

Econ 6376 Assignment 3

Author: Chris Montgomery Last Revised: 11/4/2018 Part I

The below code chunk creates the necessary inputs for the below intervention equation:

$$X_t = a_0 + a_1(t-1) + c_0 Z_t + W_t \quad (1)$$

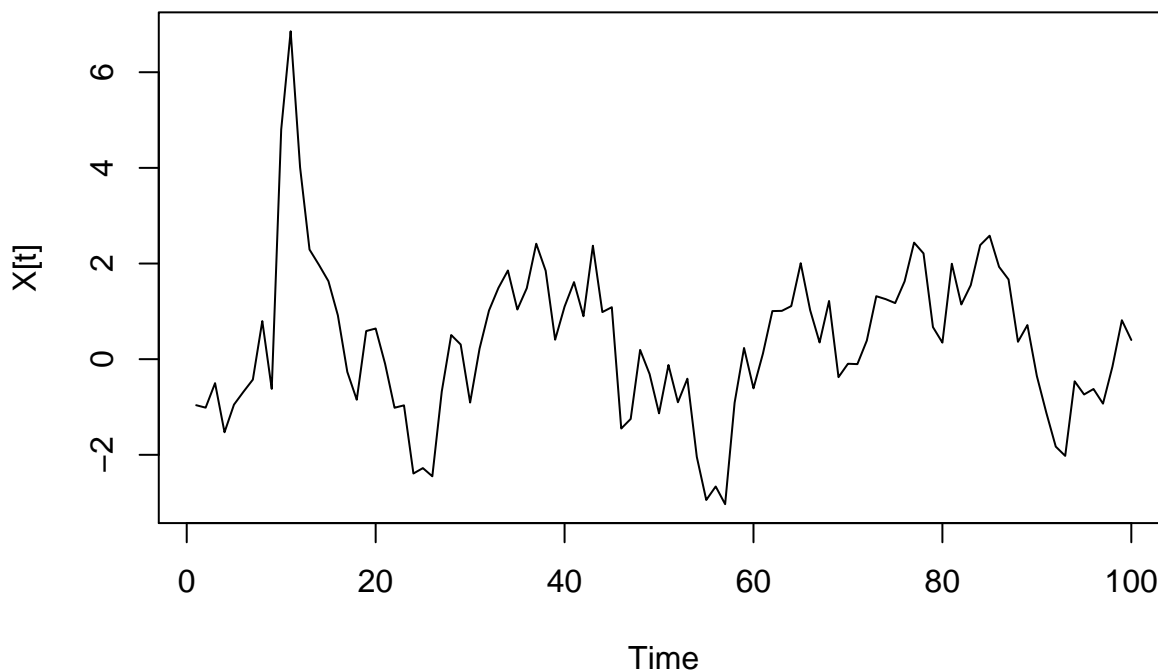
where: X_t = output at time t Z = an intervention dummy W_t = white noise forcing process a_0 ; a_1 = assumed coefficient values

```
set.seed(3)
x <- w <- rnorm(1:100)
c0 <- 4
z <- c(rep(0,100)) ; z [10:11] <- 1
a1 <- .75
a0 <- 0

for(t in 2:100){
  x[t] <- a0 + a1*x[t-1] + c0*(z[t]) + w[t]
}
```

```
plot(as.ts(x), ylab = "X[t]", main = "Simulation: White Noise Series with White Noise Intevention at t = 11")
```

Simulation: White Noise Series with White Noise Intevention at t = 11



Simulating the series yields the above plot. The visual clearly depicts a short run spike at observations 10 and 11, associated with the introduction of the intervention. Following the intervention, the series appears to converge to mean 0. This should be fairly intuitive, as the simulation assigns a a_1 parameter that is less than 1.

The below code solves for the Impulse Response function (IRF) of the intervention equation. At each time period, the IRF reports the effect of the ongoing intervention (if the intervention is still occurring) as

well as the effects on X_t associated with the intervention occurring in previous observations. Equation 2 depicts the IRF solution for a continuous intervention at a time that is i observations away from the beginning of the intervention. However, given the specifications of the above simulation, the intervention is non-continuous and thus non-permanent. Equation 3 can be used to solve for the IRF at any time t .

$$\frac{\partial y_{t+i}}{\partial z_t} = c_0(1 + a_1 + \dots + a_1^i) \quad IRF_t = c_0 \sum_{i=0}^t Z_{t-i} * a_1^i \quad (2)$$

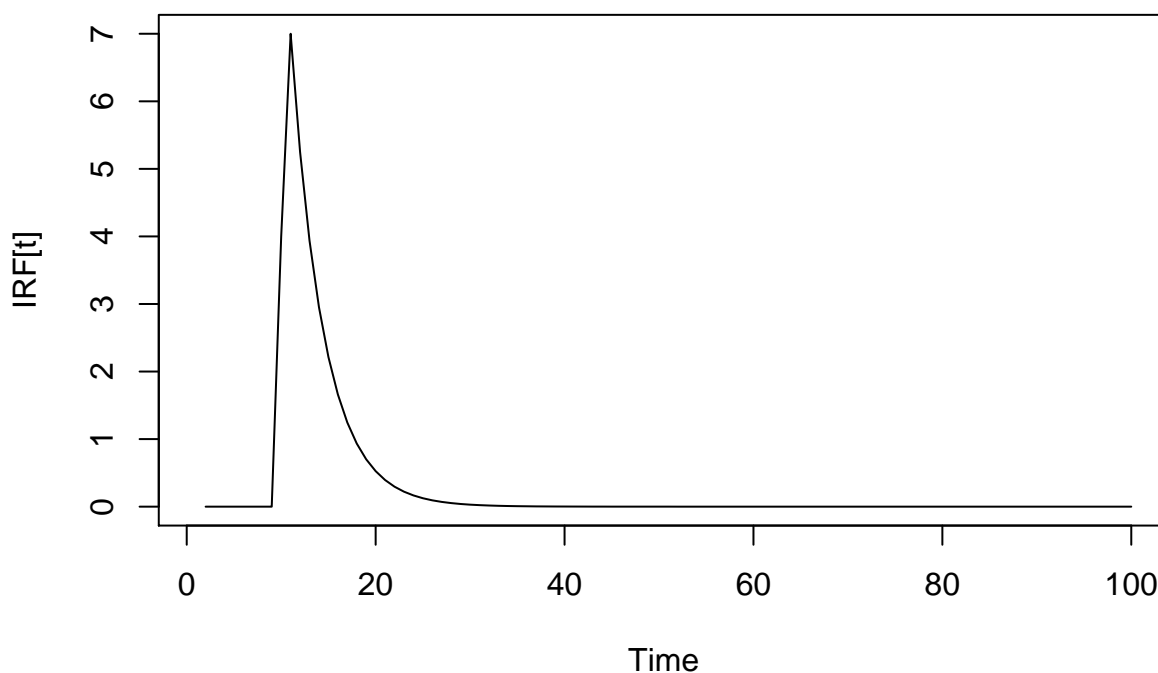
(3)

```
#create an empty vector to house our impulse values

irf <- c()

# Creates a loop that solves the IRF summation equation for each point in time from 1:100
for (i in 2:100){
  j <- i - 1
  irf[i] <- sum( c0 * z[i:1] * a1^(0:j))
}
plot(irf, main = "Impulse Response Function", xlab = "Time", ylab = "IRF[t]", type = "l")
```

Impulse Response Function



Plotting the IRF function elucidates the temporary nature of the intervention. Here we see that the IRF spikes beginning at observation 10, reaching its maximum at observation 11. At subsequent observations, the intervention dummy is turned off, and the IRF converges to zero as time increases. As such, the long run convergence of the IRF to 0 shows that, in the limit, the mean of the intervention equation remains unchanged due to the intervention. In theory, the IRF of a permanent intervention would converge to a long of 16, as denoted by the below long run IRF equation:

$$LongRunIRF = \frac{c0}{1 - a1}$$

$$16 = \frac{4}{1 - .75}$$

with $c0 = 4$ representing the initial effect of the intervention. Fitting an OLS regression does reasonably well in approximating both the initial effect of the intervention ($c0 = 4.3$), and the theorized long run IRF value (17.3). This is particularly impressive considering the fact that the intervention occurred over only 2 periods.

```
fit <- lm (x~ lag(x)+ z )
fit
```

```
##
## Call:
## lm(formula = x ~ lag(x) + z)
##
## Coefficients:
## (Intercept)      lag(x)          z
##      0.03355      0.69413      4.34451
```

