

FA_Assignment5_Duo_Zhou

Duo Zhou

7/21/2020

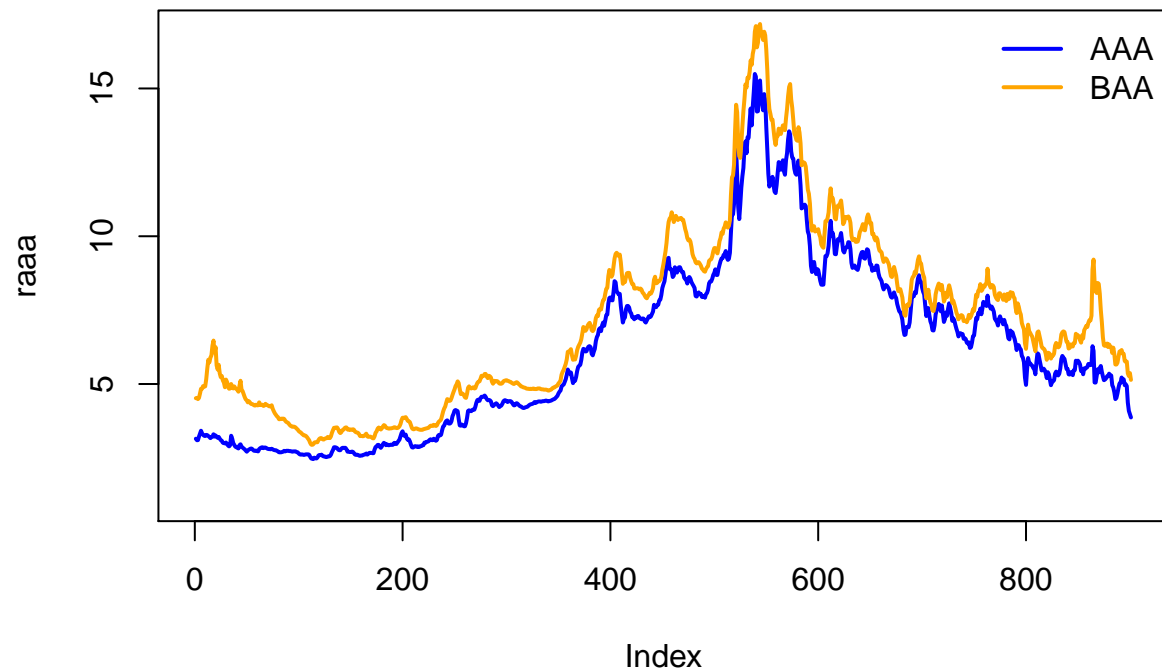
This assignment helps understanding stationarity and seasonality of linear models for time series. Consider the monthly yields of Moody's AAA and BAA bonds from exercises 4-6 on page 126. The data are in the file MYieldsData.csv. Analyze possible types of relationships between the two yield variables using regression model with stationary residuals and cointegration. What is a valid model for predicting the data?

Only the last 900 observations are chosen for both models. According to Yuri, the johansen cointegration function in R does not work well with more than 900 observations. The regression model also uses the same observations for comparison purposes.

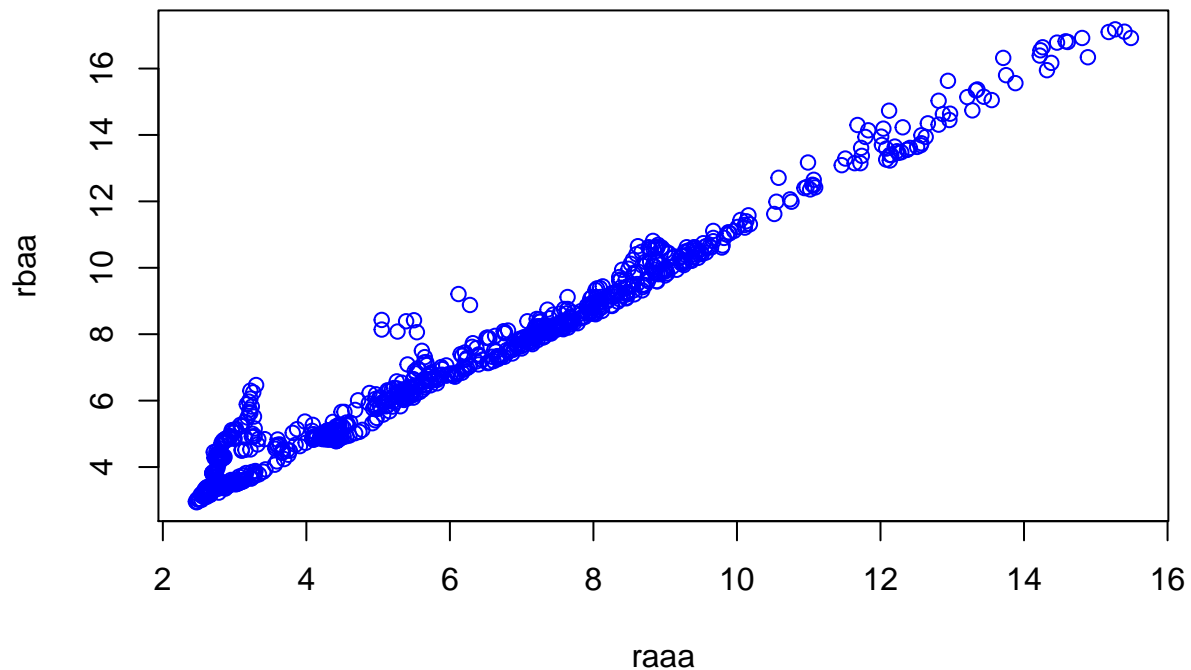
```
# Loading Data
datapath <- "C:/Users/zd000/Desktop/MSCA/Financial Analytics/Assignments/week5/"
x<-read.csv(file=paste(datapath,"MYieldsData.csv",sep="/"))
n <- nrow(x)
nb <-max(n-900,1)
raaa <- x[nb:n,2]
rbaa <- x[nb:n,3]
```

```
plot(raaa,col ="blue",type="l",lwd=2,ylim=c(1,17),main = "The monthly yields of Moody's AAA and BAA Bonds")
lines(rbaa,col="orange",lwd=2)
legend("topright", c("AAA","BAA"), lwd=c(2,2),col = c("blue","orange"), bty="n")
```

The monthly yields of Moody's AAA and BAA Bonds



```
#Scatter Plot of raaa and rbaa  
raaa<-as.vector(raaa)  
rbaa<-as.vector(rbaa)  
plot(raaa,rbaa,col= "blue")
```



Regression Model

Fit to a linear model and print the summary

```
linreg <- lm(rbaa~raab)
```

```
summary(linreg)
```

```
##
```

```
## Call:
```

```
## lm(formula = rbaa ~ raab)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.6191 -0.2955 -0.1653  0.1449  2.4247
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.692687   0.037303   18.57  <2e-16 ***
## raab         1.051992   0.005467  192.42  <2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

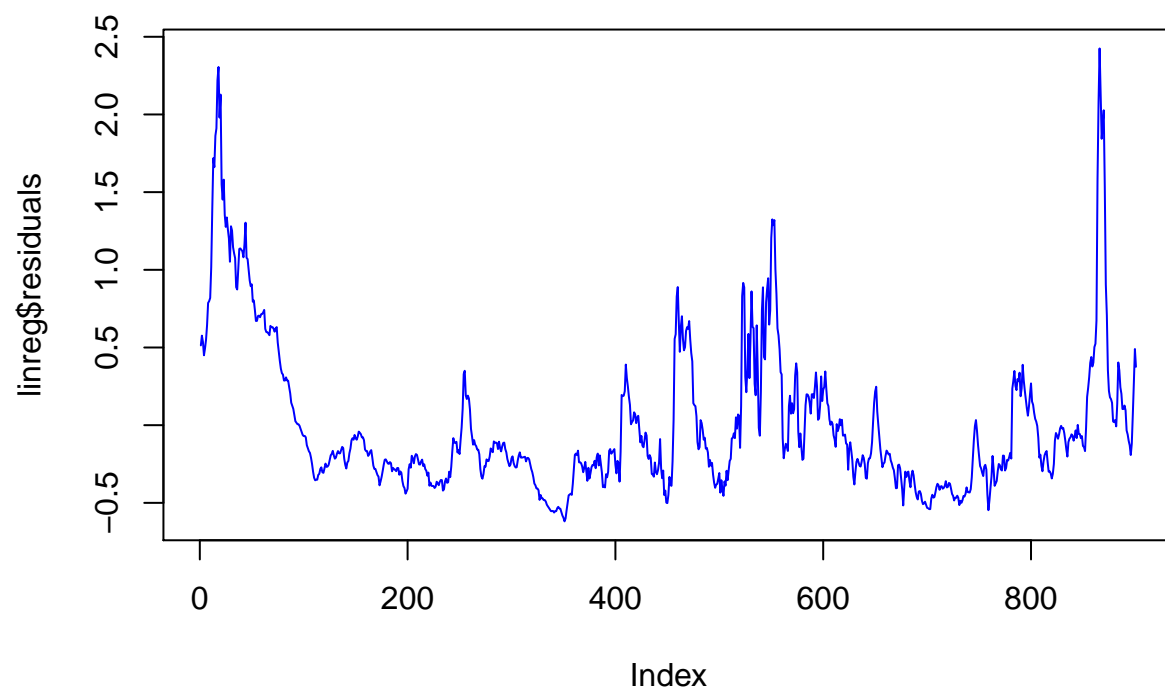
```
##
```

