

ISyE6402Project

Jesse Haulk

April 24, 2020

```
rm(list=ls())
setwd('C:/Users/super/Desktop/ISYE6402Work/Project')
```

```
library(xts)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
## Registered S3 method overwritten by 'xts':
##   method      from
##   as.zoo.xts zoo
```

```
library(quantmod)
```

```
## Loading required package: TTR
## Warning: package 'TTR' was built under R version 3.6.2
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library(aTSA)
```

```
##
## Attaching package: 'aTSA'
## The following object is masked from 'package:graphics':
##
##   identify
```

```
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 3.6.2
##
## Attaching package: 'PerformanceAnalytics'
## The following object is masked from 'package:graphics':
##
##   legend
```

```
library(tseries)
```

```
##  
## Attaching package: 'tseries'  
## The following objects are masked from 'package:aTSA':  
##  
##   adf.test, kpss.test, pp.test
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.2  
## Registered S3 methods overwritten by 'forecast':  
##   method      from  
##   fitted.fracdiff  fracdiff  
##   residuals.fracdiff fracdiff  
##  
## Attaching package: 'forecast'  
## The following object is masked from 'package:aTSA':  
##  
##   forecast
```

```
library(vars)
```

```
## Warning: package 'vars' was built under R version 3.6.2  
## Loading required package: MASS  
## Warning: package 'MASS' was built under R version 3.6.2  
## Loading required package: strucchange  
## Warning: package 'strucchange' was built under R version 3.6.2  
## Loading required package: sandwich  
## Loading required package: urca  
## Loading required package: lmtest  
##  
## Attaching package: 'vars'  
## The following object is masked from 'package:aTSA':  
##  
##   arch.test
```

```
getSymbols("PEP",src="yahoo")
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use  
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")  
## and getOption("getSymbols.auto.assign") will still be checked for  
## alternate defaults.  
##  
## This message is shown once per session and may be disabled by setting  
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.  
## [1] "PEP"
```

```
getSymbols("KO",src="yahoo")
```

```
## [1] "KO"
```

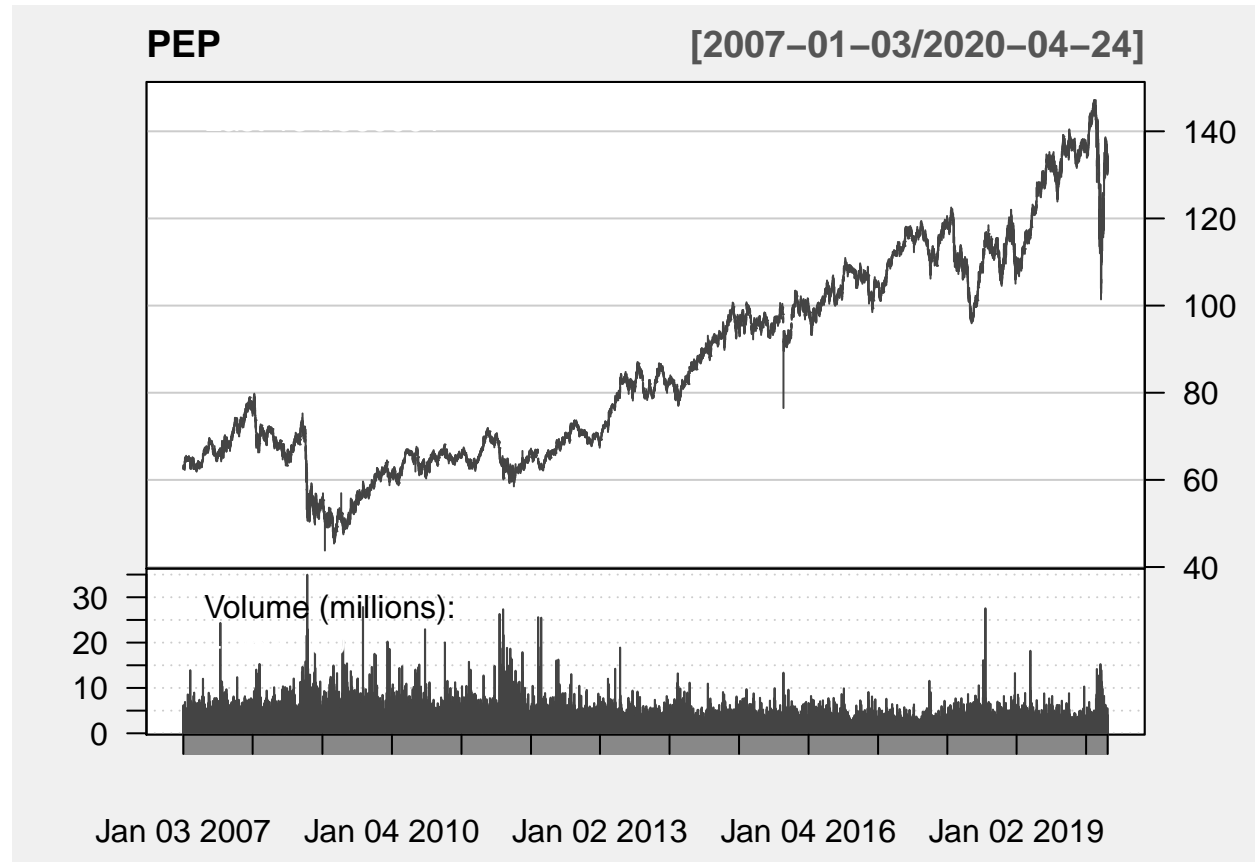
```
KO.df = read.csv('KO.csv')
```

```
PEP.df = read.csv('PEP.csv')
```