*#short run effect is just the z parameter. Long run effect is equal to 11.42 according to the model fit
#it should actually converge to 10 if left on forever*

```
fit$coefficients[3] / (1 - a1)
```

```
##          z
## 17.37803
```

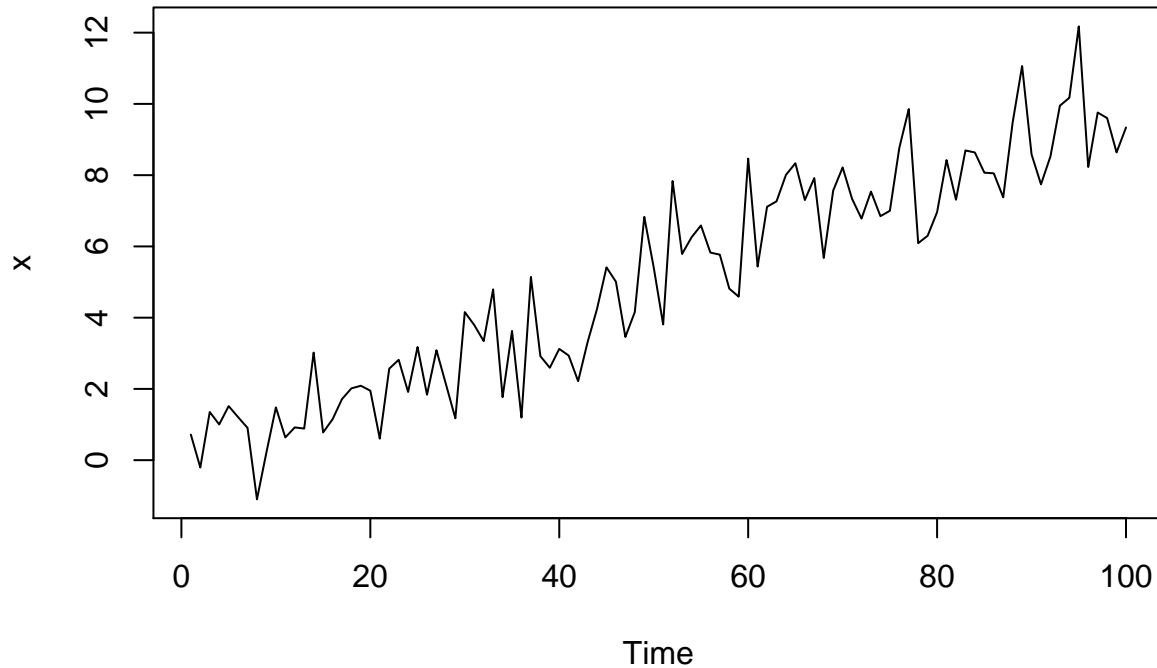
Part II

The below code simulates a time series that is dictated by a time trend and stochastic term.

```
x <- w <- rnorm(100)
a0 <- 0
a1 <- .1
time <- c()
for (t in 1:100){
  time[t] <- t
  x[t] <- a0 + a1*t + w[t]
}

plot(x, type = "l", main = "X[t] = a0 + a1*t + w[t]", xlab = "Time")
```

$$X[t] = a_0 + a_1 * t + w[t]$$



Knowing the data generating process allows us to determine which transformation is most appropriate for making the above series stationary. Given the existence of a time trend, and no AR terms, subtracting the residuals of an OLS model from the series will yield the least amount of white noise in the transformed series. This should be fairly intuitive, as the detrending method effectively removes the a_0 constant and t parameter from the equation. Thus, the detrended value of X at all t is simply a current white noise term. Taking the first difference, a suboptimal alternative transformation, introduces additional noise due to the presence of a lagged white noise term. Consider the first difference transformation:

$$X_t = a_0 + a_1 * t + w_t$$

$$X_{t-1} = a_0 + a_1 * (t - 1) + w_t - w_{t-1}$$

$$X_t - X_{t-1} = a_1 + w_t - w_{t-1}$$

The below code further elucidates the fact that detrending is a more appropriate stationary transformation for this simulation. The sum of square errors is lower for the detrended series, indicating the presence of less noise compared to the differenced series. At the same time, the ACF function of the differenced series shows a highly significant first lag. This makes sense as the first difference transformation effectively converts the series to a moving average process.

#First difference

```
diff.x <- diff(as.ts(x))
```

```
# plot(diff.x)
```

```
# mean(diff.x)
```

```
detrend <- lm(x ~ time)
```

```
detrend.x <- x - detrend$fitted.values
```

```

# plot(as.ts(detrend.x))
# mean(detrend.x)

ssr.diff <- sum((diff.x - mean(diff.x))^2); ssr.diff

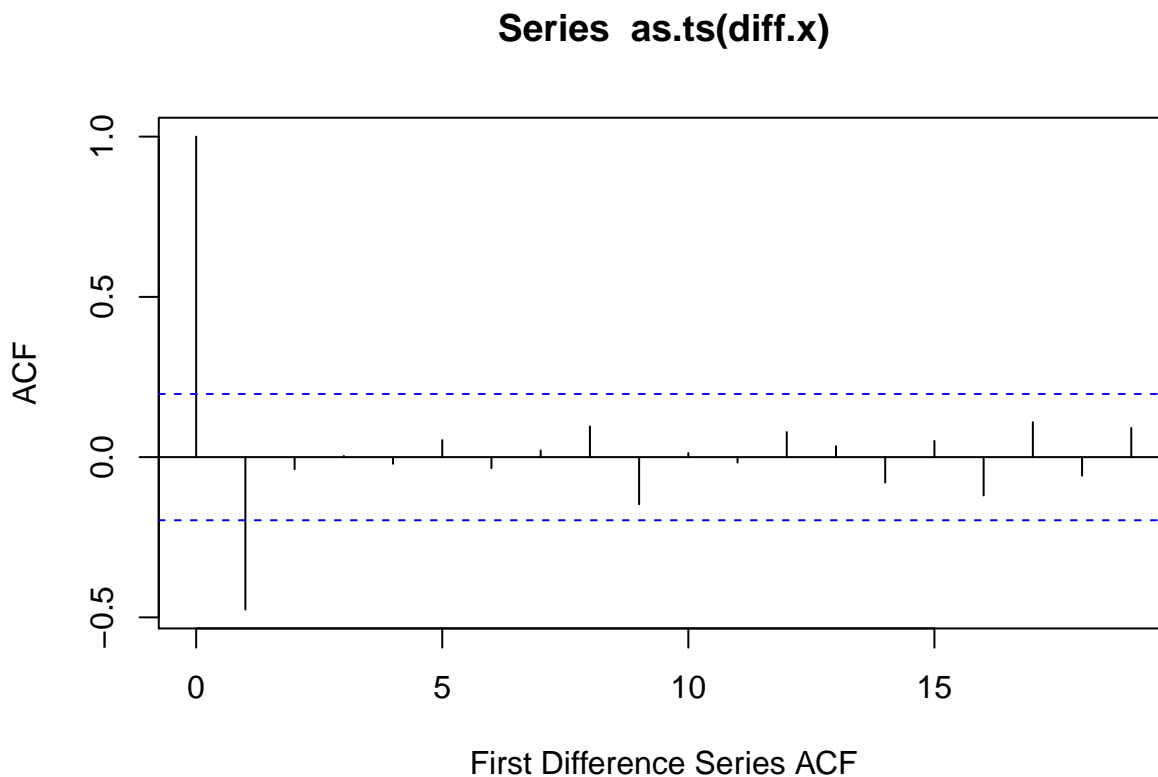
## [1] 231.6624

ssr.detrend <- sum ((detrend.x - mean(detrend.x))^2 ); ssr.detrend

## [1] 119.4269

acf(as.ts(diff.x), xlab = "First Difference Series ACF")