```
## Residual standard error: 0.4832 on 899 degrees of freedom
```

```
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9763
```

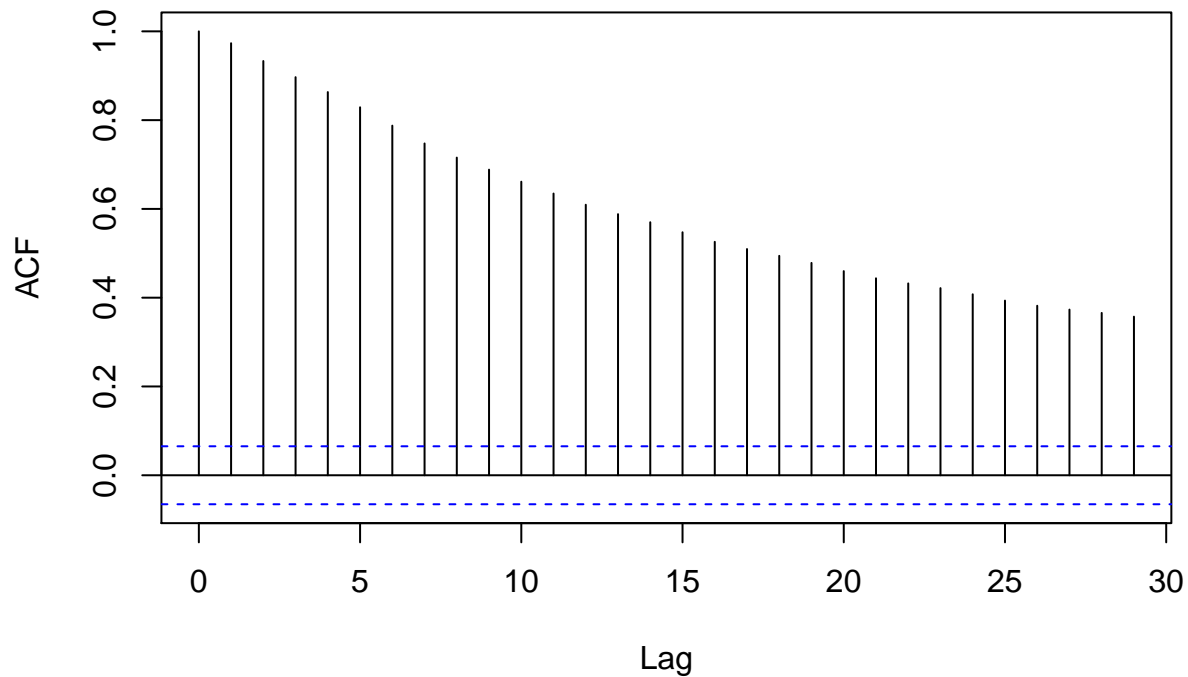
```
## F-statistic: 3.703e+04 on 1 and 899 DF, p-value: < 2.2e-16
```

```
# Explore the residuals  
plot(linreg$residuals,type="l",col ="blue")
```



```
acf(linreg$residuals)
```

Series linreg\$residuals



```
Box.test(linreg$residuals, lag=12, type="Ljung")
```

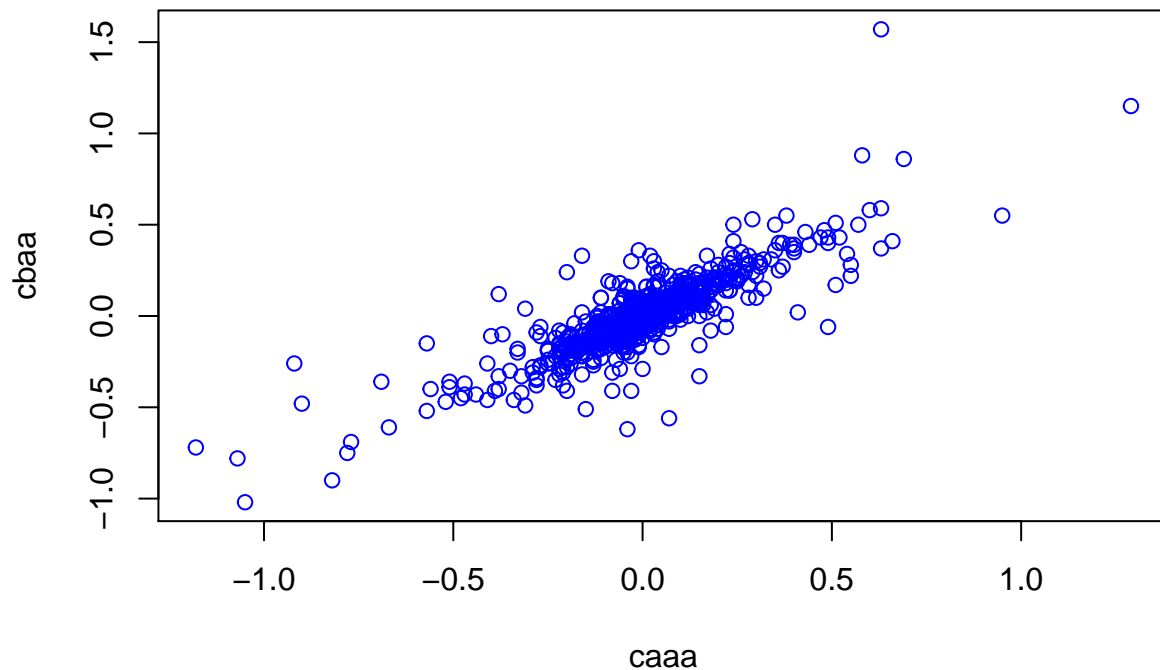
```
##  
## Box-Ljung test  
##  
## data: linreg$residuals  
## X-squared = 6751.7, df = 12, p-value < 2.2e-16
```

The sample ACF of residuals is highly significant and decays slowly, showing that the process of the residuals has long memory and is not stationary. There is a pattern of a unit-root nonstationary time series, in other words two bond yields are not cointegrated.

This behavior of residuals leads to the consideration of differencing the series of interest rates.

```
# Take the first difference of both time series  
caaa<-diff(raaa)  
cbaa<-diff(rbaa)  
plot(caaa,cbaa,main="Scatter plot of Differences", col = "blue")
```

Scatter plot of Differences



The figure shows that the differenced series remain highly correlated.

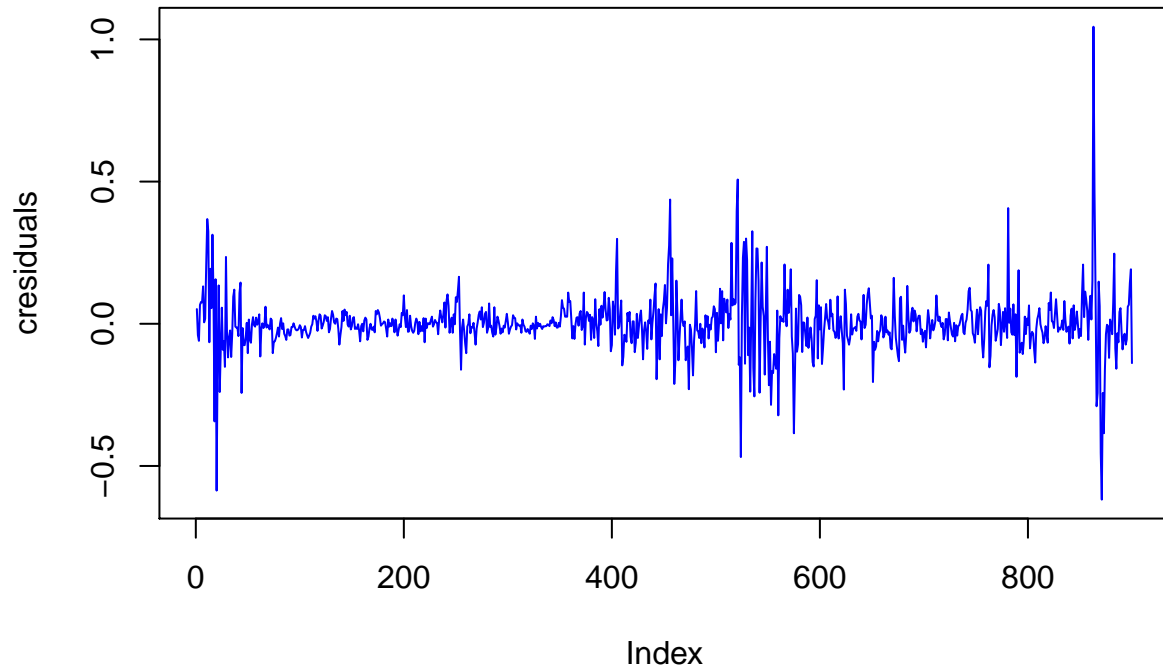
Fit linear regression to the differenced time series with no intercept.

```
clinreg <-lm(cbaa~caaa-1)
summary(clinreg)
```

```
##
## Call:
## lm(formula = cbaa ~ caaa - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61839 -0.03665 -0.00336  0.03163  1.04453
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## caaa  0.83408    0.01746   47.77  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1012 on 899 degrees of freedom
## Multiple R-squared:  0.7174, Adjusted R-squared:  0.7171
## F-statistic: 2282 on 1 and 899 DF, p-value: < 2.2e-16
```

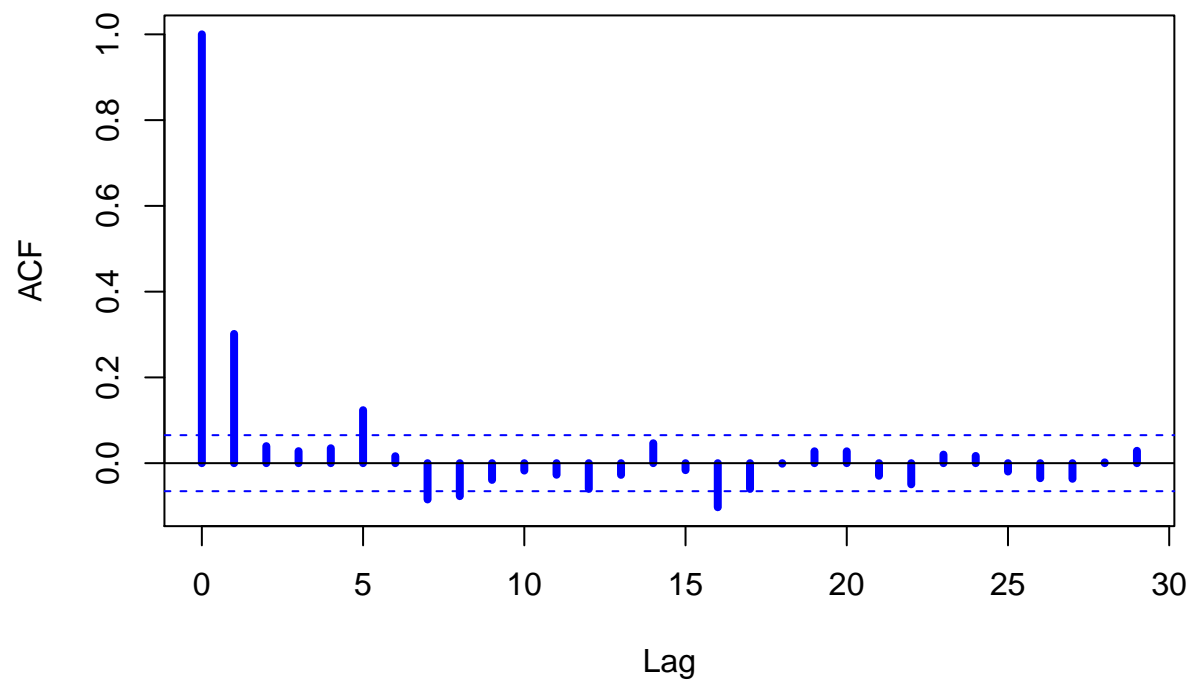
Test the residuals of clinreg