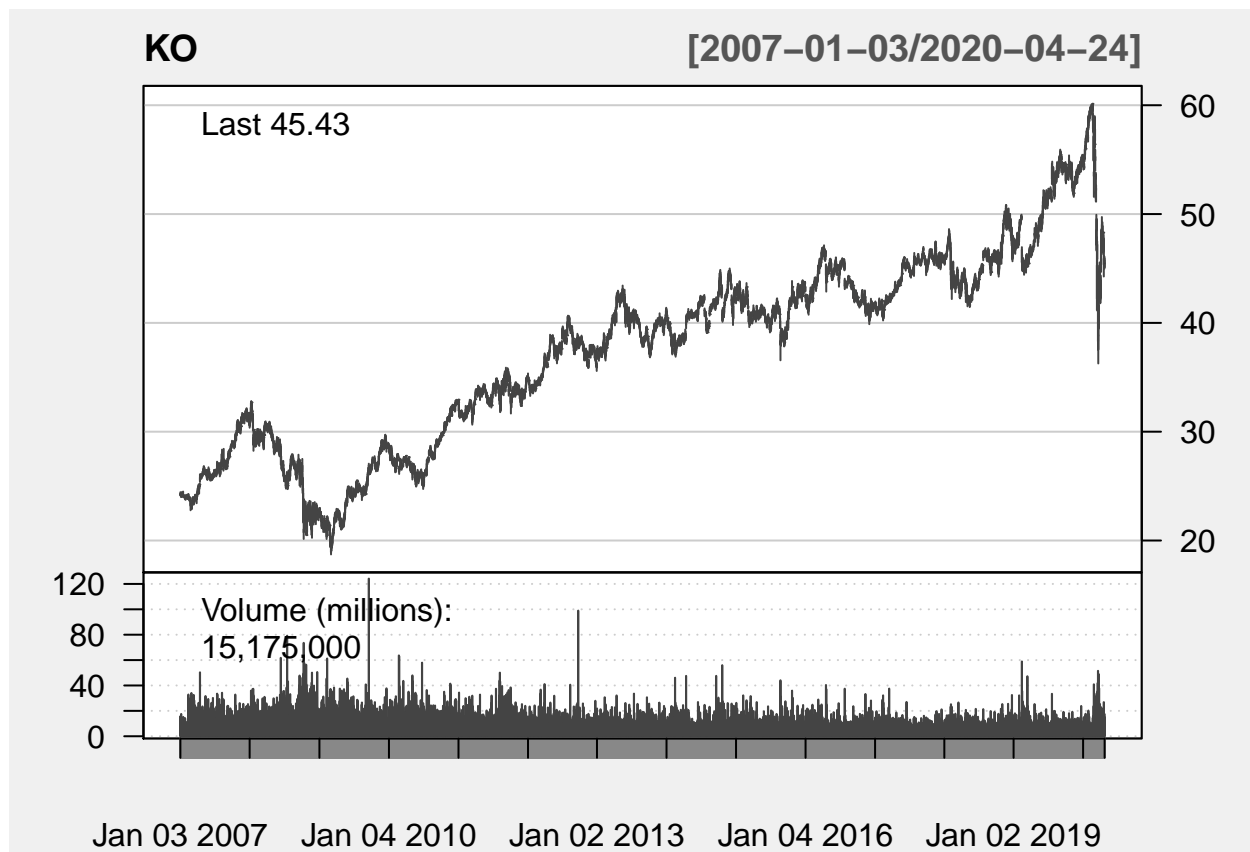
```
KO.df['dReturns'] = (KO.df$Open-KO.df$Adj.Close)/KO.df$Adj.Close
```

```
PEP.df['dReturns'] = (PEP.df$Open-PEP.df$Adj.Close)/PEP.df$Adj.Close
```

```
candleChart(PEP, multi.col = T, theme='white')
```



```
candleChart(KO, multi.col = T, theme='white')
```



#Convert to xts

```
KO.df$Date = as.POSIXct(as.character(KO.df$Date, format = '%Y/%m/%d'))
PEP.df$Date = as.POSIXct(as.character(PEP.df$Date, format = '%Y/%m/%d'))
```

```
KO.xts = xts(KO.df$dReturns, KO.df$Date)
PEP.xts = xts(PEP.df$dReturns, PEP.df$Date)
```

```
KO.2019 = KO.xts['2019']
PEP.2019 = PEP.xts['2019']
```