```

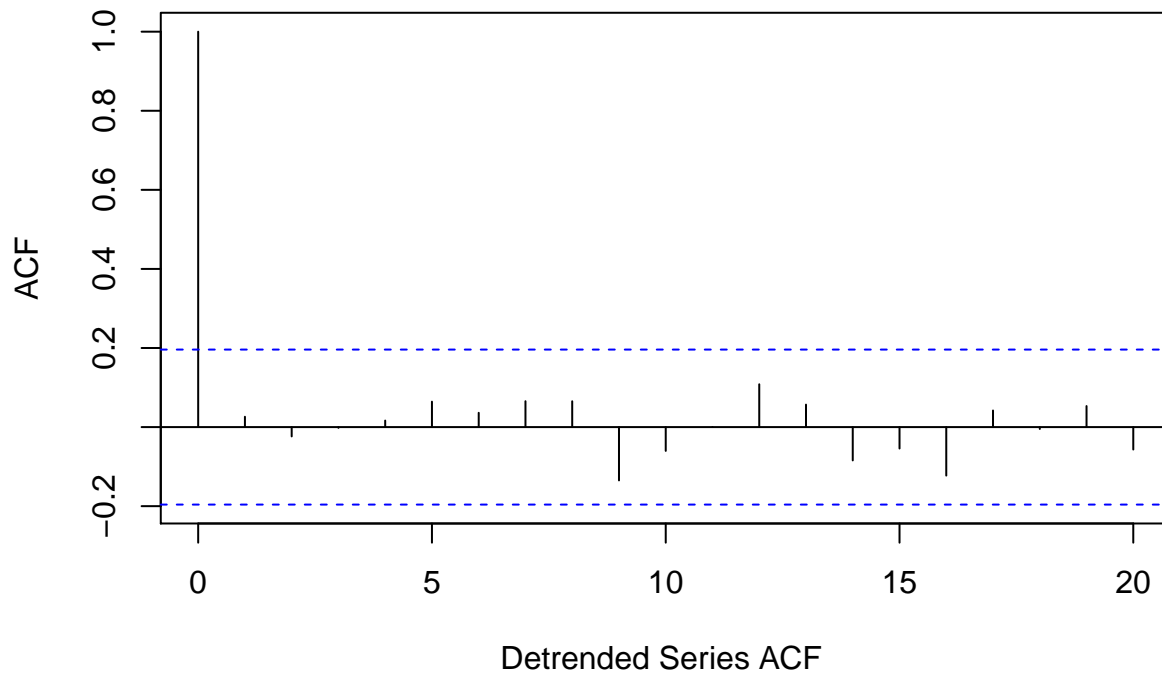


```

acf(as.ts(detrend.x), xlab = "Detrended Series ACF ")

```

Series as.ts(detrend.x)



Part III

Visual Inspection of the below stock exchanges suggest positive covariance and the presence of a positive drift parameter over time.

#PART III

#Read in our data and plot our series

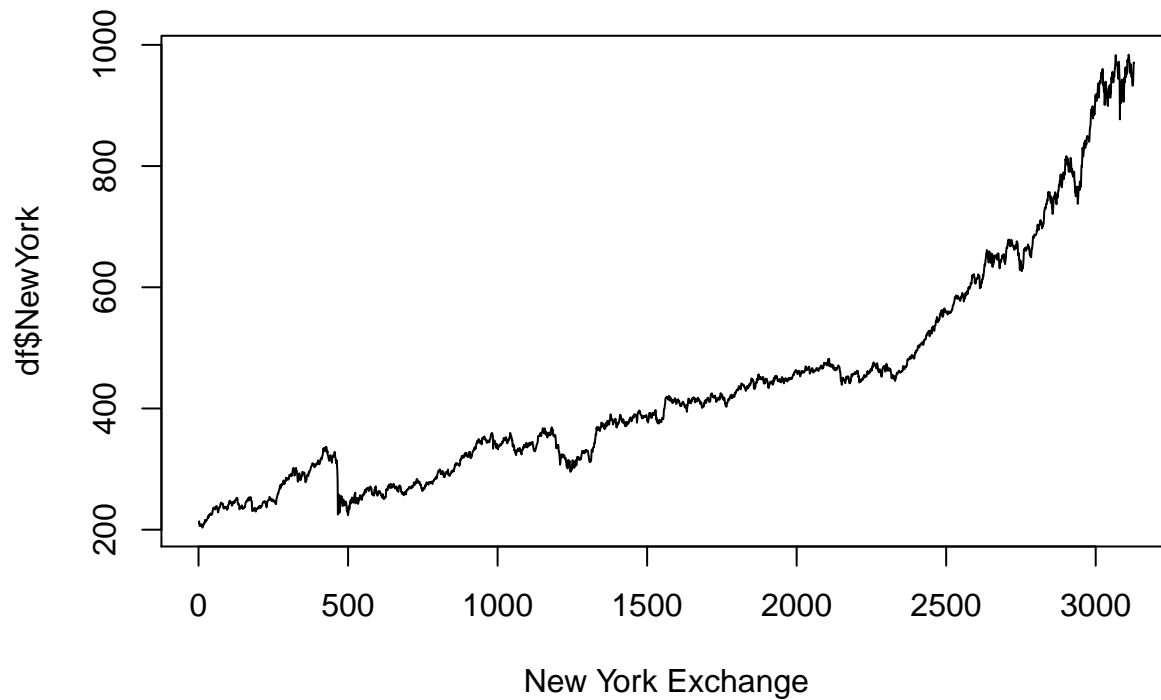
```
df <- read.delim("stockmarket.dat", sep = "")
```

```
df.ts <- as.ts(df)
```

#layout(1:2)

#par(mar=c(1,1,1,1))

```
plot(df$NewYork, xlab = "New York Exchange", type = "l")
```



```
plot(df$London, xlab = "London Exchange", type = "l")
```



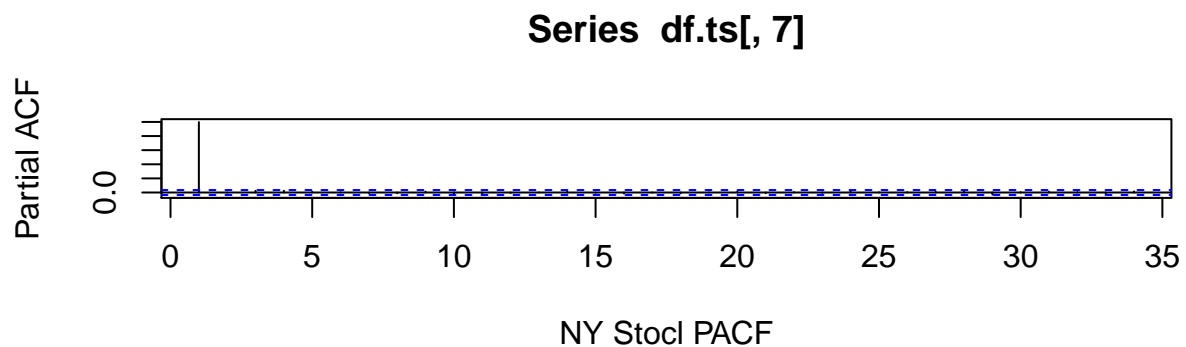
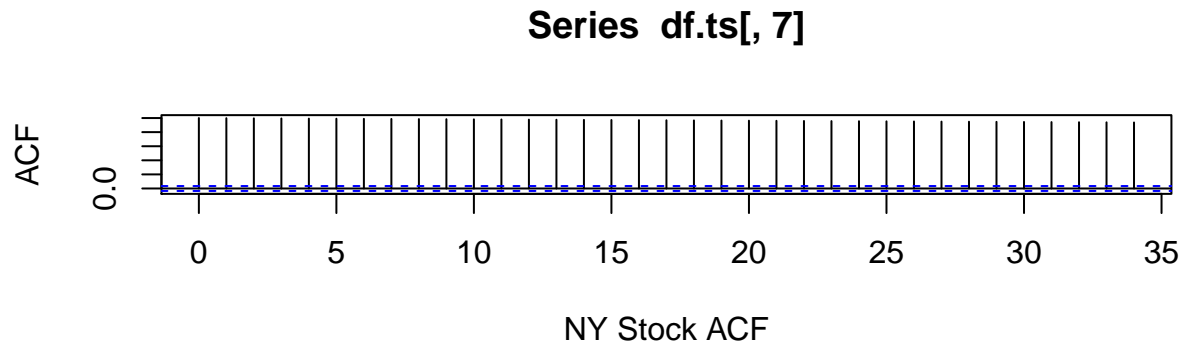
For both series, ACF, PACF, Augmented Dickey Fuller, and Phillips-Perron tests suggest that a unit root is present. As such, neither series can be considered stationary.

```
#ACF/PACF Plots suggest non-stationarity
#ADF Tests and PP tests suggest the presence of no unit root in the NY series, but not the London series

library("tseries")

layout(1:2)
```

```
acf(df.ts[,7], xlab = "NY Stock ACF") ; pacf(df.ts[,7], xlab = "NY Stock PACF");
```



```
adf.test(df.ts[,7]) ; PP.test(df.ts[,7])
```

```
## Warning in adf.test(df.ts[, 7]): p-value greater than printed p-value
```

```
##
```

