01 means p.value <= 0.01
##     : p.value = 0.10 means p.value >= 0.10
```

- Johansen tests:
    - The tricky part of using Johansen tests is knowing which deterministic assumptions to apply: “none” (no deterministic term in EC but a constant outside EC), “const” (a constant in the EC, but no deterministic term outside the EC), or “trend” (a trend variables but no constant in EC and a constant outside EC).
    - Use the following approach for choosing the right term:
        - “none” is appropriate for (linearly) trending series, granted all trends are stochastic.
        - “const” is proper only if none of the variables appear to have a sustained tendency to increase or decrease.
        - “trend” is reasomable when there is some long-run linear growth that the cointegrating relation does not account for.

Because neither TB3 and AAA are trending linearly, but seem to have some broken trend, “const” is a reasonable option.

For lag length in this model, we use the VARselect() function. SC selects lag length 3, whereas all other measures select lag length 10.

To determine what’s correct, estimate VAR(3) and perform the BG test for autocorrelation of orders 1-3. The p-value of this test is large (0.18), therefore we can maintain the null hypothesis of no autocorrelation of orders 1-3. Therefore, we use K = 3.

```
# J tests

# library(urca)

data = cbind(AAA, TB3)

#library(vars)
VARselect(data, type = "const")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10     10      3     10
##
## $criteria
##                 1            2            3            4            5
## AIC(n) -5.33838552 -5.532230868 -5.610549434 -5.610157777 -5.606011398
## HQ(n)  -5.32587465 -5.511379421 -5.581357408 -5.572625171 -5.560138214
## SC(n)  -5.30567753 -5.477717557 -5.534230798 -5.512033816 -5.486082113
## FPE(n)  0.00480362  0.003957152  0.003659061  0.003660497  0.003675711
##                  6            7            8            9          10
## AIC(n) -5.601631527 -5.658245229 -5.652268434 -5.670675971 -5.68893683
## HQ(n)  -5.547417763 -5.595690886 -5.581373512 -5.591440471 -5.60136075
## SC(n)  -5.459896917 -5.494705294 -5.466923174 -5.463525388 -5.45998092
## FPE(n)  0.003691852  0.003488657  0.003509581  0.003445582  0.00338325
```

```
var3 = VAR(data, p = 3, type = "const")
serial.test(var3, lags.bg = 3, type = "BG")
```

```
##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object var3
## Chi-squared = 16.225, df = 12, p-value = 0.1811
```

First we look at the “trace” J-test. There are four parts to the printout.

- The first shows three estimated eigenvalues in decreasing order.
- The second part shows the trade statistics for the null hypotheses of r_0 <= 1 and r_0 = 0. These need to be evaluated in reverse order, starting with r_0 = 0.
    - The r_0 = 0 test statistics is very large (38.27), much larger than the 1% critical value (24.60)
    - However, the r_0 <= 1 test statistic is smaller than the correponding 10% value (7.52)
    - Therefore, at the 1%, 5% and 10% critical values alike r_0 = 0 is rejects but r_0 <= is maintained.
    - This implies r = 1.
- The third part shows the estimated corintegration relations that correspond to the three eigenvalues in the first table.
    - Since we concluded r = 1, we consider only the first eigenvalue and cointegrating relation.

    - Therefore, the estimated EC term, normalised with respect to the first endogenous variable is: AAA - 2.2585 - 1.0265TB3 = Epsilon
- The fourth part of the printout shows the estimated speed of adjustment coefficients. Again, only the first column is relevant (i.e. alpha = [-0.0191, 0.0107])

```
data = cbind(AAA, TB3)
j.trace = ca.jo(data, type = "trace", K = 3, ecdet = "const")
summary(j.trace)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1]  3.781984e-02  4.780644e-03 -2.930396e-19
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  4.23  7.52  9.24 12.97
## r = 0  | 38.27 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##             AAA.l3     TB3.l3   constant
## AAA.l3    1.000000  1.0000000  1.0000000
## TB3.l3   -1.026532  0.0557922 -0.2276885
## constant -2.258470 -7.0850332  7.3511208
##
## Weights W:
## (This is the loading matrix)
##
##            AAA.l3       TB3.l3      constant
## AAA.d -0.01913395 -0.002857251  7.630392e-19
## TB3.d  0.01069988 -0.008111077 -7.637752e-19
```

Now we conduct the same process but using the maximum eigenvalue test.

- Only the second part of the new prinout differs from the previous, the rest are the same.
- However, the statistical conclusions do not change: we reejct the first null hypothesis but maintain the second, implying r = 1.