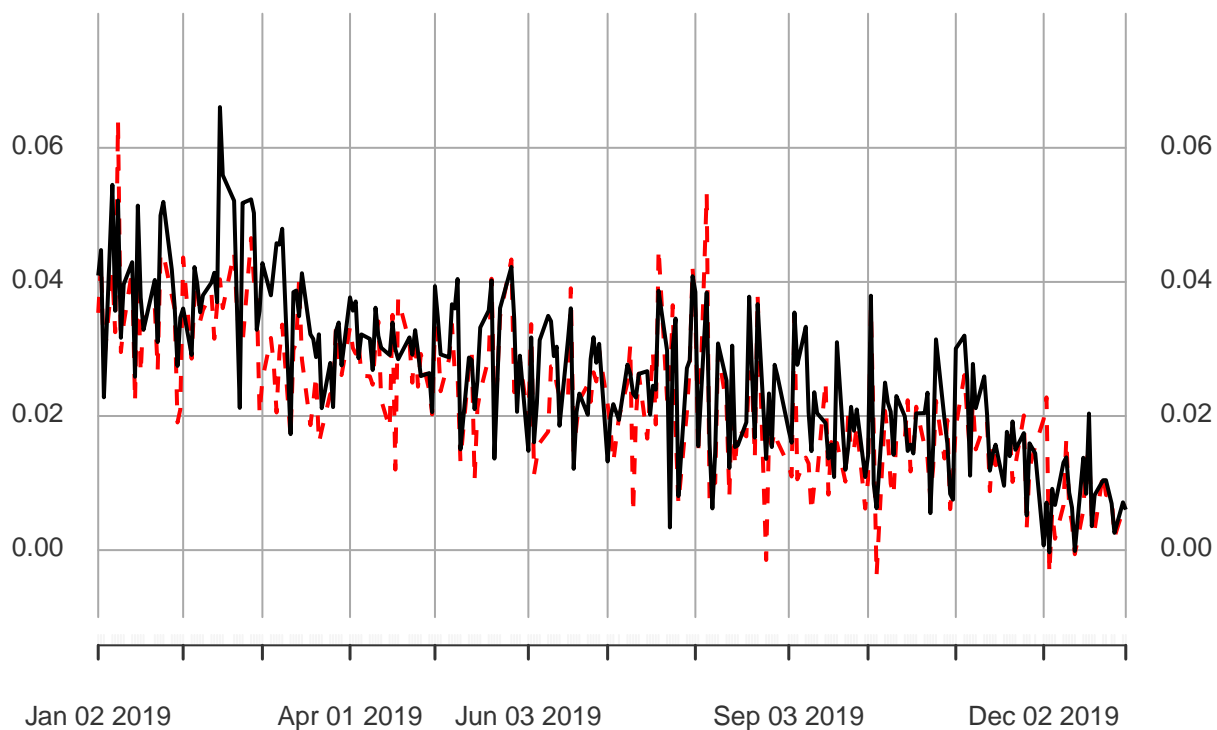
```
data.xts = merge(KO.2019, PEP.2019)
```

#plot the returns

```
plot(data.xts, ylim=c(-0.01,0.08), main='Returns for KO and PEP', col=c('black', 'red'), lty = c(1,2))
```

Returns for KO and PEP

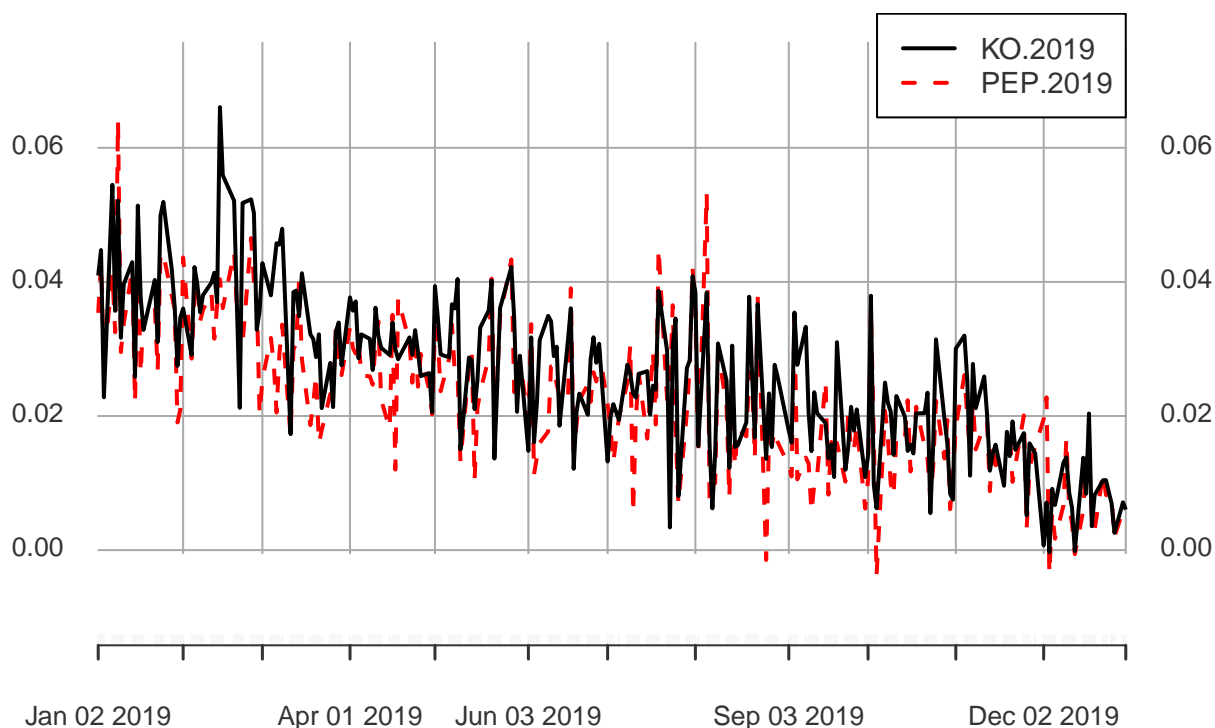
2019-01-02 / 2019-12-31



```
addLegend('topright', lty= c(1,2), lwd=c(2,2), col = c('black', 'red'), bg='white', bty='o')
```