```
cresiduals <- clinreg$residuals  
plot(cresiduals,type = "l",col = "blue")
```



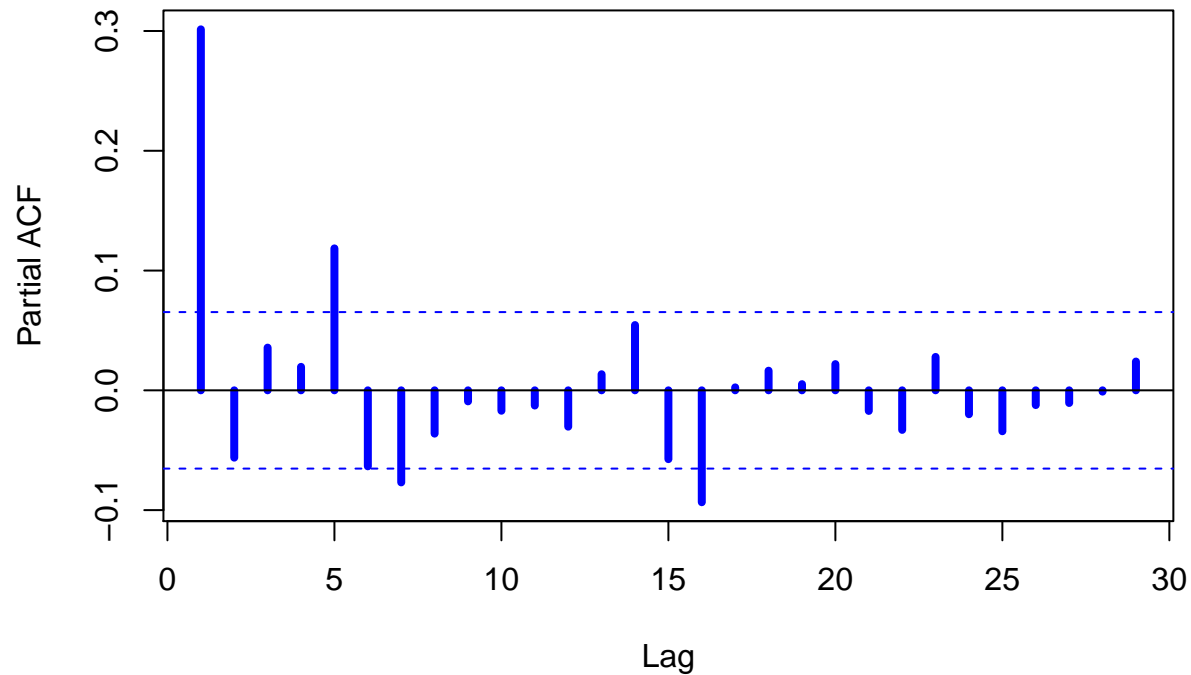
```
acf(cresiduals,main = "ACF of residuals",col ="blue",lty=1 ,lwd = 4)
```

ACF of residuals



```
pacf(cresiduals,main = "PACF of residuals",col ="blue",lty=1 ,lwd = 4)
```


PACF of residuals



```
Box.test(cresiduals,lag=10,type='Ljung')
```

```
##  
## Box-Ljung test  
##  
## data:  cresiduals  
## X-squared = 112.9, df = 10, p-value < 2.2e-16
```

The residuals from the fit to the differenced time series look more stationary. However, Box-Ljung test shows that serial correlation is still present among different lags. Both ACF and PACF plots show significant correlation at multiple lags.

Use `auto.arima` function to find the best arima model with lowest AIC/BIC.

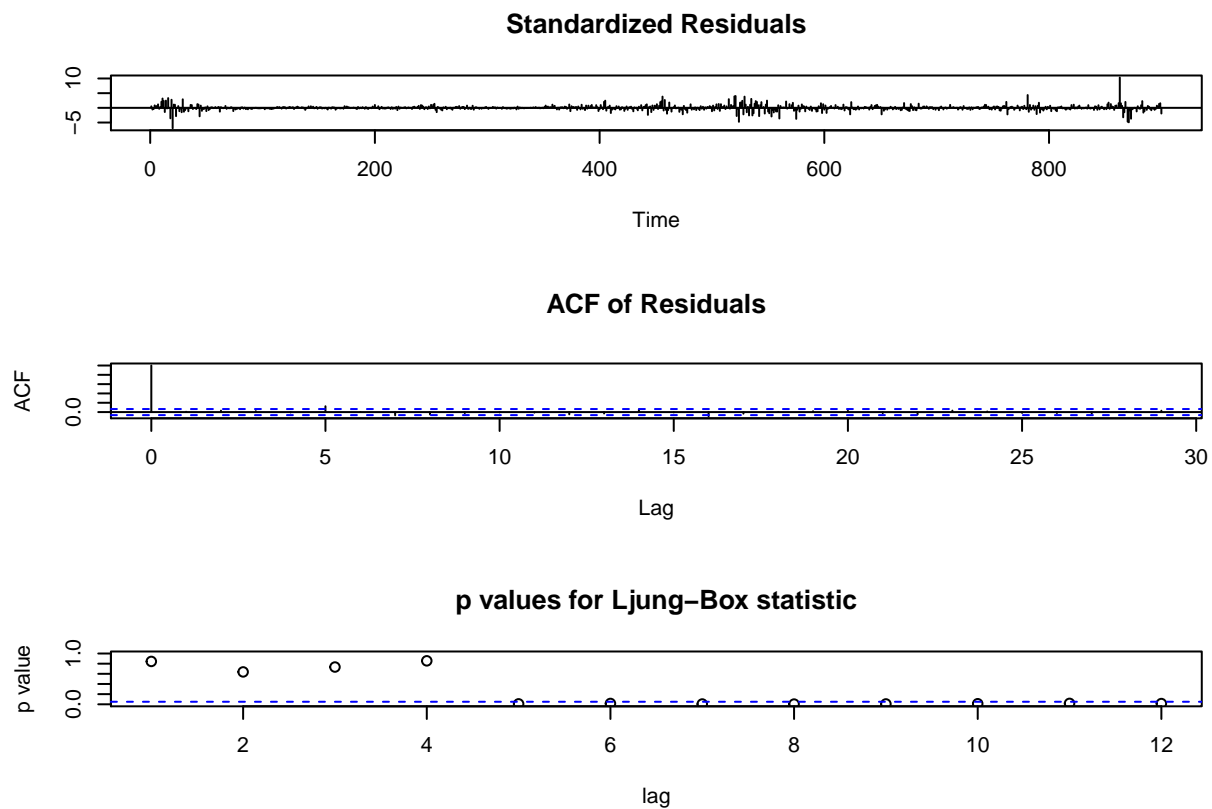
```
ma1<-auto.arima(cresiduals, seasonal=T)  
summary(ma1, which="all")
```

```
## Series:  cresiduals  
## ARIMA(0,0,1) with zero mean  
##  
## Coefficients:  
##          ma1  
##      0.3167  
## s.e.  0.0317  
##
```

```
## sigma^2 estimated as 0.009284: log likelihood=829.18
## AIC=-1654.35 AICc=-1654.34 BIC=-1644.75
##
## Training set error measures:
##           ME           RMSE          MAE MPE MAPE          MASE          ACF1
## Training set -3.455726e-05 0.09629829 0.05699362 NaN  Inf 0.7914074 0.006556291
```

Auto.arima function suggests MA(1) model

