Homework Two for Econometrics

Kristopher C. Toll

January 31, 2018

# Application

## Problem 1

### part a

# Create the time trend  
  
# pass Johnson and Johnson into it own vector  
  
a <- jj  
  
# Creating the data set  
trend <- time(jj)  
q <- factor(rep(1:4, 21))  
ln.jj <- log(jj)  
  
# Without the intercept  
reg.jj <- lm(ln.jj ~ -1 + trend + q, na.action = NULL)  
summary(reg.jj)

##   
## Call:  
## lm(formula = ln.jj ~ -1 + trend + q, na.action = NULL)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.29318 -0.09062 -0.01180 0.08460 0.27644   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## trend 0.167172 0.002259 74.00 <0.0000000000000002 \*\*\*  
## q1 -328.276371 4.450537 -73.76 <0.0000000000000002 \*\*\*  
## q2 -328.248248 4.451102 -73.75 <0.0000000000000002 \*\*\*  
## q3 -328.178140 4.451666 -73.72 <0.0000000000000002 \*\*\*  
## q4 -328.446897 4.452231 -73.77 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1254 on 79 degrees of freedom  
## Multiple R-squared: 0.9935, Adjusted R-squared: 0.9931   
## F-statistic: 2407 on 5 and 79 DF, p-value: < 0.00000000000000022

Looking at the results of the output we can see that without an intercept term we are able to determine what the effects of each quarter are. It seems that in each quarter, there will be a decrease of 328 dollars in each quarter. The trend variable means that as each time period moves up by one, we have an average increase of 0.167. However, by removing the intercept term and including each quarter in the model, we have colinear problem and thus the estimates cannot be taken seriously. The intercept is passed into each quarter.

### Part b

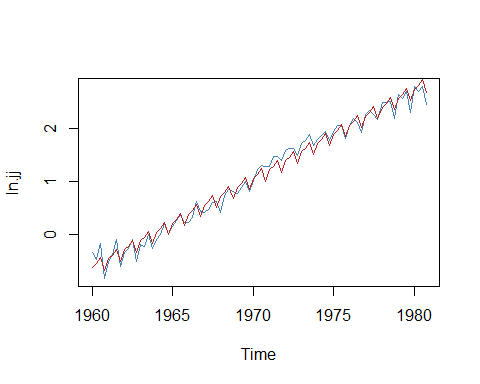
# With the intercept  
reg.jj2 <- lm(ln.jj ~ trend + q, na.action = NULL)  
summary(reg.jj2)

##   
## Call:  
## lm(formula = ln.jj ~ trend + q, na.action = NULL)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.29318 -0.09062 -0.01180 0.08460 0.27644   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -328.276371 4.450537 -73.761 < 0.0000000000000002 \*\*\*  
## trend 0.167172 0.002259 73.999 < 0.0000000000000002 \*\*\*  
## q2 0.028123 0.038696 0.727 0.4695   
## q3 0.098231 0.038708 2.538 0.0131 \*   
## q4 -0.170527 0.038729 -4.403 0.0000331 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1254 on 79 degrees of freedom  
## Multiple R-squared: 0.9859, Adjusted R-squared: 0.9852   
## F-statistic: 1379 on 4 and 79 DF, p-value: < 0.00000000000000022

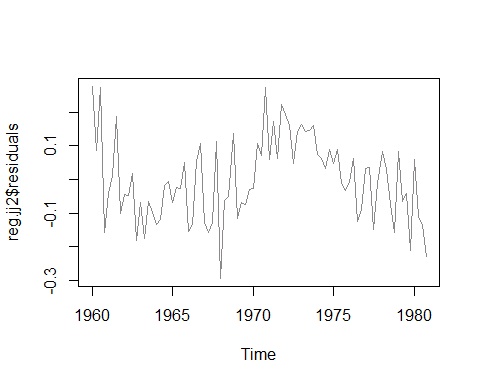
By including the intercept parameter, we can now get a more precise estimate for each quarter. The first quarter is part of the intercept term and now the interpretations of each coefficient estiamte are acurate.