Returns for KO and PEP

2019-01-02 / 2019-12-31



#is the Spread mean reverting?

```
Diff.2019 = KO.2019-PEP.2019
```

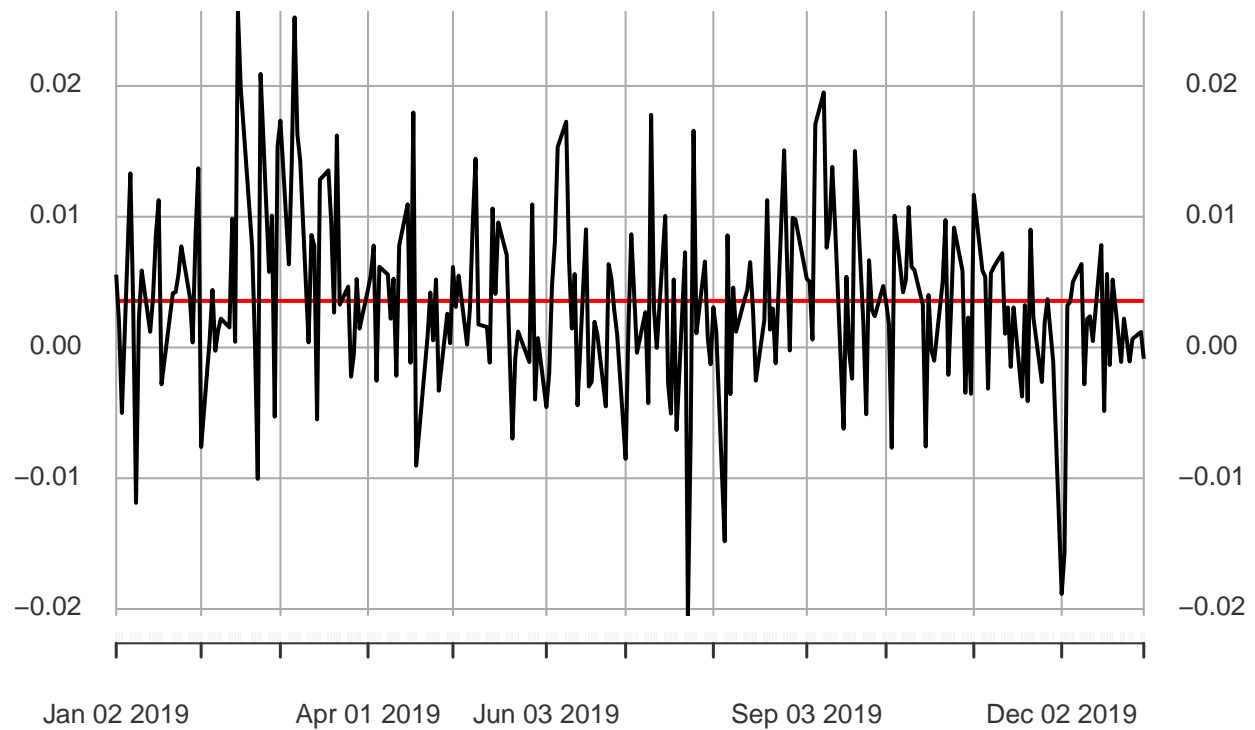
```
mean = mean(Diff.2019)
mean.df = cbind(rep(0, 1259), as.character(KO.df$Date))
mean.df = as.data.frame(mean.df)
colnames(mean.df) = c('mean', 'Date')
mean.df$mean = mean
mean.df$Date = as.POSIXct(as.character(mean.df$Date, format = '%Y/%m/%d'))
mean.xts = xts(mean.df$mean, mean.df$Date)
mean.2019 = mean.xts['2019']
```

```
data2.xts = merge(Diff.2019, mean.2019)
```

#plot Spread

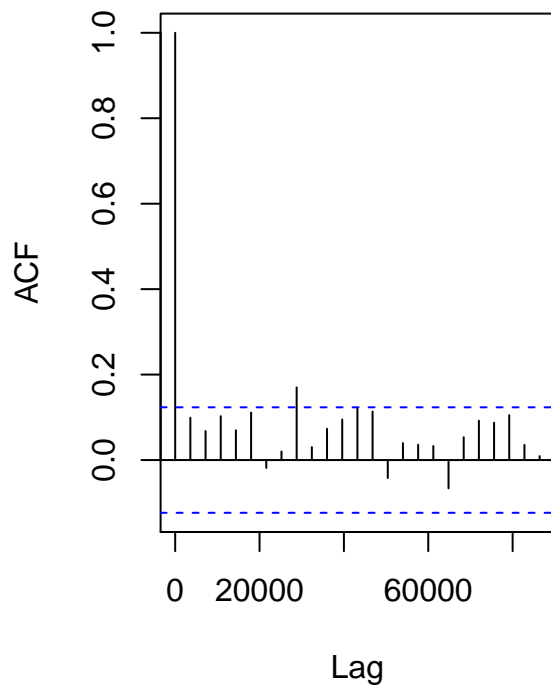
```
plot(data2.xts, main='')
```

2019-01-02 / 2019-12-31

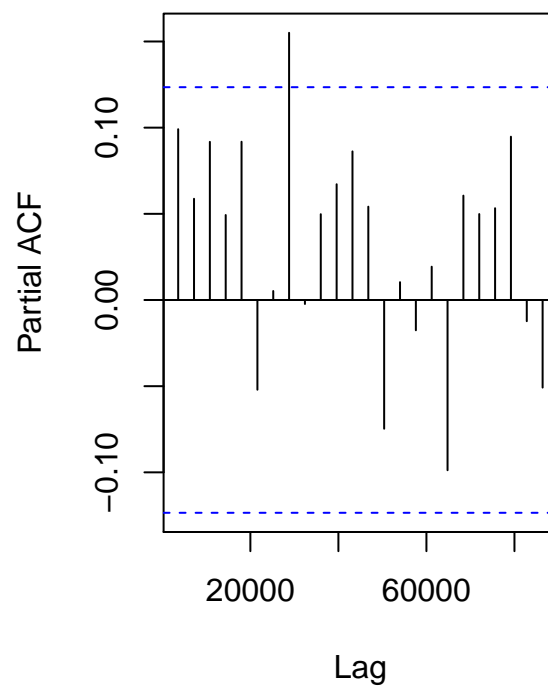


```
par(mfrow=c(1,2))
par(mar=c(5,4,4,2))
acf(Diff.2019, main= 'ACF of the Spread')
pacf(Diff.2019, main= 'PACF of the Spread')
```