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

```
##
```

```
## data: df.ts[, 7]
```

```
## Dickey-Fuller = 0.30744, Lag order = 14, p-value = 0.99
```

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

```
##
```

```
## Phillips-Perron Unit Root Test
```

```
##
```

```
## data: df.ts[, 7]
```

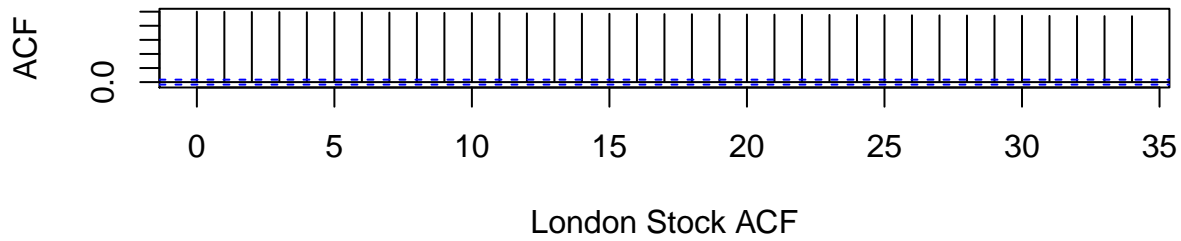
```
## Dickey-Fuller = 0.42795, Truncation lag parameter = 9, p-value =
```

```
## 0.99
```

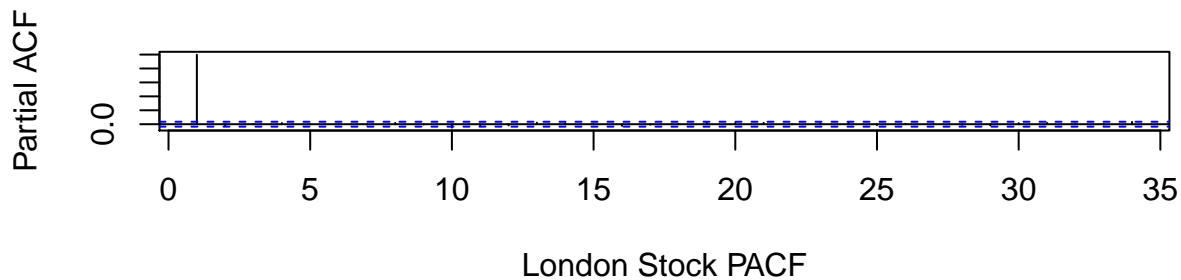
```
layout(1:2)
```

```
acf(df.ts[,3], xlab = "London Stock ACF") ; pacf(df.ts[,3], xlab = "London Stock PACF") ; adf.test(df.ts[,3]) ; PP.test(df.ts[,3])
```


Series df.ts[, 3]



Series df.ts[, 3]



```
##
## Augmented Dickey-Fuller Test
##
## data: df.ts[, 3]
## Dickey-Fuller = -1.641, Lag order = 14, p-value = 0.7303
## alternative hypothesis: stationary
##
## Phillips-Perron Unit Root Test
##
## data: df.ts[, 3]
## Dickey-Fuller = -1.3647, Truncation lag parameter = 9, p-value =
## 0.8473
```

Fitting a VAR(1) model to the two series results in the below output. London stock values, denoted as x_t , appear to positively correlated with the first lag of the New York stock values, as shown by the positive y_{t-1} coefficient in the first VAR equation. As expected by the unit-root tests above, London stock values are positively correlated with its first lag, with a near 1 value for the lagged coefficient.

New York stock exchange values appear to be weakly negatively correlated with the first lag of the London Stock Exchange. This is denoted by the -0.0003 value for the x_{t-1} coefficient in the second VAR equation below. Like the London Exchange, the New York stock values are correlated with their first lag, with a near 1 coefficient value.

```
#VAR(1) Model
library("vars")
```

```
## Loading required package: MASS
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##   select
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##   as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
x <- df.ts[,3]; y<- df.ts[,7]
```

```
var1 <- VAR(cbind(x,y), p=1)
```

```
var1
```

```
##
```

```
## VAR Estimation Results:
```

```
## =====
```

```
##
```

```
## Estimated coefficients for equation x:
```

```
## =====
```

```
## Call:
```

```
## x = x.l1 + y.l1 + const
```

```
##
```

```
##      x.l1      y.l1      const
```

```
## 0.98541813 0.07494087 8.38623196
```

```
##
```

```
##
```

```
## Estimated coefficients for equation y:
```

```
## =====
```

```
## Call:
```

```
## y = x.l1 + y.l1 + const
```

```
##
```

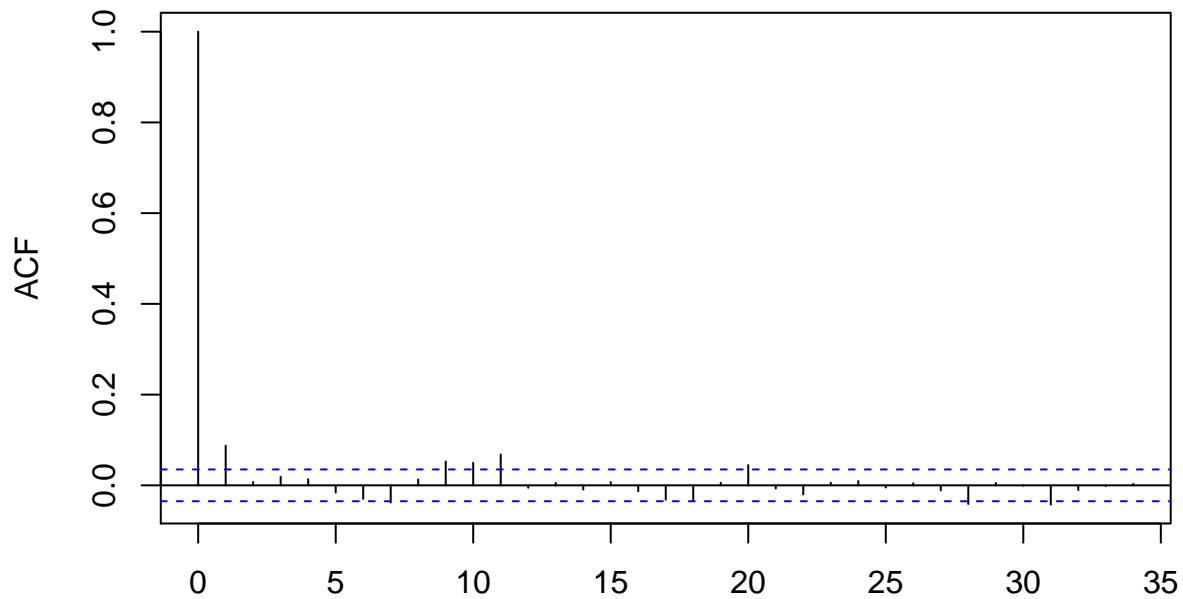
```
##      x.l1      y.l1      const
```

```
## -0.000344888 1.0025400926 0.0782517887
```

Residual plots of the VAR model show more significant lags than expected at the 95% confidence level. This suggests that the VAR (1) model may not be the best fit.

```
acf(resid(var1)[, 1], xlab = "VAR(1): London VAR Residual ACF ")
```