```
# Diagnostic of the residuals of MA(1) model
tsdiag(ma1,gof=12)
```



We can see that up to lag 4, the BOX-Ljung test concludes no rejection to Null hypothesis. The Null hypothesis is that there is no correlation between the residuals of the MA(1) model among the first 4 lags.

Finally, we estimated a model for two bond yields series:

$$x_{t,baa} = x_{t-1,baa} + \beta_2(x_{t,aaa} - x_{t-1,aaa}) + \epsilon_t, t = 2, \dots, T$$

$$\epsilon_t = a_t - \theta * a_{t-1}, t = 2, \dots, T$$

Build forecast for r3 rate according this model.

Find forecast for $\epsilon_t, t = 2, \dots, T$ using MA(1) model:

$$\epsilon_t = -\theta * a_t = -\theta * ResidualsMA(1)_{t-1}$$

Note that arma() uses MA coefficients with opposite signs as the book.

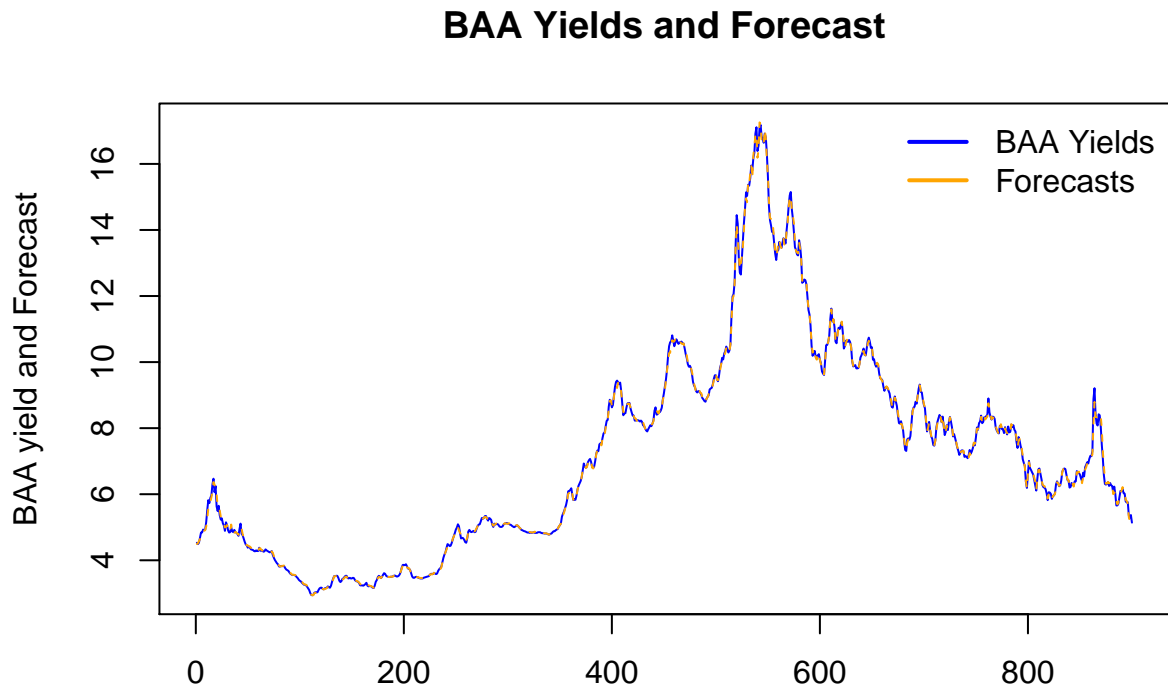
Forecast the BAA yields using the equation above and Plot the BAA yields and the forecast

```

theta1 <-ma1$coef[1]
a_t <- theta1 *residuals(ma1)

x_1<-rbaa[-length(rbaa)]
forec <- x_1 + clinreg$coefficients*caaa+a_t
matplot(cbind(rbaa[-1],forec),type = "l",col = c("blue","orange"),main= "BAA Yields and Forecast",ylab=
legend("topright", c("BAA Yields","Forecasts"), lwd=2,col = c("blue","orange"), bty="n")

```



Check the scatter plot of BAA Yields forecast differences versus differences of AAA Yields.

```

difforec <- diff(forec)
cr<-cbind(difforec,caaa[-1])
plot (cr[,1],cr[,2],col = "blue",main = "Differences of Forecasted BAA Yields vs Differences of AAA Yields",
xlab="Difference of Forecasts of BAA Yields",ylab="Differences of AAA Yields")

```

Differences of Forecasted BAA Yields vs Differences of AAA Yields

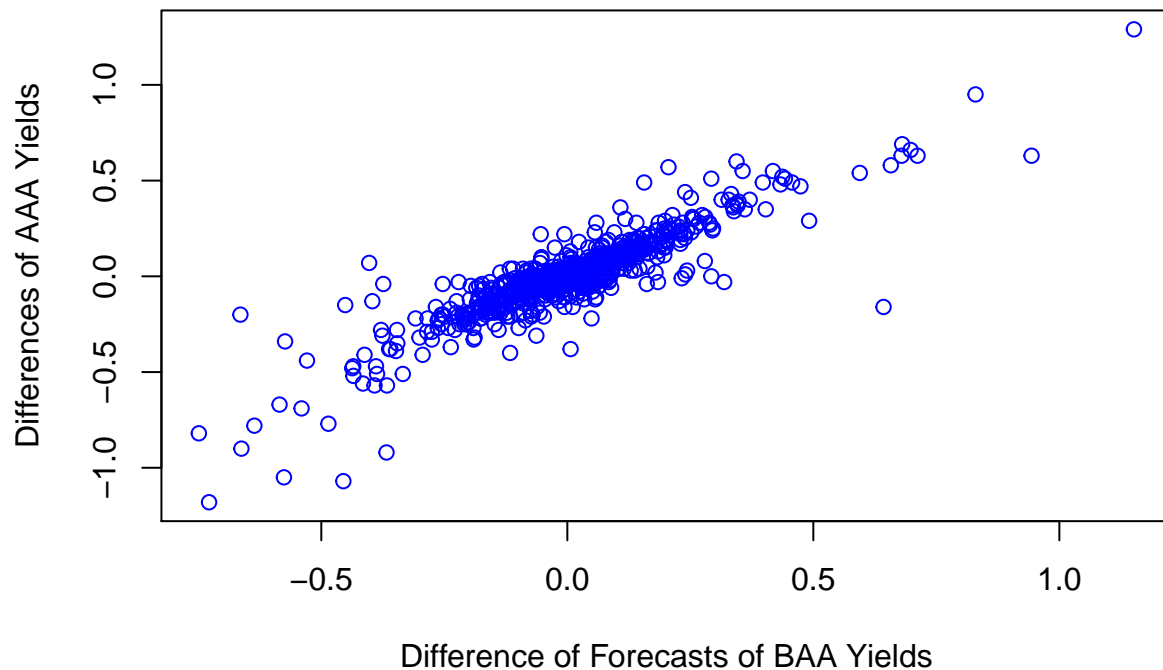


Figure shows that regression model with ARIMA residuals preserved the “short term” dependence of yields increments.

Cointegration

Fit cointegration model

```
data <- cbind(raaa,rbaa)
cajo <- ca.jo(data, ecdet = "none", type="eigen", K=2, spec="longrun")
summary(cajo)

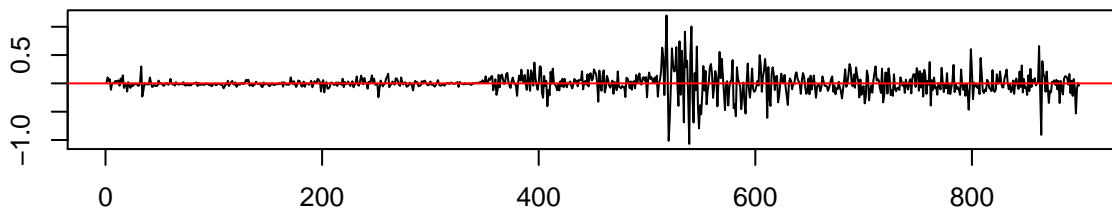
##
## #####
## # Johansen-Procedure #
## #####
##
## Test type: maximal eigenvalue statistic (lambda max) , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.021863898 0.002207871
##
## Values of teststatistic and critical values of test:
##
##          test 10pct  5pct  1pct
## r <= 1 |   1.99   6.50   8.18 11.65
```

```
## r = 0 | 19.87 12.91 14.90 19.19
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##          raaa.l2  rbaa.l2
## raaa.l2  1.0000000 1.000000
## rbaa.l2 -0.9343809 2.267935
##
## Weights W:
## (This is the loading matrix)
##
##          raaa.l2          rbaa.l2
## raaa.d 0.004425655 -0.0008455733
## rbaa.d 0.034114530 -0.0006455301
```

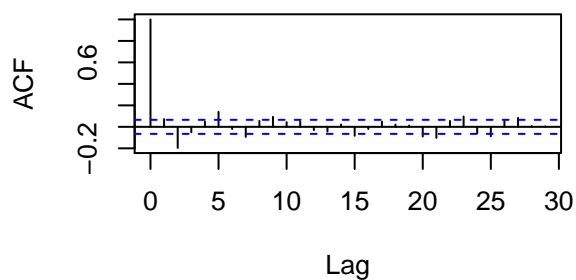
Residuals and their ACF's and PACF's for aaa and baa yields, respectively