```
data = cbind(AAA, TB3)
j.trace = ca.jo(data, type = "eigen", K = 3, ecdet = "const")
summary(j.trace)
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , without linear trend and constant in c
ointegration
##
## Eigenvalues (lambda):
## [1]  3.781984e-02  4.780644e-03 -2.930396e-19
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  4.23  7.52  9.24 12.97
## r = 0  | 34.04 13.75 15.67 20.20
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##             AAA.l3     TB3.l3   constant
## AAA.l3    1.000000  1.0000000  1.0000000
## TB3.l3   -1.026532  0.0557922 -0.2276885
## constant -2.258470 -7.0850332  7.3511208
##
## Weights W:
## (This is the loading matrix)
##
##            AAA.l3       TB3.l3      constant
## AAA.d -0.01913395 -0.002857251  7.630392e-19
## TB3.d  0.01069988 -0.008111077 -7.637752e-19
```

d) Estimate a VECM

Like testing for cointegration, a VECM can be estimated based on the EG or J approaches.

In the former, we have to estimate the long-run equilibrium relationship between the two variables, then estimated the VECM eqns one-by-one with the lagged residuals for the equilibrium used as the EC term.

- Step 1: estimate the long-run equilibrium between the two variables.
    - This prinout show the LR equilibrium relationship is: AAA = 3.1347 + 0.7962(TB3).
    - There is a significantly positive relationship between AAA and TB3.
- Step 2: save the residuals from this regression
- Step 3: take the first difference of the dependent variable (AAA)
- Step 4: take the lag of the residuals from LR equilibrium eqn
- Step 5: regress the former (first difference of the dependent variable) on the latter (the lag of the residuals from the LR equilibrium eqn)

Then you have the printout of the first VECM equation: DAAA = 0.0030 - 0,0263(ec_t-1).

Then you need to derive the second equation of the VECM…

```
# Step 1

EC.1 <- lm(AAA ~ TB3)
summary(EC.1)
```

```
##
## Call:
## lm(formula = AAA ~ TB3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6549 -1.2472 -0.0918  1.2109  3.6640
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.13472    0.07539   41.58   <2e-16 ***
## TB3          0.79623    0.01485   53.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.359 on 884 degrees of freedom
## Multiple R-squared:  0.7649, Adjusted R-squared:  0.7646
## F-statistic:  2876 on 1 and 884 DF,  p-value: < 2.2e-16
```

```
# Step 2

e.1 = ts(EC.1$residuals, start = c(1950,1), end = c (2023,9), frequency = 12)

# Step 3

DAAA = window(diff(AAA), start = c(1950,2), end = c (2023,9),frequency = 12)

# Step 4

le.1 = window(stats::lag(e.1, -1), start = c(1950,2), end = c (2023,9), frequency = 12)

# Step 5

ec.11 <- lm(DAAA ~ le.1)
summary(ec.11)
```

```
##
## Call:
## lm(formula = DAAA ~ le.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16592 -0.08071 -0.01575  0.07993  1.23484
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.003034   0.006834   0.444    0.657
## le.1        -0.026308   0.005047  -5.213 2.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2032 on 882 degrees of freedom
## Multiple R-squared:  0.02989,    Adjusted R-squared:  0.02879
## F-statistic: 27.17 on 1 and 882 DF,  p-value: 2.319e-07
```

To derive the second equation of the VECM:

- You need to regress the relevant series on the lagged residuals from the LR equation.