ACF of the Spread



PACF of the Spread



#Test for Cointegration

```
tseries::adf.test(Diff.2019, alternative = 's')
```

```
## Warning in tseries::adf.test(Diff.2019, alternative = "s"): p-value smaller than
## printed p-value
```

```
##
```

```
## Augmented Dickey-Fuller Test
```

```
##
```

```
## data: Diff.2019
```

```
## Dickey-Fuller = -5.686, Lag order = 6, p-value = 0.01
```

```
## alternative hypothesis: stationary
```

Small p-value rejects the null hypothesis of non-stationarity. Yes, the spread is likely mean reverting.

###Data Preparation for Training and Testing

```
train.2019 = Diff.2019['2019-01/2019-11']
```

```
test.2019 = Diff.2019['2019-12']
```

ARIMA model

```
final.aic = Inf
```

```
final.order = c(0,0,0,0)
```

```
for (p in 1:6) for (d in 0:1) for (q in 1:6) for (s in 0:1){
```

```
  current.aic = AIC(Arima(train.2019, order=c(p,d,q), seasonal = list(order=c(0,s,0)), method='ML'))
```



```

if (current.aic < final.aic){
  final.aic = current.aic
  final.order = c(p,d,q,s)
}
}

```

Lowest AIC is ARIMA(1,0,1) model for the 2019 returns difference.

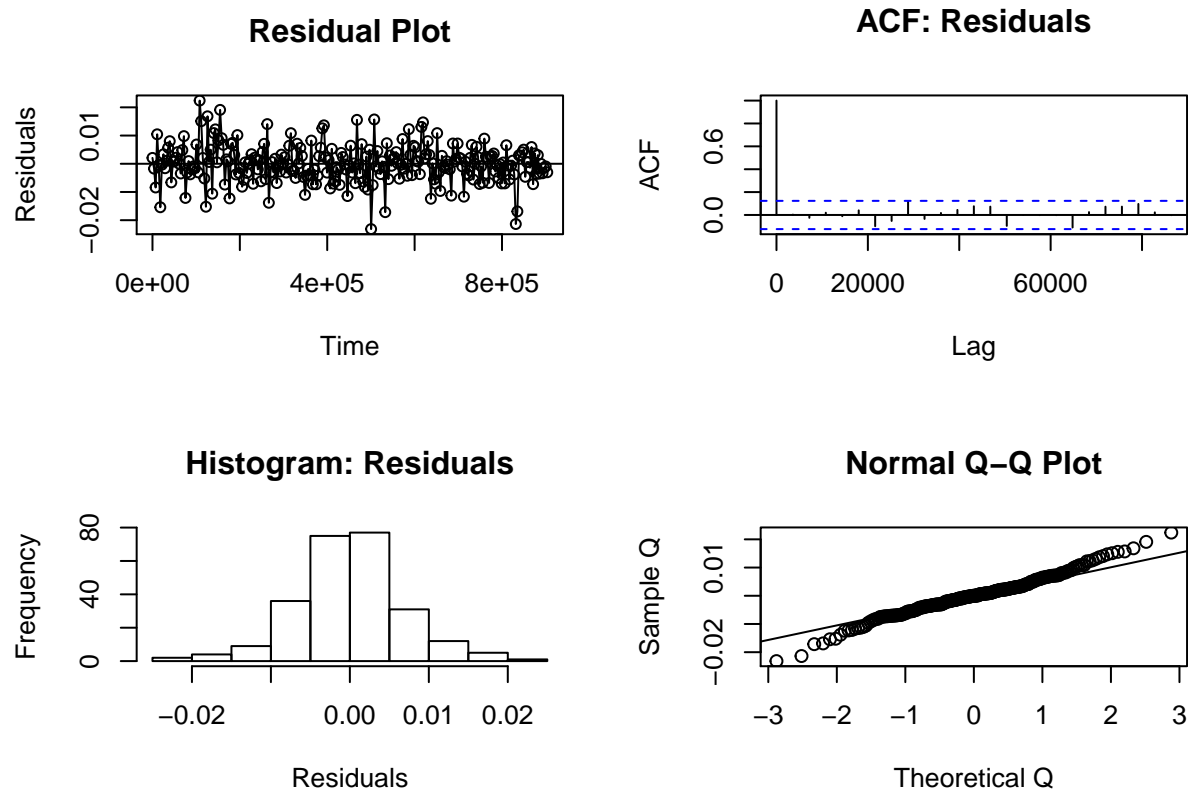
```
model.arima = Arima(Diff.2019, order=c(1,0,1), method='ML')
```

#Residual Analysis

```

par(mfrow=c(2,2))
plot(resid(model.arima), ylab='Residuals',type='o',main="Residual Plot")
abline(h=0)
acf(resid(model.arima),main="ACF: Residuals")
hist(resid(model.arima),xlab='Residuals',main='Histogram: Residuals')
qqnorm(resid(model.arima),ylab="Sample Q",xlab="Theoretical Q")
qqline(resid(model.arima))