```
plotres(cajo)
```

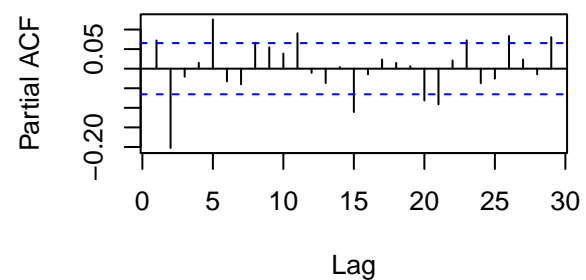
Residuals of 1. VAR regression



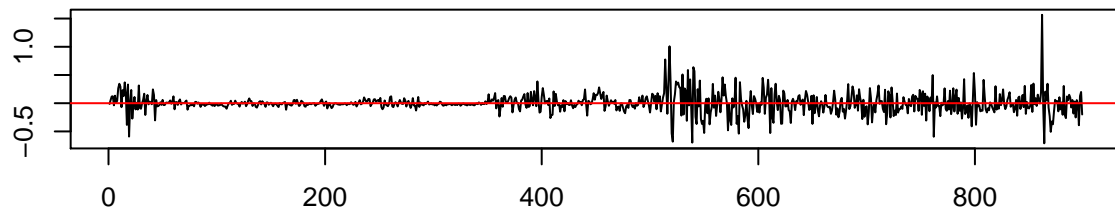
Autocorrelations of Residuals



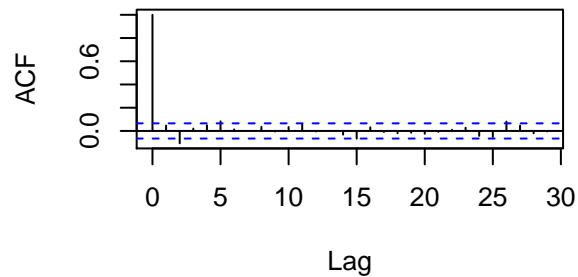
Partial Autocorrelations of Residuals



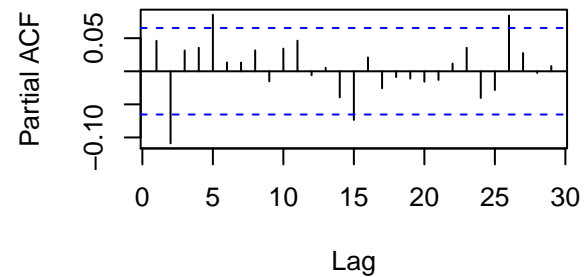
Residuals of 2. VAR regression



Autocorrelations of Residuals

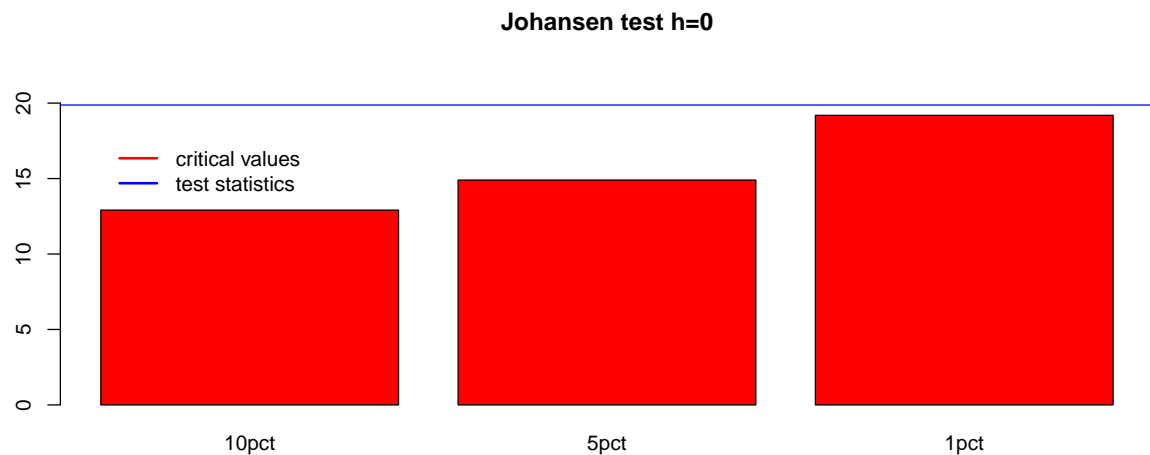
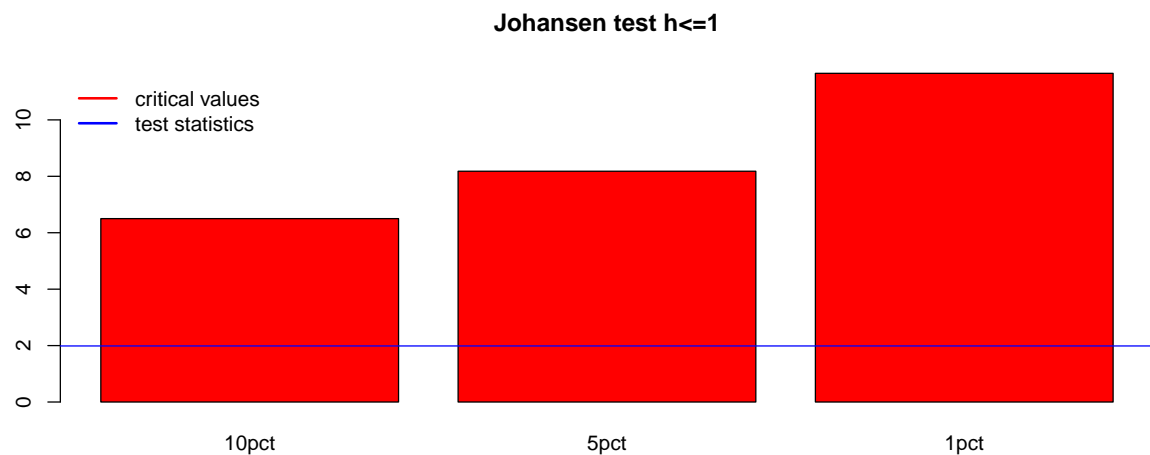


Partial Autocorrelations of Residuals



Check statistics and critical values of the test for cointegration order

```
par(mfcol=c(2,1))
barplot(cajo@cval[1,],main = "Johansen test h<=1",col = "red")
abline(h=cajo@teststat[1], col="blue")
legend("topleft", c("critical values","test statistics"), lwd=2,col = c("red","blue"), bty="n")
barplot(cajo@cval[2,],main = "Johansen test h=0",col = "red",ylim=c(0,22))
abline(h=cajo@teststat[2], col="blue")
legend(0.2,18, c("critical values","test statistics"), lwd=2,col = c("red","blue"), bty="n")
```



The first chart shows the critical values and test statistic for $H_0: r \leq 1$, where r is the order of cointegration. For all levels of 10%, 5%, 1% the statistic is below the critical values. So, $H_0: r \leq 1$ cannot be rejected.

The second chart shows the same variables, but for $H_0: r = 0$. For this null hypothesis the test statistic is above the critical values for 10%, 5% and 1%. So, with levels of 1% or more $H_0: r = 0$ is rejected.

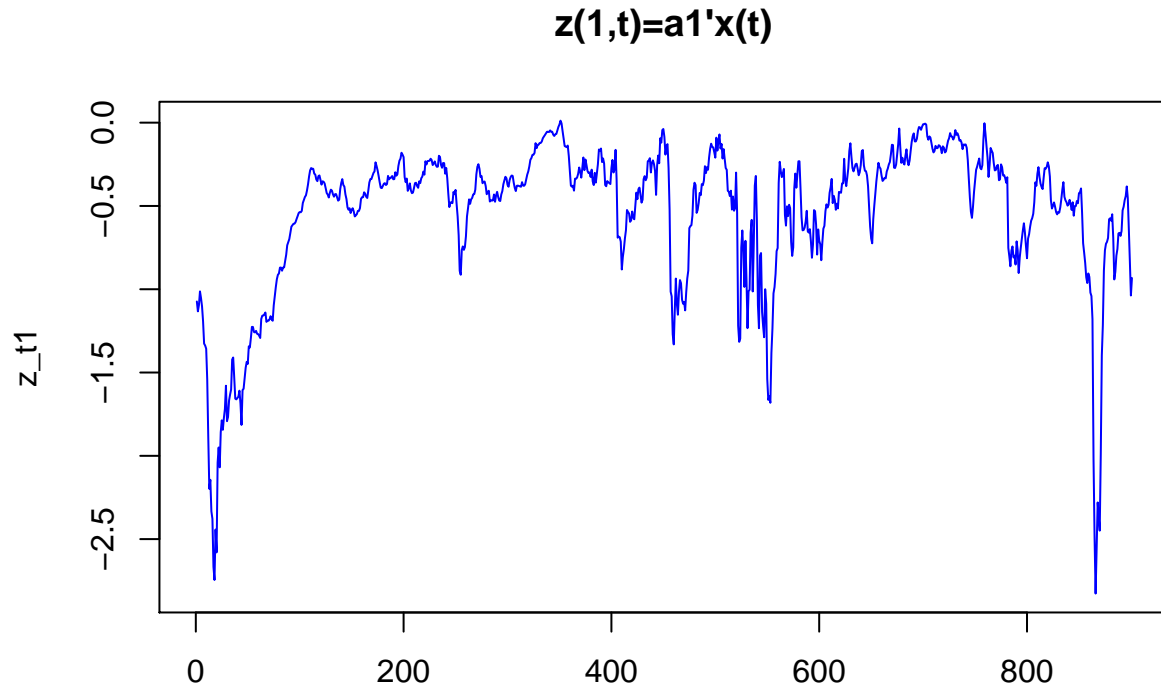
Conclusion: the cointegrating order equals 1.

By definition of cointegration with order $r=1$ process $z_{t,1} = \alpha^T \{1\} * x_{t,1}$ must be stationary ($I(0)$), where α is the cointegration vector.

```
# cointegration vector
(a_1 <- cajo@V[,1])
```

```
##      raaa.12      rbaa.12
## 1.0000000 -0.9343809
```

```
z_t1= data %*% a_1
matplot(z_t1,type = "l", main = "z(1,t)=a1'x(t)", col = "blue")
```



Perform Augmented Dickey-Fuller Test

```
#Estimate autoregression model for process zt1 and find the order chosen
zar <-ar(z_t1, aic = TRUE,method = "yule-walker")
zar$order # k=18 is chosen
```

```
## [1] 18
```

```
adf.test(z_t1,k=18,alternative="stationary")
```

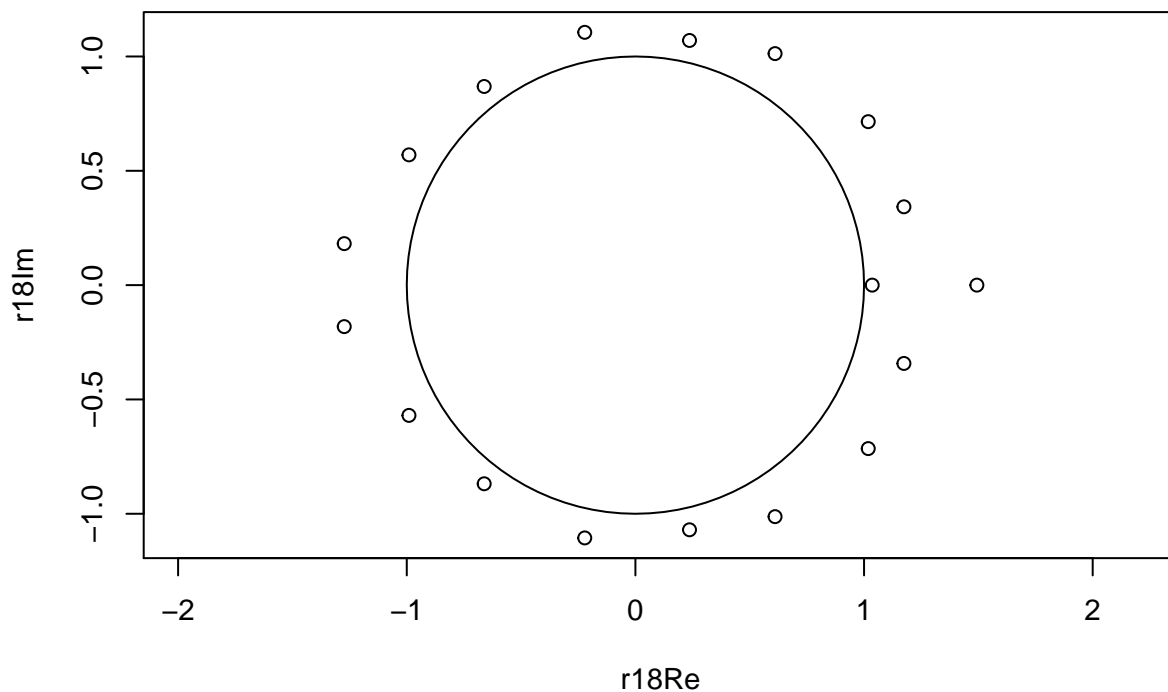
```
##
## Augmented Dickey-Fuller Test
##
## data: z_t1
## Dickey-Fuller = -4.1623, Lag order = 18, p-value = 0.01
## alternative hypothesis: stationary
```