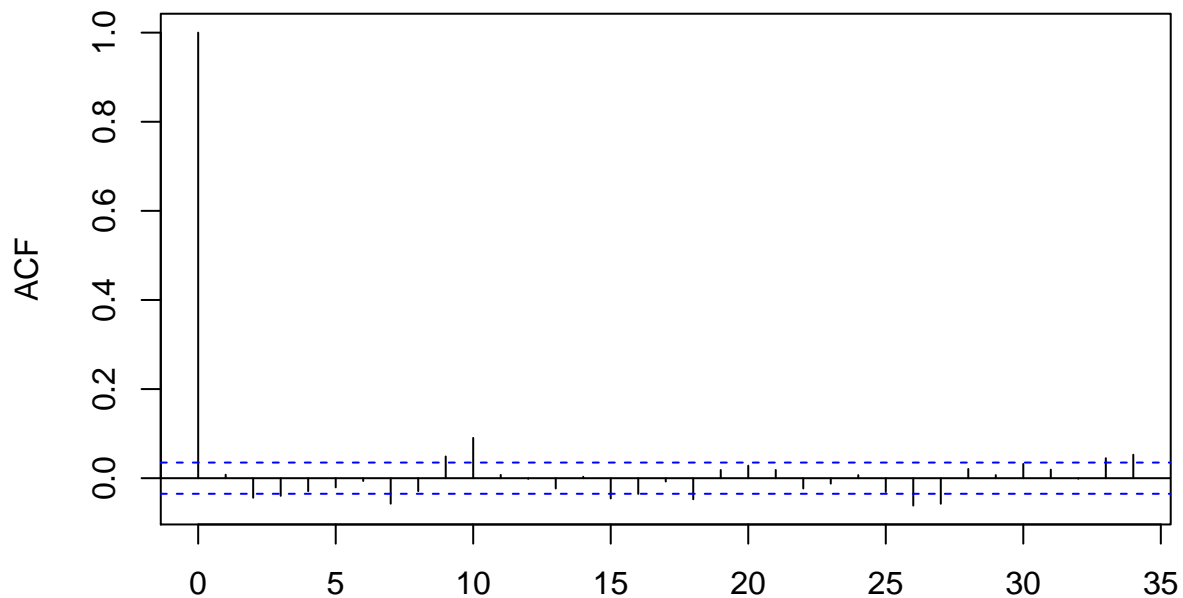
Series resid(var1)[, 1]



VAR(1): London VAR Residual ACF

```
acf(resid(var1)[, 2], xlab = "VAR(1): NY VAR Residual ACF")
```

Series resid(var1)[, 2]



VAR(1): NY VAR Residual ACF

The default VAR function in R selected a 12th order model as the optimal fit. This decision is supported by the lower AIC value associated with the VAR(12) fit. However, the residuals for the VAR(12) model appear to have significantly more auto-correlation than expected at the 95% confidence interval. This raises

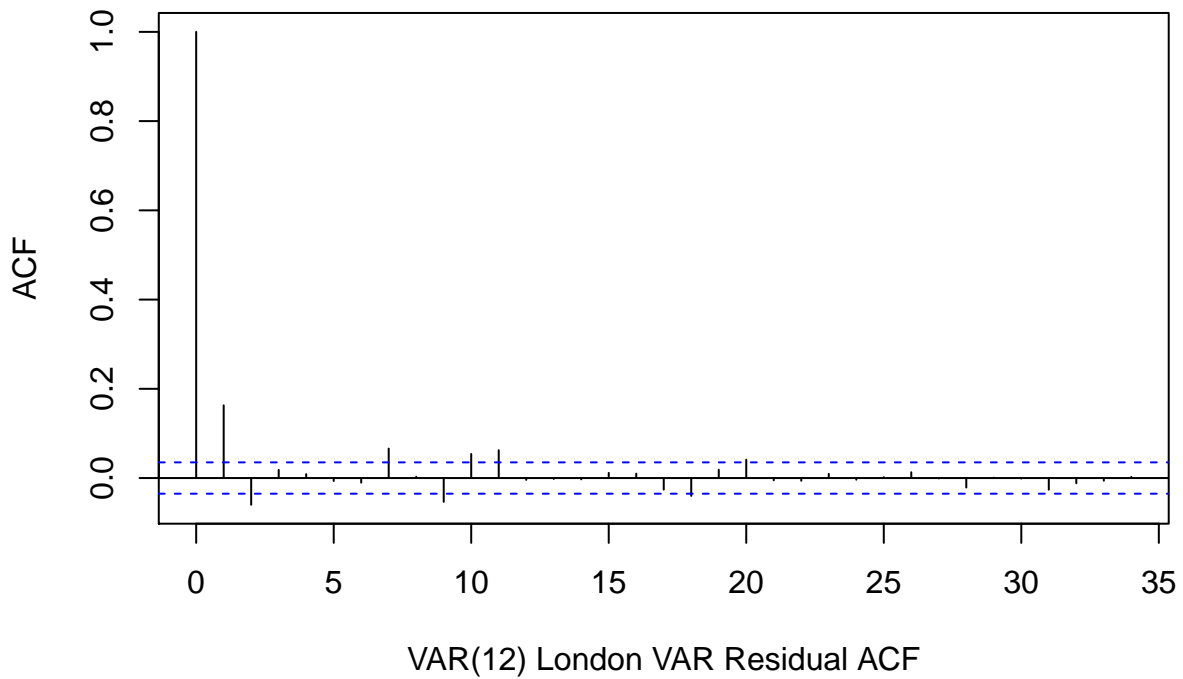
uncertainty as to whether a VAR model is appropriate for the two series.

```
var.ar <- ar(cbind(x,y))
var.ar
```

```
##
## Call:
## ar(x = cbind(x, y))
##
## $ar
## , , 1
##
##      x      y
## x 0.8677918 0.8361
## y -0.0006525 1.0045
##
## , , 2
##
##      x      y
## x 0.23615 -1.3822
## y 0.02166 -0.1171
##
## , , 3
##
##      x      y
## x -0.10627 0.45125
## y -0.02469 0.09328
##
## , , 4
##
##      x      y
## x 0.02490 -0.01499
## y 0.01289 -0.02990
##
## , , 5
##
##      x      y
## x -0.055517 0.27298
## y -0.004773 0.02476
##
## , , 6
##
##      x      y
## x -0.015044 0.08468
## y -0.009737 0.05243
##
## , , 7
##
##      x      y
## x -0.11163 0.5108
## y -0.02353 0.1049
##
## , , 8
##
##      x      y
```

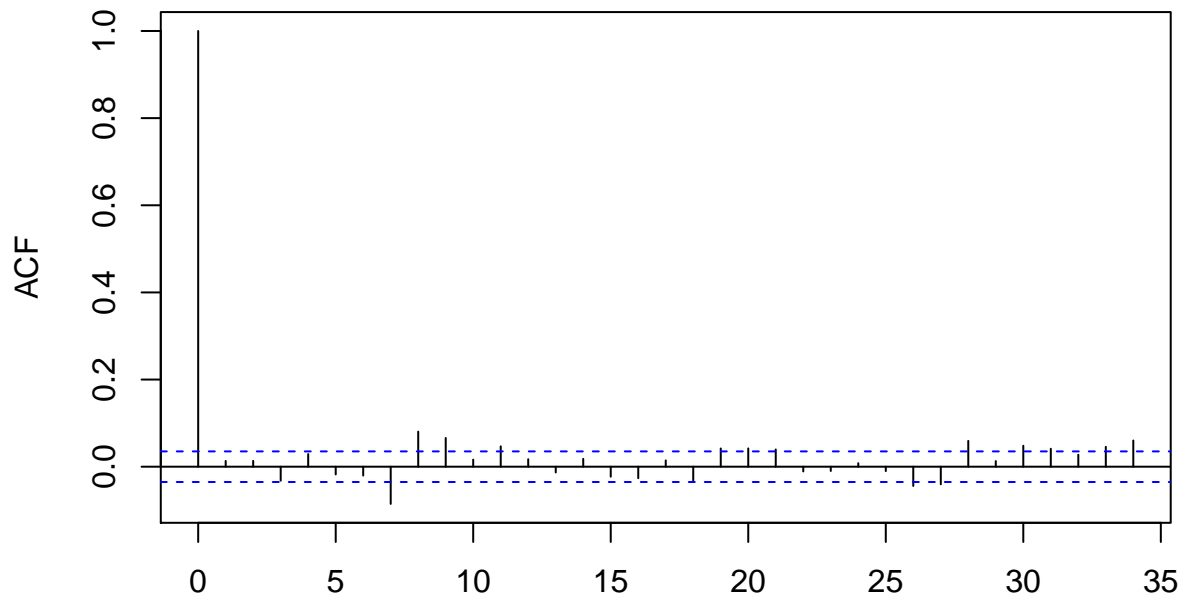
```
## x 0.20155 -0.9281
## y 0.04472 -0.2059
##
## , , 9
##
##          x          y
## x 0.10385 -0.38801
## y 0.01195 -0.03443
##
## , , 10
##
##          x          y
## x -0.15976 0.7093
## y -0.04077 0.1850
##
## , , 11
##
##          x          y
## x 0.005659 -0.1204
## y 0.013620 -0.0827
##
##
## $var.pred
##          x          y
## x 2944.6 539.3
## y 539.3 124.7
acf(var.ar$res[-c(1:12), 1], xlab = "VAR(12) London VAR Residual ACF")
```