```



The residual plot shows the variance is constant across time. Histogram of the Residuals shows unimodal normal distribution. From the Q-Q Plot, the data looks normally distributed with fat tails. There's no residuals that fall outside 0.05 in the acf plot.

```
Box.test(model.arima$residuals, lag = 6, type='Box-Pierce', fitdf = 4)
```

```
##
## Box-Pierce test
```

```
##
## data: model.arima$residuals
## X-squared = 3.1318, df = 2, p-value = 0.2089
Box.test(model.arima$residuals, lag = 6, type='Ljung-Box', fitdf = 4)
```

```
##
## Box-Ljung test
##
## data: model.arima$residuals
## X-squared = 3.2278, df = 2, p-value = 0.1991
```

High p-values indicate independence in the residuals.

#Test

```
fore = forecast::forecast(model.arima, h=21)
fore = as.data.frame(fore)
```

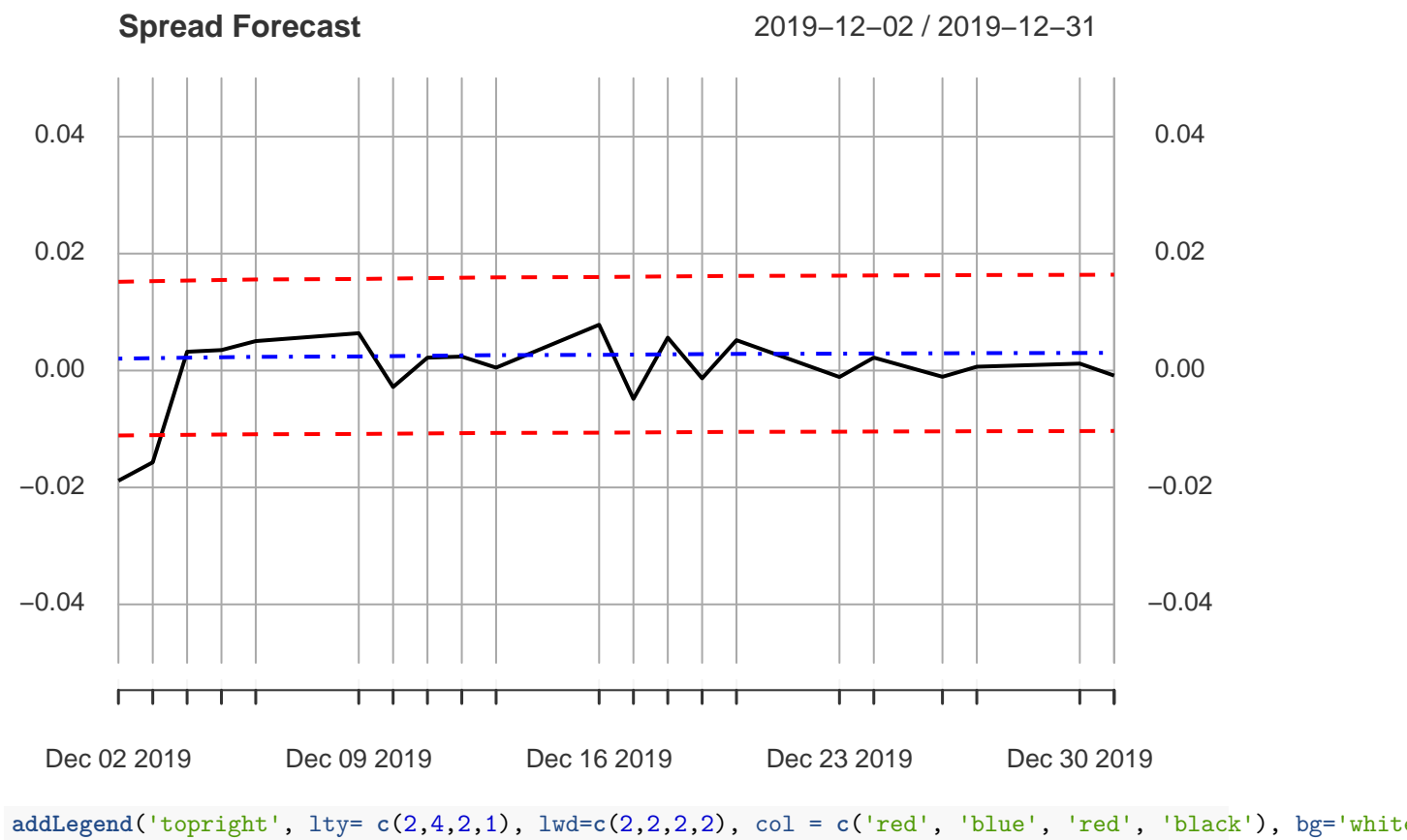
```
point.fore = cbind(rep(0,21), as.character(index(test.2019)))
point.fore = as.data.frame(point.fore)
colnames(point.fore)= c('Point Estimate', 'Date')
point.fore$`Point Estimate` = fore$`Point Forecast`
point.fore$Date = as.POSIXct(as.character(point.fore$Date, format = '%Y/%m/%d'))
point.fore.xts = xts(point.fore$`Point Estimate`, point.fore$Date)
```

```
lo.fore = cbind(rep(0,21), as.character(index(test.2019)))
lo.fore = as.data.frame(lo.fore)
colnames(lo.fore)= c('Low Estimate', 'Date')
lo.fore$`Low Estimate` = fore$`Lo 95`
lo.fore$Date = as.POSIXct(as.character(lo.fore$Date, format = '%Y/%m/%d'))
lo.fore.xts = xts(lo.fore$`Low Estimate`, lo.fore$Date)
```

```
hi.fore = cbind(rep(0,21), as.character(index(test.2019)))
hi.fore = as.data.frame(hi.fore)
colnames(hi.fore)= c('High Estimate', 'Date')
hi.fore$`High Estimate` = fore$`Hi 95`
hi.fore$Date = as.POSIXct(as.character(hi.fore$Date, format = '%Y/%m/%d'))
hi.fore.xts = xts(hi.fore$`High Estimate`, hi.fore$Date)
```