```
par(mfrow = c(1, 1), cex = 0.9)
polyPar<-c(1,-zar$ar)
r18<-polyroot(polyPar)
Mod(r18)
```



```
## [1] 1.182574 1.091905 1.091905 1.035686 1.095930 1.142542 1.127636 1.244236
## [9] 1.127636 1.142542 1.095930 1.223316 1.244236 1.286316 1.182574 1.223316
## [17] 1.286316 1.493480
```

```
r18Re<-Re(r18)
r18Im<-Im(r18)
plot(r18Re,r18Im,xlim=c(min(r18Re),max(r18Re)),asp=1,ylim=c(min(r18Im),max(r18Im)))
draw.circle(0,0,radius=1)
```



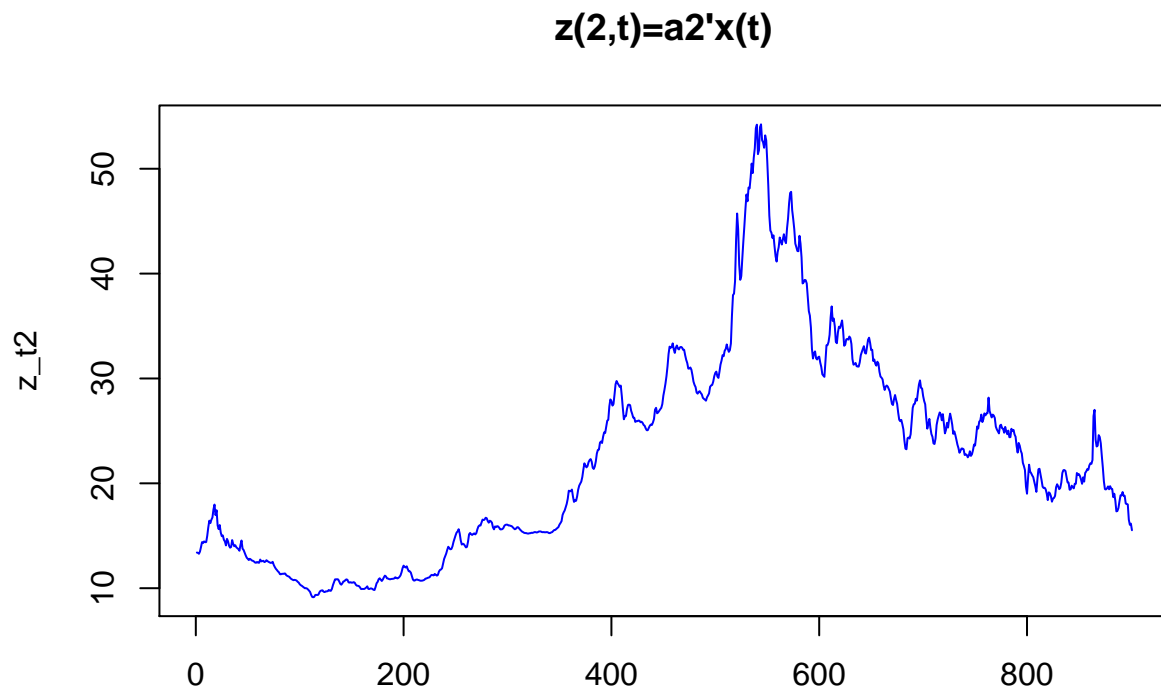
The ADF test gives us a significant p-value and we can reject the null hypothesis that there is at least one unit root.

The plot of the characteristic equation roots shows that all the roots are larger than the unit circle.

The process is stationary

Since cointegration order equals 1, vector a_2 is not a cointegration vector and the process $z_{t,2} = a_2^T * x_t$ should not be stationary. Let's verify.

```
a_2<- cajo@V[,2]
z_t2= data %*% a_2
matplot(z_t2,type ="l", main = "z(2,t)=a2'x(t)", col = "blue")
```



The plot does not look stationary at all.

ADF test for z_t2

```
zar2 <-ar(z_t2, aic = TRUE,method = "yule-walker")
zar2$order # k=3 is chosen
```

```
## [1] 3
```

```
adf.test(z_t2,k=3,alternative="stationary")
```

```
##
## Augmented Dickey-Fuller Test
##
## data: z_t2
## Dickey-Fuller = -0.74234, Lag order = 3, p-value = 0.9668
## alternative hypothesis: stationary
```

The ADF test gives us a very insignificant p-value and we cannot reject the null hypothesis that there is at least one unit root. ADF test verifies that z_t2 is not stationary and a_2 is not a cointegration vector.

Prediction using Cointegration model

$$\Delta X_t = \Gamma * \Delta X_{t-1} + \Pi_1 * X_{t-2} + \mu + \epsilon_t$$

```
(mu <-cajo@GAMMA[,1])
```

```
##      raaa.d      rbaa.d  
## 0.02194379 0.03322686
```

```
(PI<-cajo@PI)
```

```
##      raaa.l2      rbaa.l2  
## raaa.d 0.003580081 -0.006052953  
## rbaa.d 0.033469000 -0.033339986
```

```
(Gamma<-cajo@GAMMA[,2:3])
```

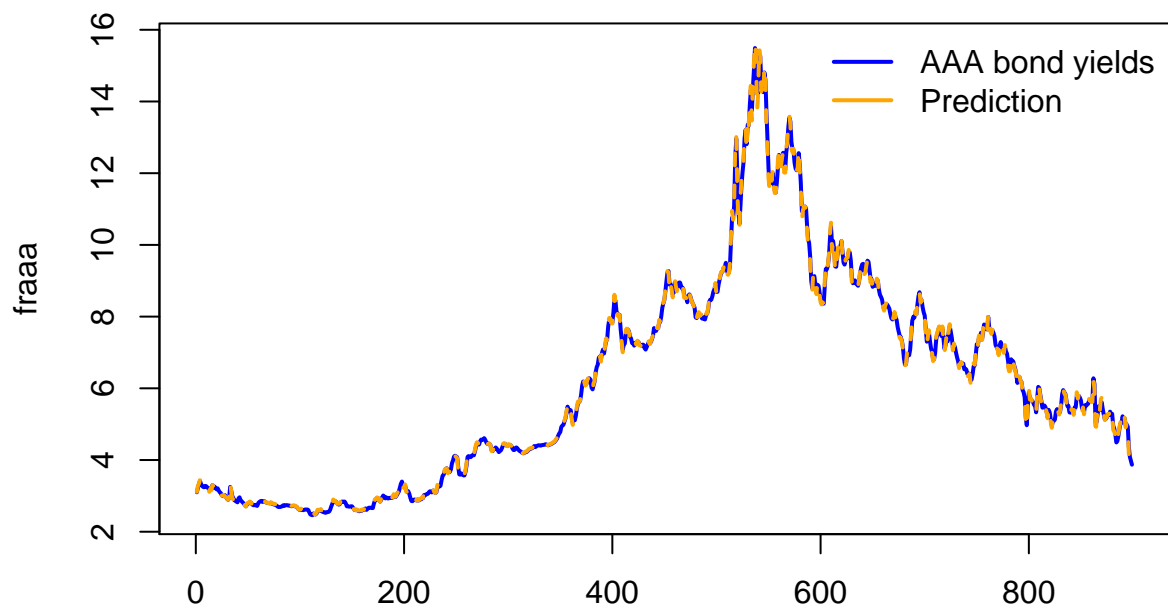
```
##      raaa.dl1      rbaa.dl1  
## raaa.d 0.5089958 -0.2098921  
## rbaa.d 0.3220645 0.1054791
```

```
dX_1 <- cajo@Z0  
X_2 <- cajo@ZK  
deltaX_t_1 <- Gamma %*% t(dX_1) + PI %*%t(X_2)  
deltaX_t_1<-apply(deltaX_t_1,2,"+",mu)  
nrowsdata <- dim(data)[1]  
data_t_2 = data[3:nrowsdata,]  
deltaX_t_1 <- t(deltaX_t_1)  
forecX <- data_t_2+deltaX_t_1
```

Plot predictions of the aaa bond yields