Series var.ar\$res[-c(1:12), 1]



```
acf(var.ar$res[-c(1:12), 2], xlab = "VAR(12) NY VAR Residual ACF")
```

Series var.ar\$res[-c(1:12), 2]



VAR(12) NY VAR Residual ACF

```
#AIC for both models suggest the VAR(12) model is a better fit  
AIC(var1)
```

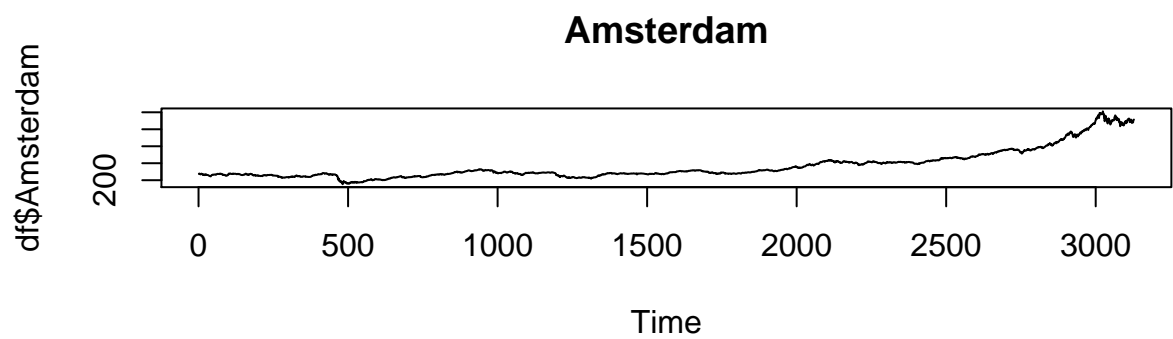
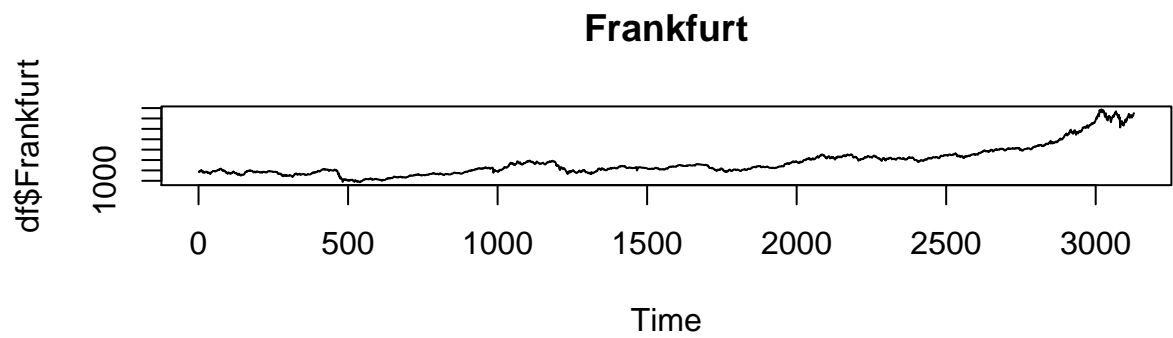
```
## [1] 46344.67
```

```
sum (var.ar$aic)
```

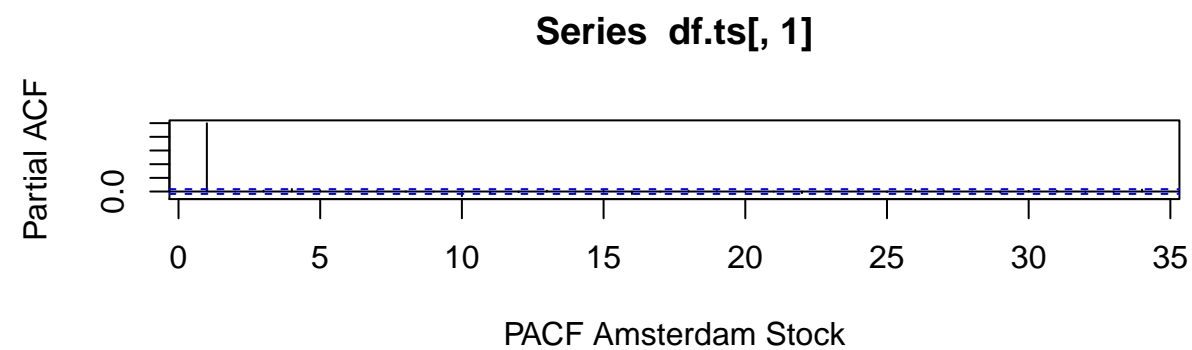
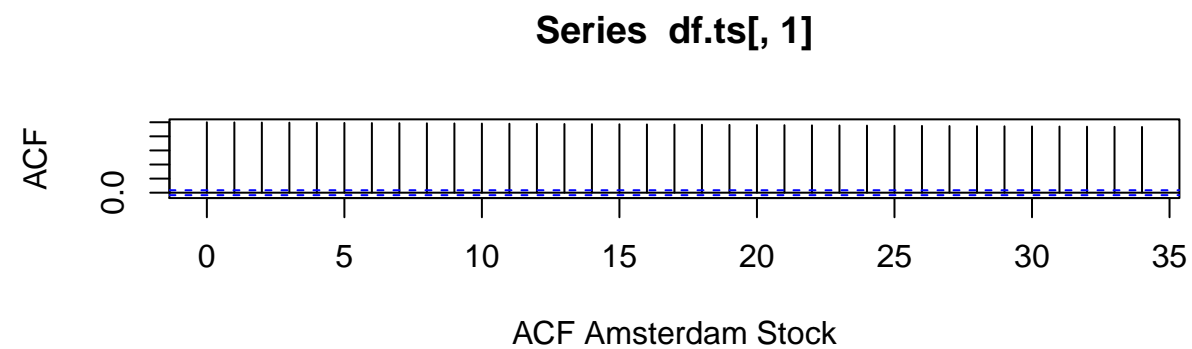
```
## [1] 29968
```

Repeating the above for the Frankfurt and Amsterdam stock series yields similar results. Like the NY and London stock series, the Frankfurt and Amsterdam series appear to be non-stationary and characterized by a unit root.

```
layout(1:2)  
plot(df$Frankfurt, main = "Frankfurt", type = "l", xlab = "Time")  
plot(df$Amsterdam, main = "Amsterdam", type = "l", xlab = "Time")
```



```
layout(1:2)
acf(df.ts[,1], xlab = "ACF Amsterdam Stock" ) ; pacf(df.ts[,1], xlab = "PACF Amsterdam Stock");
```

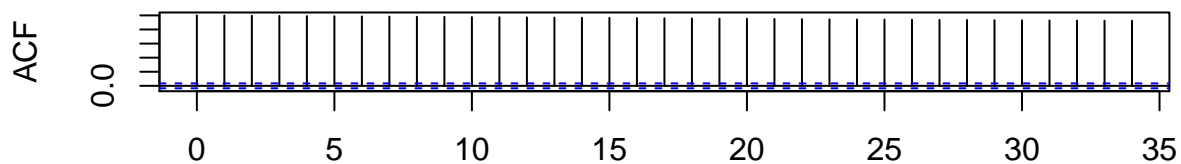


```
adf.test(df.ts[,1]) ; PP.test(df.ts[,1])
```

```
##
## Augmented Dickey-Fuller Test
##
## data: df.ts[, 1]
## Dickey-Fuller = -0.3736, Lag order = 14, p-value = 0.9876
## alternative hypothesis: stationary
##
## Phillips-Perron Unit Root Test
##
## data: df.ts[, 1]
## Dickey-Fuller = 0.27478, Truncation lag parameter = 9, p-value =
## 0.99
```