### c

fitted.ts <- as.ts(reg.jj2)  
plot(ln.jj, col="steelblue")  
lines(fitted(reg.jj2), col="firebrick")



plot(reg.jj2$residuals, col="snow4") # I picked snow4 because it is white just like white noise

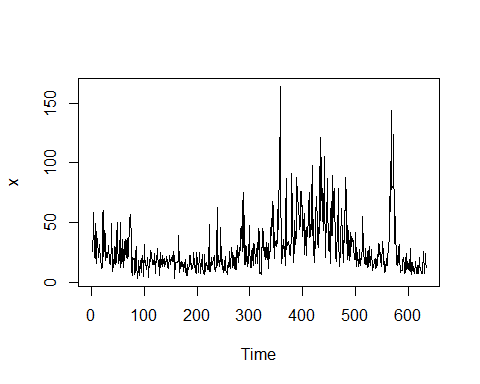


The model with the intercept fits the model very and well. The adjusted r squared value is .9852 and the ploted residuals do appear to be random which sugjests white noise.

## Problem 2

### Part a

x <- varve  
plot(x)



There do seem to be some trends from year to year. Which sugjests nonstationarity but we can't verify from this plot alone.

### Part b

half <- .5\*length(x)  
x.1 <- x[1:half]  
varx.1 <- var(x.1)  
  
x.2 <- x[(half+1):length(x)]  
varx.2 <- var(x.2)  
  
print(c(varx.1, "variance of first half"))

## [1] "133.457415667053" "variance of first half"

print(c(varx.2, "variance of second half"))

## [1] "594.490438823224" "variance of second half"

We can see that there is a difference between the variance of the first half of the series and the second half.

### Part c

y <- log(x)  
y.1 <- y[1:half]  
y.2 <- y[(half+1):length(y)]  
  
print(c(var(y.1), "variance of first half"))

## [1] "0.270721652653357" "variance of first half"

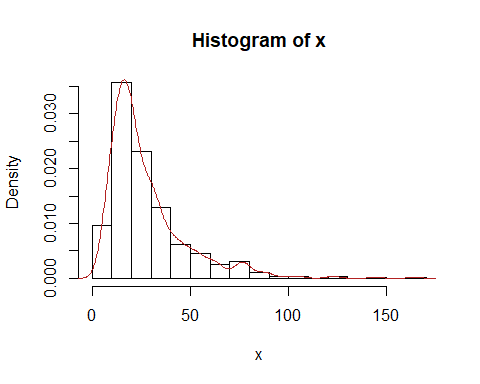
print(c(var(y.2), "variance of second half"))

## [1] "0.451371011716303" "variance of second half"

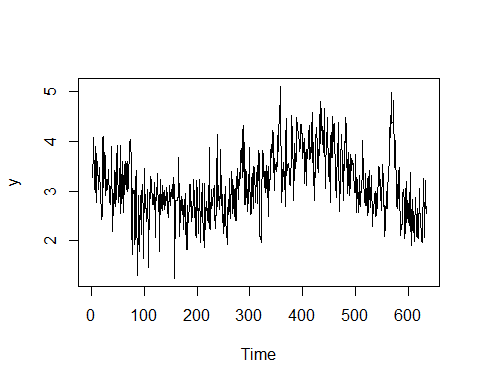
Here we can see that the differenc ein variance between the first half and second half are not as exagerated and this series is more stable.

### Part d

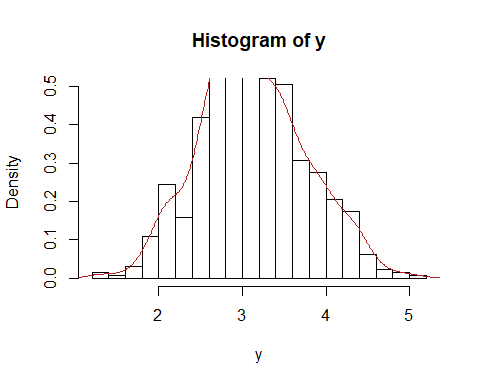
hist(x, probability = TRUE, breaks = 20, ylim = c(0, 0.035))  
lines(density(x), col = "firebrick") # I overlaid the kernel on the histogram