```
fraaa = cbind(raaa[3:length(raaa)],forecX[,1])  
matplot(fraaa,col =c("blue","orange"),type="l",lwd=c(2,2),  
        main = "AAA bond yields and prediction")  
legend("topright", c("AAA bond yields","Prediction"), lwd=c(2,2),  
       col = c("blue","orange"), bty="n")
```

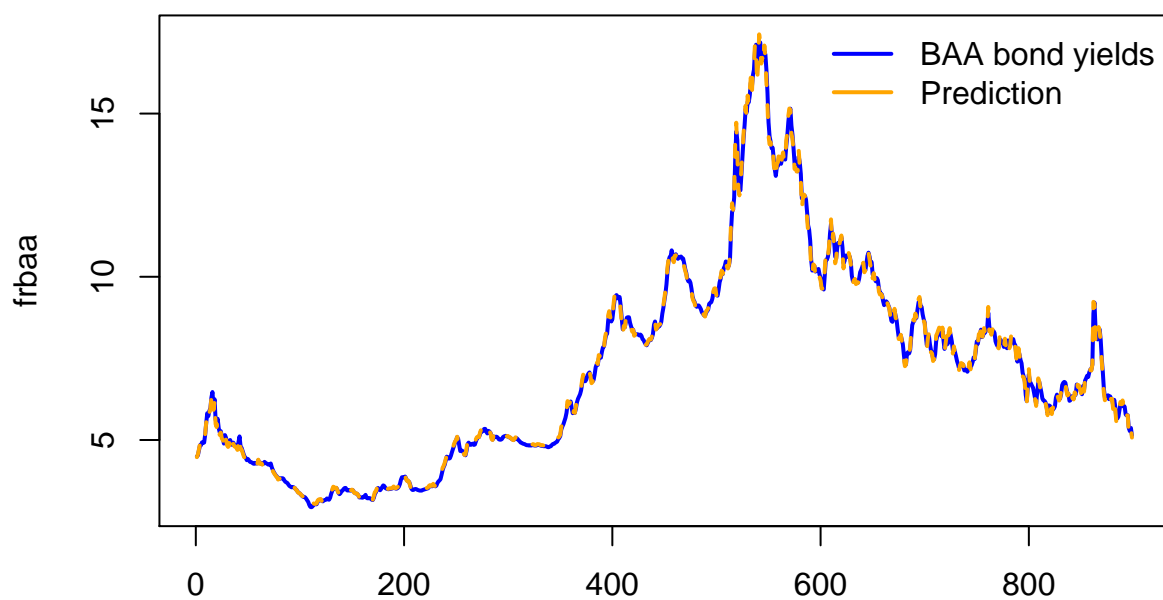
AAA bond yields and prediction



Plot predictions of the aaa bond yields

```
frbaa = cbind(rbaa[3:length(rbaa)],forecX[,2])
matplot(frbaa,col =c("blue","orange"),type="l",lwd=c(2,2),
        main = "BAA bond yields and prediction")
legend("topright", c("BAA bond yields","Prediction"), lwd=c(2,2),
      col = c("blue","orange"), bty="n")
```

BAA bond yields and prediction



Figures show that cointegration model preserved long term dependence of both AAA and BAA Bond Yields. Difference the forecasts and plot them.

```
dfraaaa = diff(fraaaa)
dfrbaa = diff(frbaa)
plot(dfraaaa,dfrbaa,col ="blue",
     main = "Scatter plot for change of prediction for AAa and BAA Yields",
     xlab="Differenced Forecasts of AAA Yields",ylab="Differenced Forecasts of BAA Yields")
```

Scatter plot for change of prediction for AAa and BAA Yields

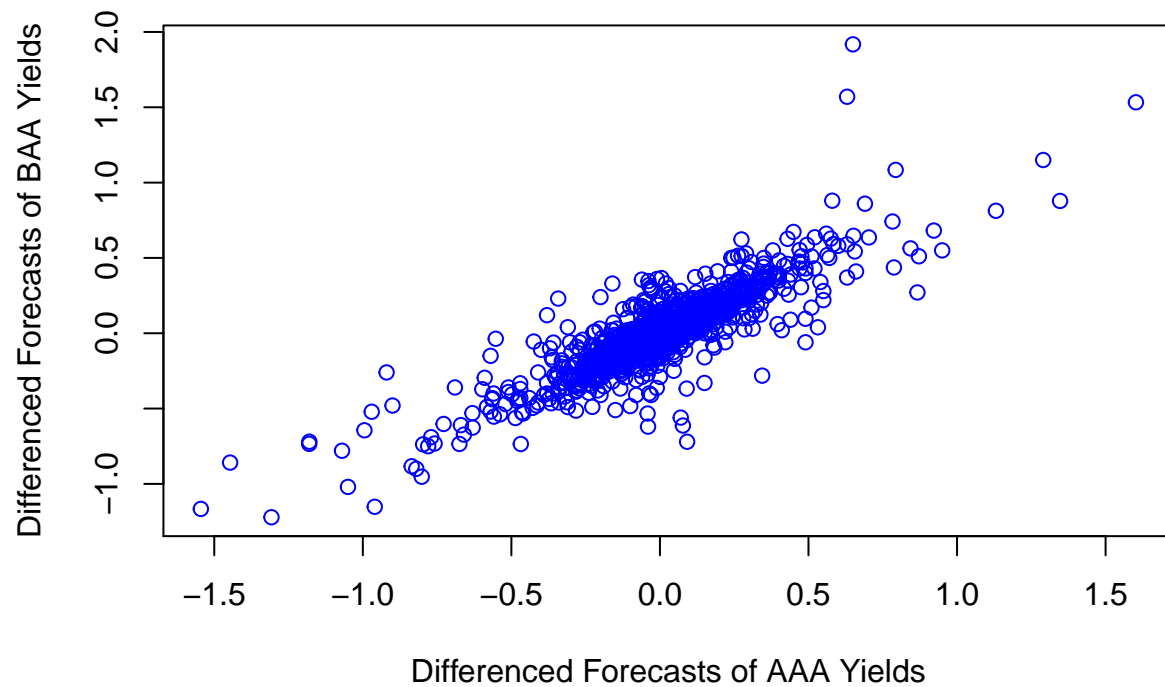
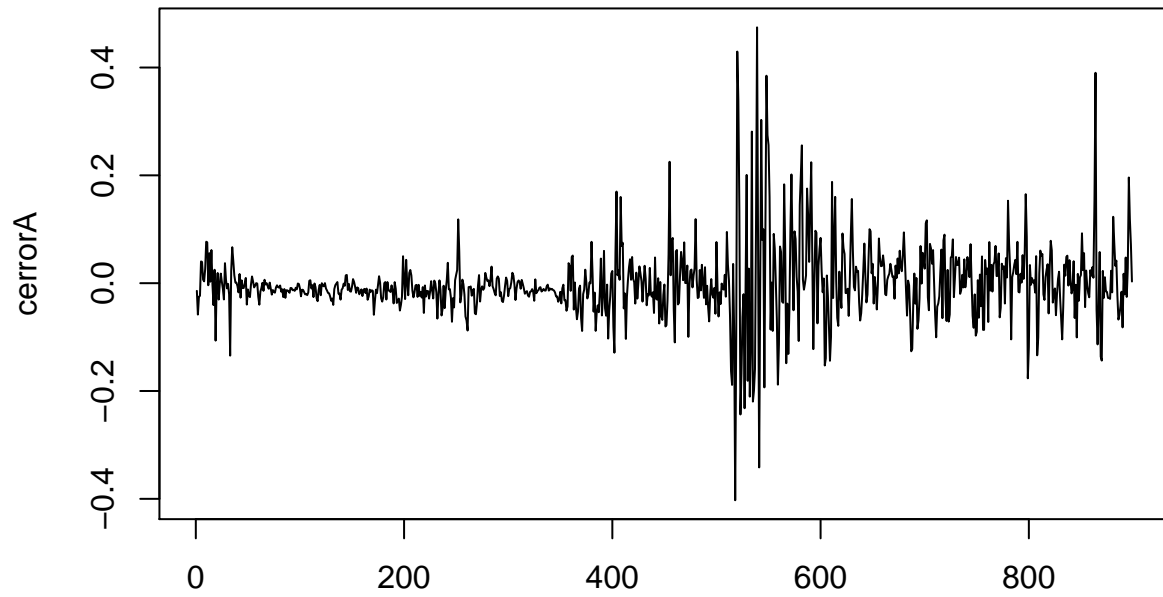


Figure shows that cointegration model also captured short term dependence of Yields differences.

Check errors of prediction by the cointegration model.

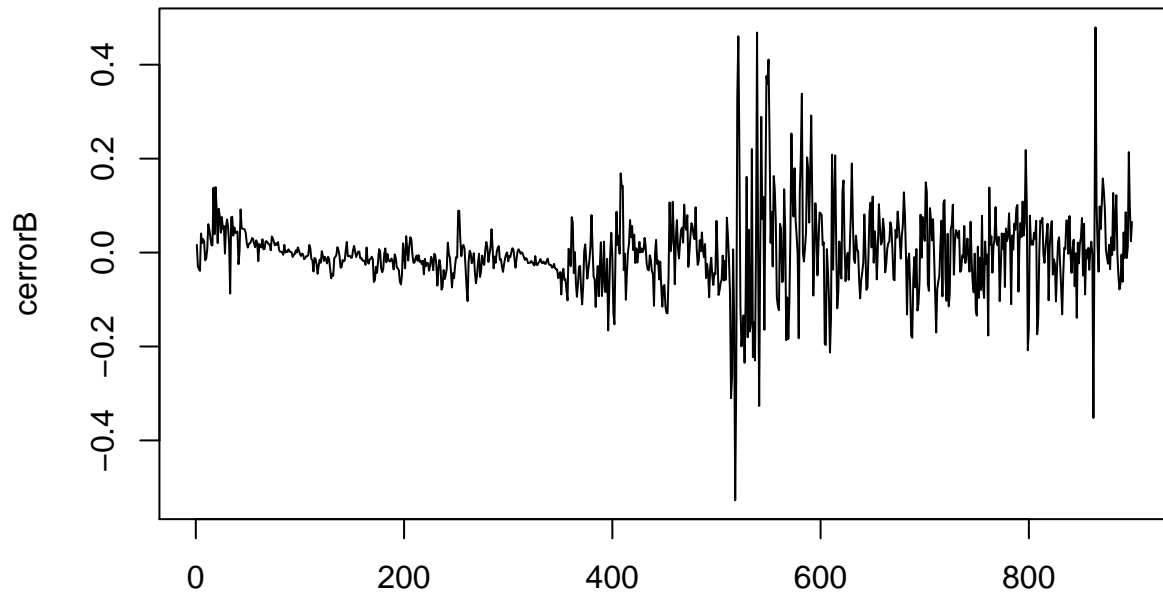
```
errorA<-raaa[3:length(raaa)]-forecX[,1]
errorB<-rbaa[3:length(rbaa)]-forecX[,2]
matplot(errorA,main = " Error of Prediction of AAA Bond Yield",type = "l")
```

Error of Prediction of AAA Bond Yield



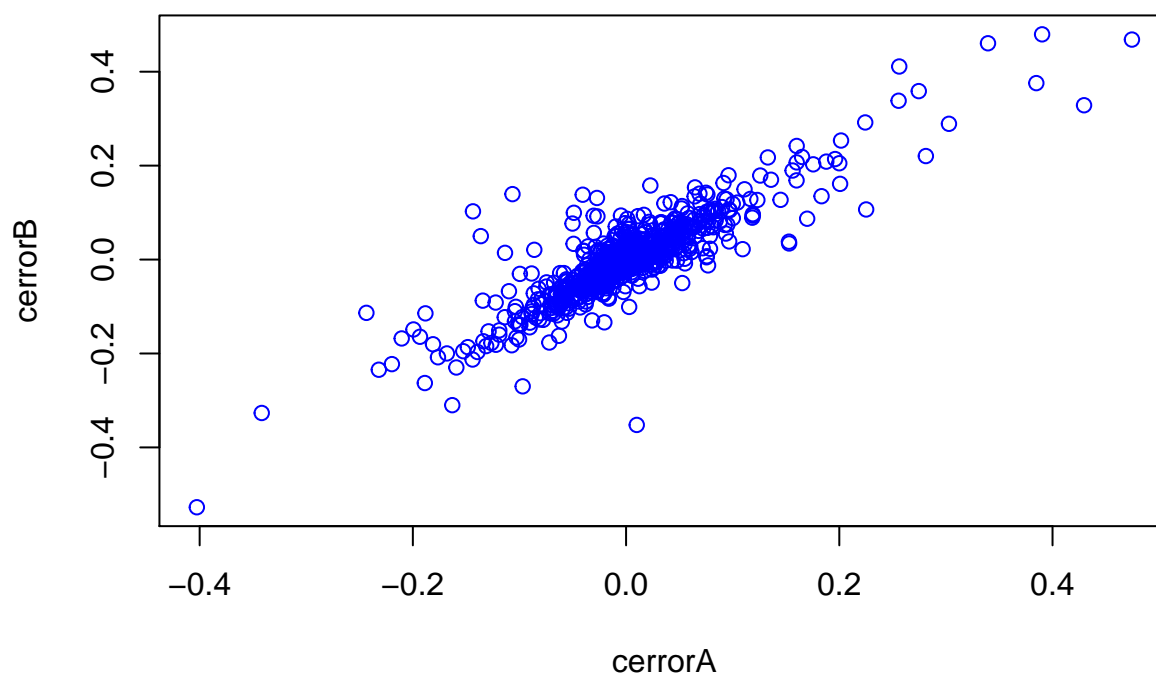
```
matplot(cerrorB,main = " Error of Prediction of BAA Bond Yield",type = "l")
```

Error of Prediction of BAA Bond Yield