```
DTB3 = window(diff(TB3), start = c(1950,2), end = c (2023,9),
frequency = 12)
ec.12 = lm(DTB3 ~ le.1)
summary(ec.12)
```

```
##
## Call:
## lm(formula = DTB3 ~ le.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6153 -0.0866  0.0029  0.1135  2.6071
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.004786   0.013276   0.360    0.719
## le.1        0.004173   0.009805   0.426    0.671
##
## Residual standard error: 0.3947 on 882 degrees of freedom
## Multiple R-squared:  0.0002053,  Adjusted R-squared:  -0.0009282
## F-statistic: 0.1811 on 1 and 882 DF,  p-value: 0.6705
```

Reflections on this model:

- The first regression is strongly significant although R_squared is very low.
- The second regression is insignificant, as its its estimate of the speed of adjustment coefficient. This means the system of AAA and TB3 adjusts to deviations from the LR equilibrium through changes in AAA and the development of TB3 are not directly linked to the equilibrium error.

- We did not check if these VECM equations are free of autocorrelation, and indeed they are not. However, if you augmented these equations with the first lags of the first differences of AAA and TB3, it is possible to eliminate autocorrelation.
- The results of the EG two-step procedure can be sensitive to normalisation. Therefore, in practice it would be important to re-estimate the LR equilibrium between the two variables normalised to TB3, using the residuals from the new equilibrium regresison and estimate the two equations of the VECM. Do this to prep for the exam.

Now we move onto the Johansen procedure, starting with the trace method.

- comparing this to the previous J-trace printout, you can see that:
    - i. the coefficients of the EC term (ect1) arethe same estimated speed of adjustment coefficients term as in the fourth part of the J trade test printout.
    - ii. the $beta vector is the estimated cointegration realtion and it is the same as teh first column vector in the third part of teh J-trace test printout.
- From this printout we get all the terms for the VECM(2):
    - DAAA(hat) = -0.0191(ect_t-1) + 0.3407(DAAA_t-1) - 0.2765(DAAA_t-2) + 0.0407(DTB3) + 0.0298(DTB3_t-2)
    - DTB3(hat) = 0.0107(ect_t-1) + 0.2963(DAAA_t-1) - 0.2491(DAAA_t-2) + 0.3405(DTB3_t-1) - 0.1355(DTB3_t-2)
- The EC term is:
    - ect = AAA - 2.2585 - 1.0265(TB3_t)

```
vecm.j = cajorls(j.trace, r = 1)
print(vecm.j)
```

```
## $rlm
##
## Call:
## lm(formula = substitute(form1), data = data.mat)
##
## Coefficients:
##          AAA.d     TB3.d
## ect1     -0.01913   0.01070
## AAA.dl1   0.34070   0.29626
## TB3.dl1   0.04067   0.34051
## AAA.dl2  -0.27653  -0.24907
## TB3.dl2   0.02975  -0.13554
##
##
## $beta
##                ect1
## AAA.l3    1.000000
## TB3.l3   -1.026532
## constant -2.258470
```

e) Obtain the VAR(3) representaion of the estimated VECM(2) model.

Any VECM can be transformed into an equivalent VAR in levels with the vec2var() function.

From this, we get the following VAR(3) model:

- AAA(hat) = 0.0432 + 1.3407(AAA_t-1) - 0.6172(AAA_t-2) + 0.2574(AAA_t-3) + 0.0407(TB3_t-1) - 0.0109(TB3_t-2) - 0.0101(TB3_t-3)
- TB3(hat) = 0.0242 + 0.2963(AAA_t) - 0.5453(AAA_t-2) + 0.2598(AAA_t-3) + 1.3405(TB3_t-1) - 0.4761(TB3_t-2) + 0.1246(TB3_t-3)

```
var.j = vec2var(j.trace, r = 1)
print(var.j)
```

```
##
## Coefficient matrix of lagged endogenous variables:
##
## A1:
##        AAA.l1     TB3.l1
## AAA 1.3406951 0.04066789
## TB3 0.2962563 1.34051385
##
##
## A2:
##         AAA.l2      TB3.l2
## AAA -0.6172266 -0.01092015
## TB3 -0.5453254 -0.47605324
##
##
## A3:
##        AAA.l3      TB3.l3
## AAA 0.2573975 -0.01010613
## TB3 0.2597690  0.12455562
##
##
## Coefficient matrix of deterministic regressor(s).
##
##        constant
## AAA  0.04321345
## TB3 -0.02416536
```