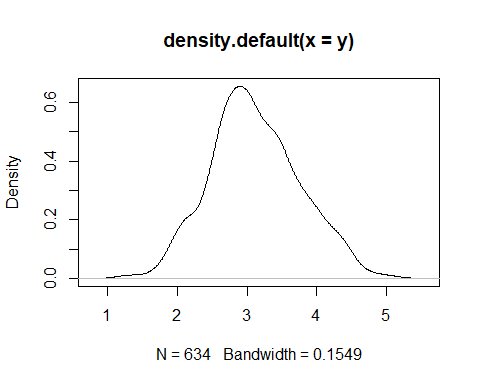
plot(y)



hist(y, probability = TRUE, breaks = 20, ylim = c(0, 0.5))  
lines(density(y), col = "firebrick") # I overlaid the kernel on the histogram

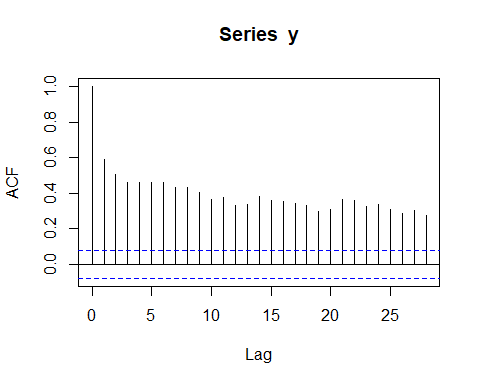


plot(density(y))

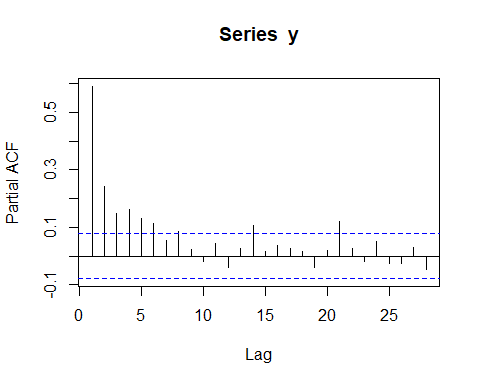
 The log transformation does a little to improve normaility, however, it isn't perfect

### Part e

acf(y)

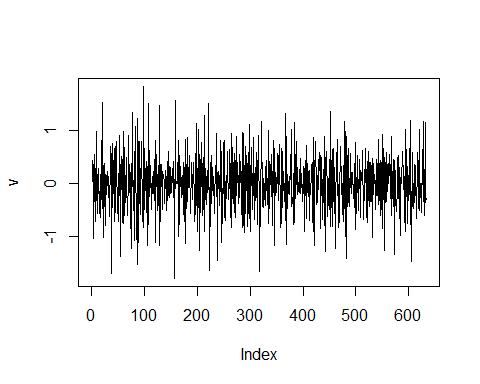


pacf(y)

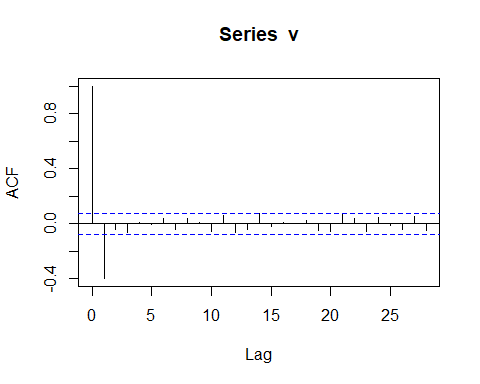


By examining the acf we notice that there is a steady decrease in autocavariance

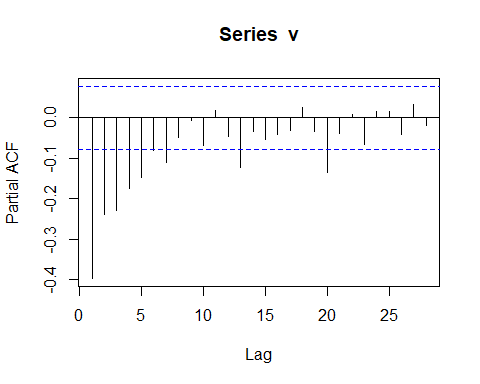
t <- 2:length(y)  
v <- (y[t] - y[t-1])  
plot(v, type = "l")



acf(v)



pacf(v)



From the plots above we can now see that this process is not generated by random walk and is now weakly stationary

### Part g

In examinging the plots we see that acf cuts off after one lag and partial acf seems to decay expontentialy. This sugjests an MA(1) process.