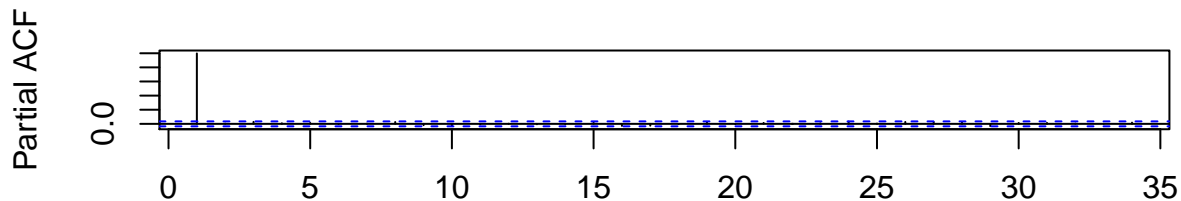
```
layout(1:2)
acf(df.ts[,2], xlab = "ACF Frankfurt Stock") ; pacf(df.ts[,2], xlab = "PACF Frankfurt Stock");
```

Series df.ts[, 2]



ACF Frankfurt Stock

Series df.ts[, 2]



PACF Frankfurt Stock

```
adf.test(df.ts[,2]); PP.test(df.ts[,2])
```

```
##
## Augmented Dickey-Fuller Test
##
## data: df.ts[, 2]
## Dickey-Fuller = -0.50187, Lag order = 14, p-value = 0.9817
## alternative hypothesis: stationary
##
## Phillips-Perron Unit Root Test
##
```



```
## data: df.ts[, 2]
## Dickey-Fuller = -0.089866, Truncation lag parameter = 9, p-value =
## 0.99
```

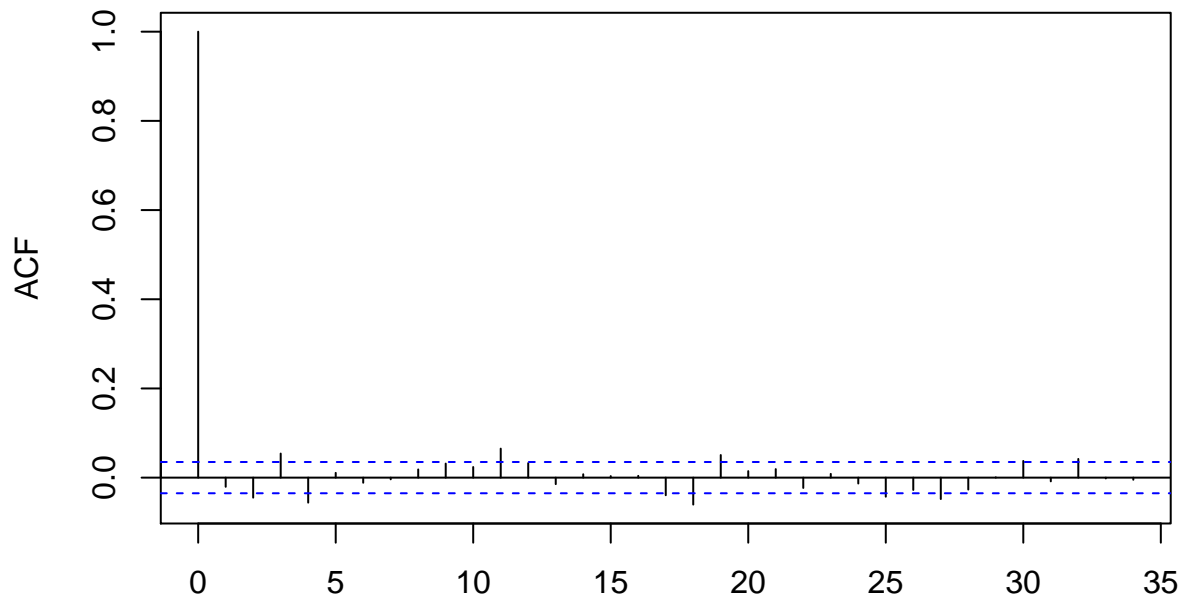
For these two series, VAR(1) model suggests that the Amsterdam stock exchange is positively correlated with the first lag of the Frankfurt stock exchange. The Frankfurt exchange however, does not appear to be correlated with the first lag of the Amsterdam exchange. Residuals plots for both series show more greater than expected auto-correlation in lags, suggesting that VAR(1) might not be appropriate on the series without transformation.

```
y <- df.ts[,1]; x<- df.ts[,2]

var1 <- VAR(cbind(x,y), p=1)
var1
```

```
##
## VAR Estimation Results:
## =====
##
## Estimated coefficients for equation x:
## =====
## Call:
## x = x.l1 + y.l1 + const
##
##      x.l1      y.l1      const
## 0.98153105 0.08025206 6.20144336
##
##
## Estimated coefficients for equation y:
## =====
## Call:
## y = x.l1 + y.l1 + const
##
##      x.l1      y.l1      const
## -0.001749018 1.008129550 0.510731188
##
## acf(resid(var1)[, 1], xlab = "ACF: VAR(1) Residuals Amsterdam Stock")
```

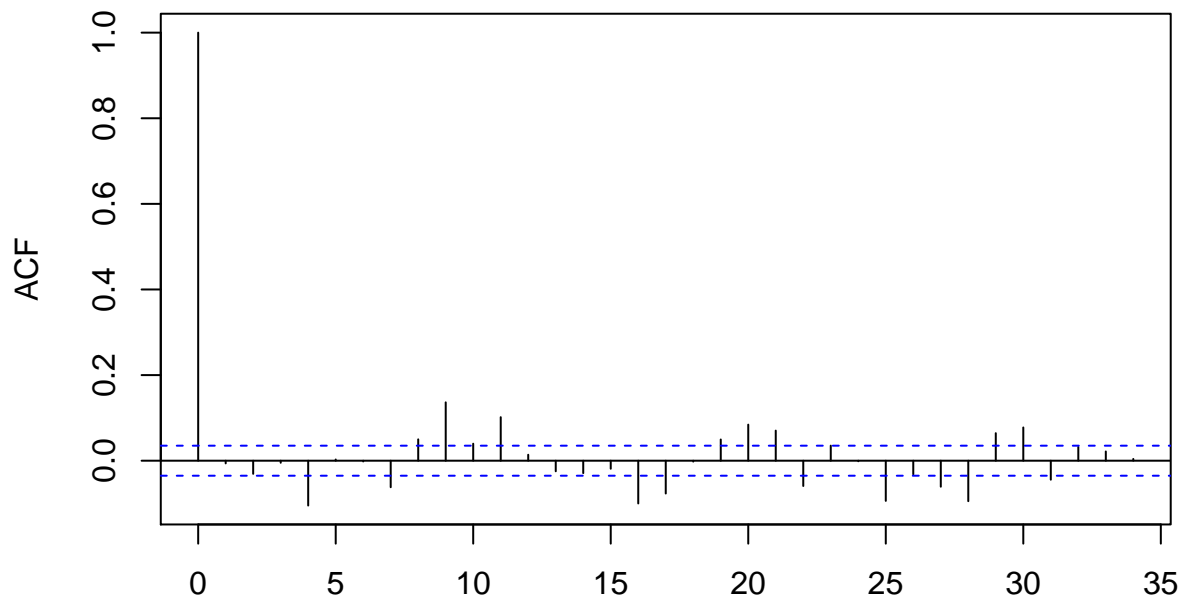
Series resid(var1)[, 1]



ACF: VAR(1) Residuals Amsterdam Stock

```
acf(resid(var1)[, 2], xlab= "ACF: VAR(1) Residuals Frankfurt Stock")
```

Series resid(var1)[, 2]