```
plot(cerrorA,ccerrorB,col ="blue",main = "Scatter plot for errors of prediction for AAA and BAA Bond Yie.
```


Scatter plot for errors of prediction for AAA and BAA Bond Yields



```
cor(cbind(errorA,errorB))
```

```
##           errorA   errorB
## errorA 1.0000000 0.8770328
## errorB 0.8770328 1.0000000
```

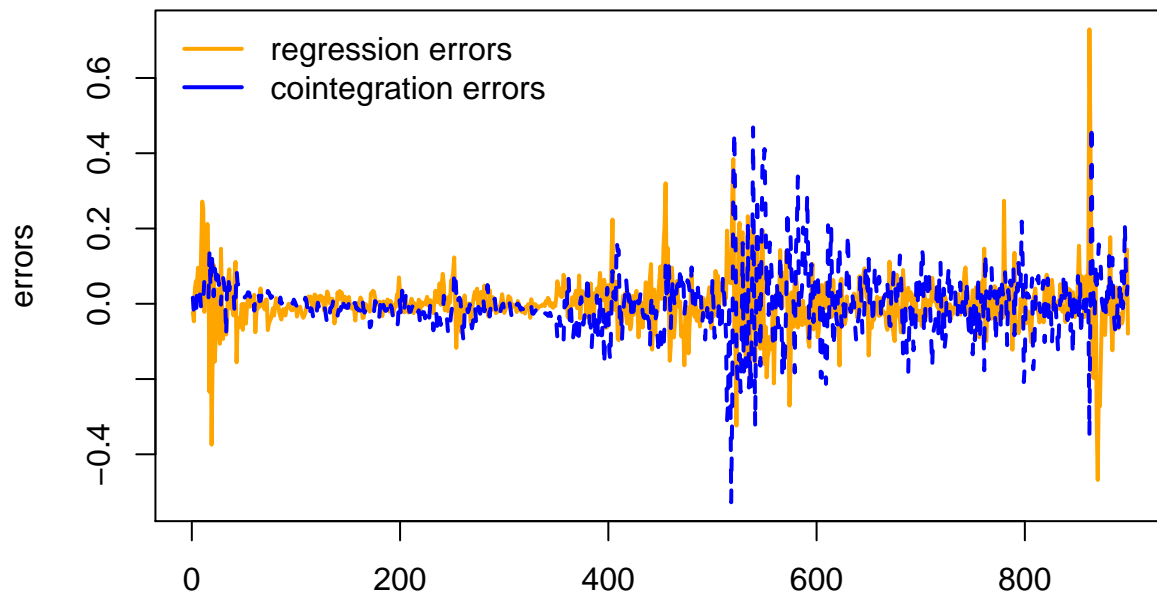
The standard errors for both bonds using cointegration model are highly correlated. They tend to increase together.

Model Comparison

Compare the errors of the regression model with stationary residuals for BAA Bond to the errors of cointegration model for the BAA Bond.

```
linregerror<- rbaa[-1]-forec
errors<-cbind(linregerror[-1],errorB)
matplot(errors,type="l",col=c("orange","blue"),lwd=2,
        main="3-Year Yield Errors for Regression and Cointegration Model")
legend("topleft",c("regression errors","cointegration errors"),lwd=2,
      col=c("orange","blue"),bty="n")
```

3-Year Yield Errors for Regression and Cointegration Model



```
paste("The total variance for Regression Model Error is ", var(errors[,1]))
```

```
## [1] "The total variance for Regression Model Error is  0.00529261835839574"
```

```
paste("The total variance for Cointegration Model Error is ", var(errors[,2]))
```

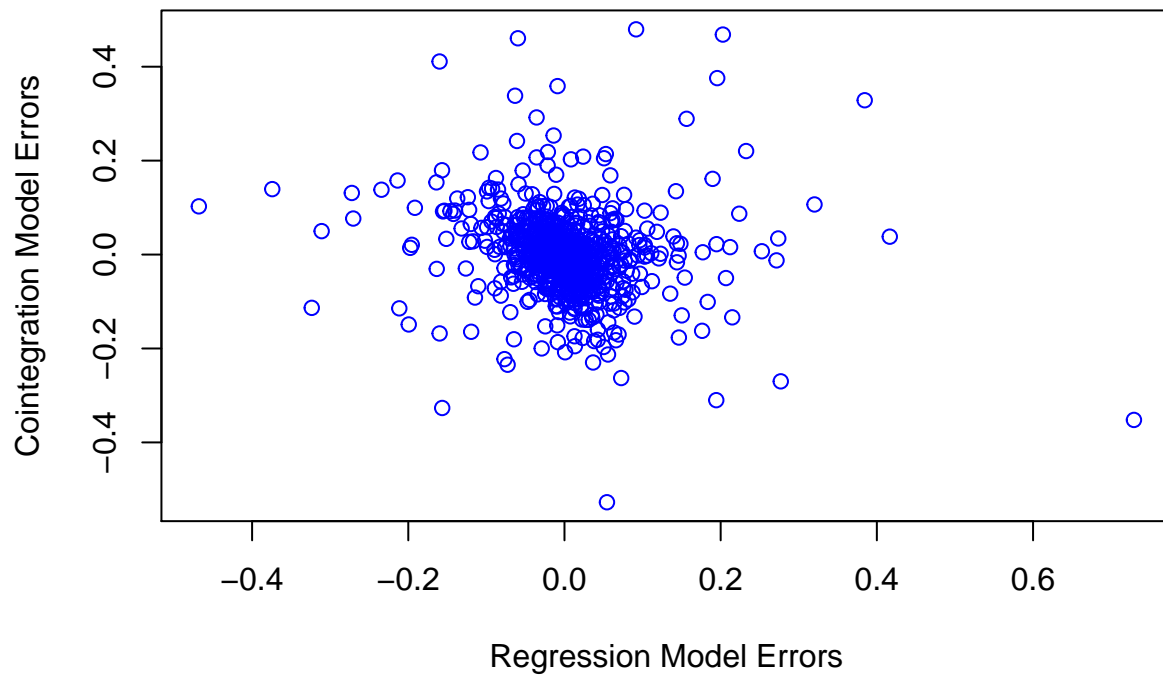
```
## [1] "The total variance for Cointegration Model Error is  0.00686033294477217"
```

The variance level of cointegration errors is higher than the regression model errors.

Check for a relationship between the errors of the two models

```
plot(errors[,1],errors[,2],col = "blue",  
      main = "Scatter Plot of Regression model Errors vs Cointegration errors",  
      xlab="Regression Model Errors", ylab="Cointegration Model Errors")
```

Scatter Plot of Regression model Errors vs Cointegration errors



```
cor(errors)
```

```
##               cerrorB
##      1.0000000 -0.1563515
## cerrorB -0.1563515  1.0000000
```

There is a small negative correlation between the errors of the two models.

Based on the prediction plots and error plots, both seem to be valid models for predicting the last 900 observations of the data. However the regression model did a better job of forecasting based on comparing the variance of the errors of both models.