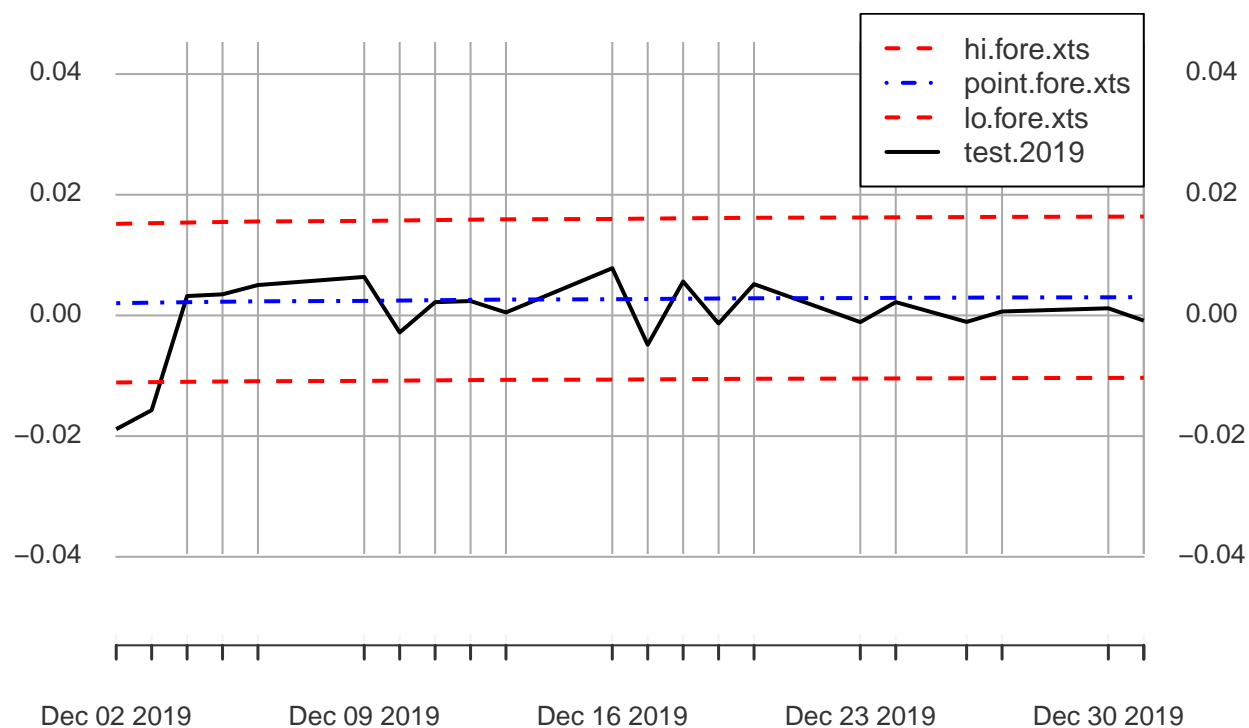
```
fore.merge = merge(hi.fore.xts, point.fore.xts, lo.fore.xts, test.2019)
```

```
plot(fore.merge, ylim=c(-0.05,0.05), main='Spread Forecast', col = c('red', 'blue', 'red', 'black'), lty=
```