ACF: VAR(1) Residuals Frankfurt Stock

The residual plots for the VAR(12), the default choice for R's `ar` function, likewise show high degrees of auto-correlation. AIC values however suggest that the VAR(12) is a better fit for a VAR including Amsterdam and Frankfurt stock series.

```
var.ar <- ar(cbind(x,y))
var.ar
```

```
##
## Call:
## ar(x = cbind(x, y))
##
## $ar
## , , 1
##
##      x      y
## x  0.77375 1.068
## y -0.02174 1.090
##
## , , 2
##
##      x      y
## x  0.063508 -0.428056
## y -0.005345  0.009233
##
## , , 3
##
##      x      y
## x  0.21885 -0.9692
## y  0.05401 -0.2480
##
## , , 4
##
##      x      y
## x -0.07176 0.3153
## y -0.02400 0.1145
##
## , , 5
##
##      x      y
## x -0.030340 0.20341
## y -0.002358 0.02992
##
## , , 6
##
##      x      y
## x -0.0022427 -0.010569
## y  0.0003293 -0.004098
##
## , , 7
##
##      x      y
## x -0.19658 0.8795
## y -0.04416 0.1919
##
## , , 8
##
##      x      y
## x  0.35704 -1.5573
```

```

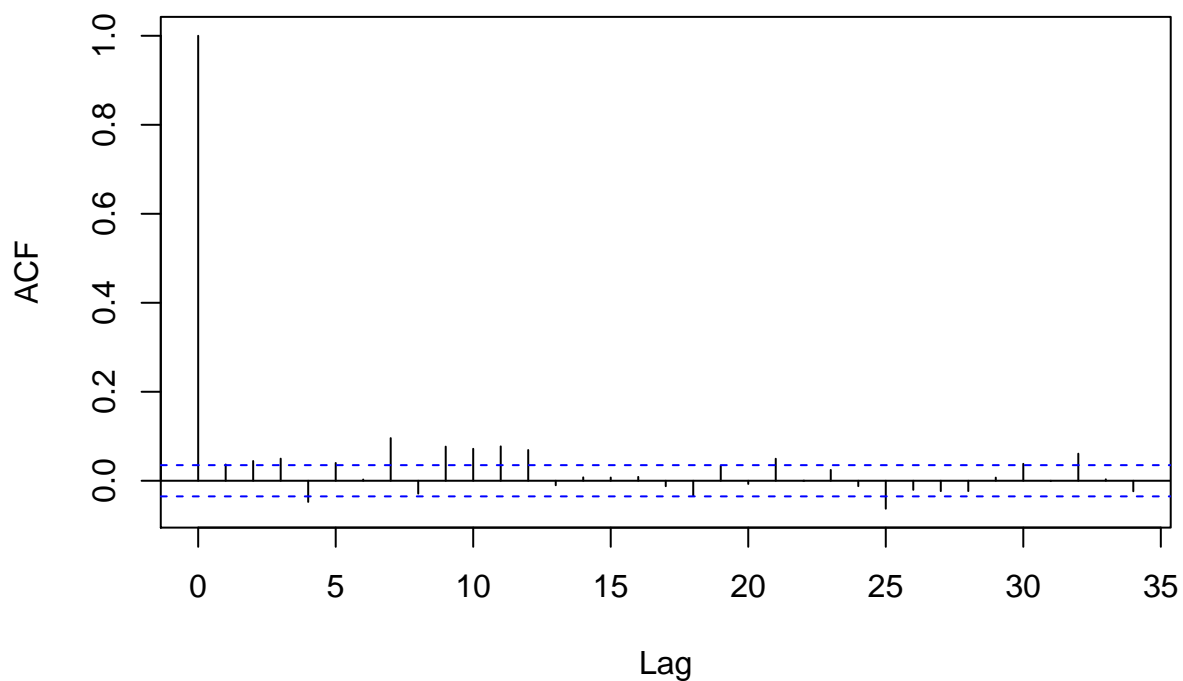
## y 0.06985 -0.2918
##
## , , 9
##
##      x      y
## x -0.2517 1.2740
## y -0.0586 0.2996
##
## , , 10
##
##      x      y
## x  0.007981 -0.14851
## y -0.002171 -0.02882
##
## , , 11
##
##      x      y
## x  0.040338 -0.221001
## y -0.000698 -0.008778
##
## , , 12
##
##      x      y
## x 0.06893 -0.3239
## y 0.03149 -0.1417
##
## $var.pred
##      x      y
## x 2403.1 492.2
## y 492.2 117.2
var.ar$ar
## , , x
##
##      x      y
## 1  0.773753501 -0.0217399950
## 2  0.063507831 -0.0053451933
## 3  0.218851839  0.0540134547
## 4 -0.071757133 -0.0239983331
## 5 -0.030339655 -0.0023576728
## 6 -0.002242705  0.0003293084
## 7 -0.196577752 -0.0441569006
## 8  0.357038568  0.0698534939
## 9 -0.251675881 -0.0585992087
## 10 0.007981210 -0.0021708015
## 11 0.040337826 -0.0006980161
## 12 0.068930206  0.0314866834
##
## , , y
##
##      x      y
## 1  1.06789516  1.089503583
## 2 -0.42805556  0.009232761

```

```
## 3 -0.96915327 -0.247966780
## 4  0.31527070  0.114548381
## 5  0.20340681  0.029917366
## 6 -0.01056903 -0.004097781
## 7  0.87952856  0.191906361
## 8 -1.55732127 -0.291823772
## 9  1.27401587  0.299626015
## 10 -0.14851183 -0.028822073
## 11 -0.22100076 -0.008777965
## 12 -0.32385896 -0.141745659
```

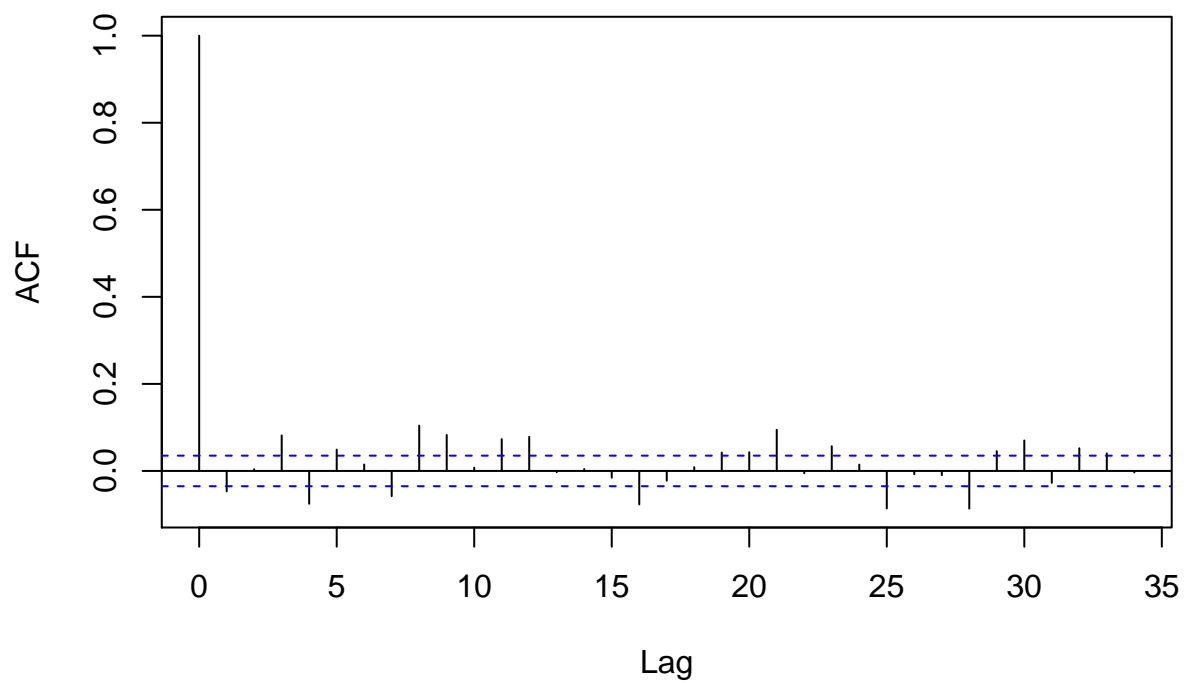
```
acf(var.ar$res[-c(1:12)], 1, main = "ACF: VAR(12) Residuals Amsterdam Stock")
```

ACF: VAR(12) Residuals Amsterdam Stock



```
acf(var.ar$res[-c(1:12)], 2, main = "ACF: VAR(12) Residuals Frankfurt Stock")
```

ACF: VAR(12) Residuals Frankfurt Stock



```
AIC(var1)
```

```
## [1] 45592.75
```

```
sum (var.ar$aic)
```

```
## [1] 29805.06
```