Spread Forecast

2019-12-02 / 2019-12-31



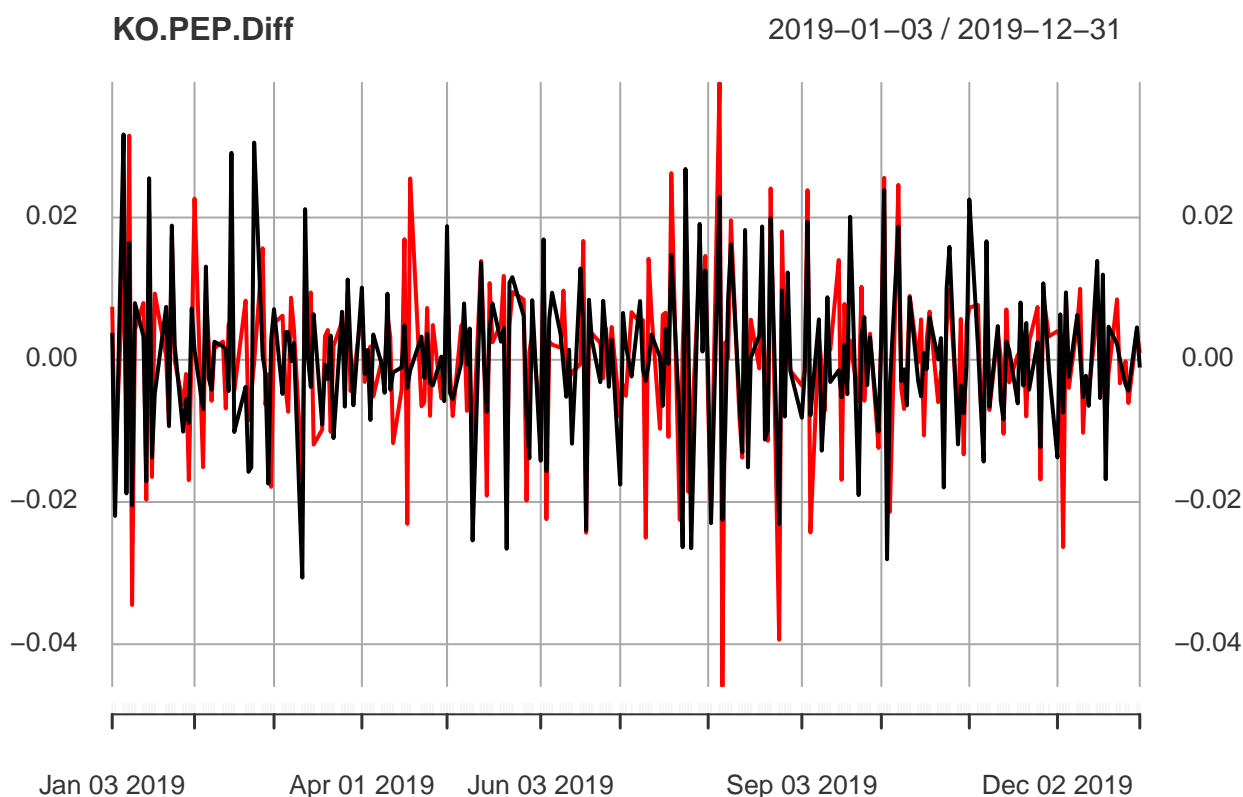
#VAR Model

```
KO.PEP.2019 = merge(KO.2019, PEP.2019)
```

```
KO.PEP.Diff = diff(KO.PEP.2019)
```

```
KO.PEP.Diff = na.omit(KO.PEP.Diff)
```

```
plot(KO.PEP.Diff)
```



```
model.var = VAR(KO.PEP.Diff)
summary(model.var)
```

```
##
## VAR Estimation Results:
## =====
## Endogenous variables: KO.2019, PEP.2019
## Deterministic variables: const
## Sample size: 250
## Log Likelihood: 1676.444
## Roots of the characteristic polynomial:
## 0.5095 0.4551
## Call:
## VAR(y = KO.PEP.Diff)
##
##
## Estimation results for equation KO.2019:
## =====
## KO.2019 = KO.2019.l1 + PEP.2019.l1 + const
##
##               Estimate Std. Error t value Pr(>|t|)
## KO.2019.l1  -0.4551061  0.0763135  -5.964 8.49e-09 ***
## PEP.2019.l1  -0.0001919  0.0732168  -0.003  0.998
## const        -0.0002165  0.0006240  -0.347  0.729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

##
##
## Residual standard error: 0.009865 on 247 degrees of freedom
## Multiple R-Squared: 0.2073, Adjusted R-squared: 0.2009
## F-statistic: 32.31 on 2 and 247 DF, p-value: 3.443e-13
##
##
## Estimation results for equation PEP.2019:
## =====
## PEP.2019 = K0.2019.l1 + PEP.2019.l1 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## K0.2019.l1 -0.0068120  0.0765737  -0.089    0.929
## PEP.2019.l1 -0.5094322  0.0734665  -6.934 3.55e-11 ***
## const      -0.0002047  0.0006261  -0.327    0.744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.009899 on 247 degrees of freedom
## Multiple R-Squared: 0.2645, Adjusted R-squared: 0.2585
## F-statistic: 44.41 on 2 and 247 DF, p-value: < 2.2e-16
##
##
## Covariance matrix of residuals:
##           K0.2019  PEP.2019
## K0.2019  9.733e-05 6.540e-05
## PEP.2019 6.540e-05 9.799e-05
##
## Correlation matrix of residuals:
##           K0.2019  PEP.2019
## K0.2019  1.0000  0.6697
## PEP.2019 0.6697  1.0000
##
## Restricted VAR
model.var.restrict = restrict(model.var)
summary(model.var.restrict)
##
## VAR Estimation Results:
## =====
## Endogenous variables: K0.2019, PEP.2019
## Deterministic variables: const
## Sample size: 250
## Log Likelihood: 1676.368
## Roots of the characteristic polynomial:
## 0.455 0.5136
## Call:
## VAR(y = K0.PEP.Diff)
##
##
## Estimation results for equation K0.2019:
## =====
## K0.2019 = K0.2019.l1

```

```

##
##           Estimate Std. Error t value Pr(>|t|)
## K0.2019.l1 -0.45500      0.05642  -8.065 3.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.009828 on 249 degrees of freedom
## Multiple R-Squared: 0.2071, Adjusted R-squared: 0.2039
## F-statistic: 65.04 on 1 and 249 DF, p-value: 3.095e-14
##
##
## Estimation results for equation PEP.2019:
## =====
## PEP.2019 = PEP.2019.l1
##
##           Estimate Std. Error t value Pr(>|t|)
## PEP.2019.l1 -0.51363      0.05431  -9.457 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.009862 on 249 degrees of freedom
## Multiple R-Squared: 0.2643, Adjusted R-squared: 0.2613
## F-statistic: 89.43 on 1 and 249 DF, p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##           K0.2019 PEP.2019
## K0.2019  9.733e-05 6.54e-05
## PEP.2019 6.540e-05 9.80e-05
##
## Correlation matrix of residuals:
##           K0.2019 PEP.2019
## K0.2019  1.0000  0.6697
## PEP.2019 0.6697  1.0000

```