FA_Assignment4_Duo_Zhou

Duo Zhou

7/18/2020

This assignment helps understanding stationarity and seasonality of linear models for time series

Exercise 7 on page 126 of the Textbook

Consider the quarterly earnings per share of Johnson & Johnson from the first quarter of 1992 to the second quarter of 2011. The data are in the file q-jnj-earns-9211.txt available on the textbook web page. Take log transformation if necessary. Build a time series model for the data. Perform model checking to assess the adequacy of the fitted model. Write down the model. Refit the model using data from 1992 to 2008. Perform 1-step to 10-step forecasts of quarterly earnings and obtain a forecast plot.

Loading and Examining the data

```
datapath <- "C:/Users/zd000/Desktop/MSCA/Financial Analytics/Assignments/week4/"
da=read.table(file=paste(datapath,"q-jnj-earns-9211.txt" ,sep="/"),header=T)
head(da)
```

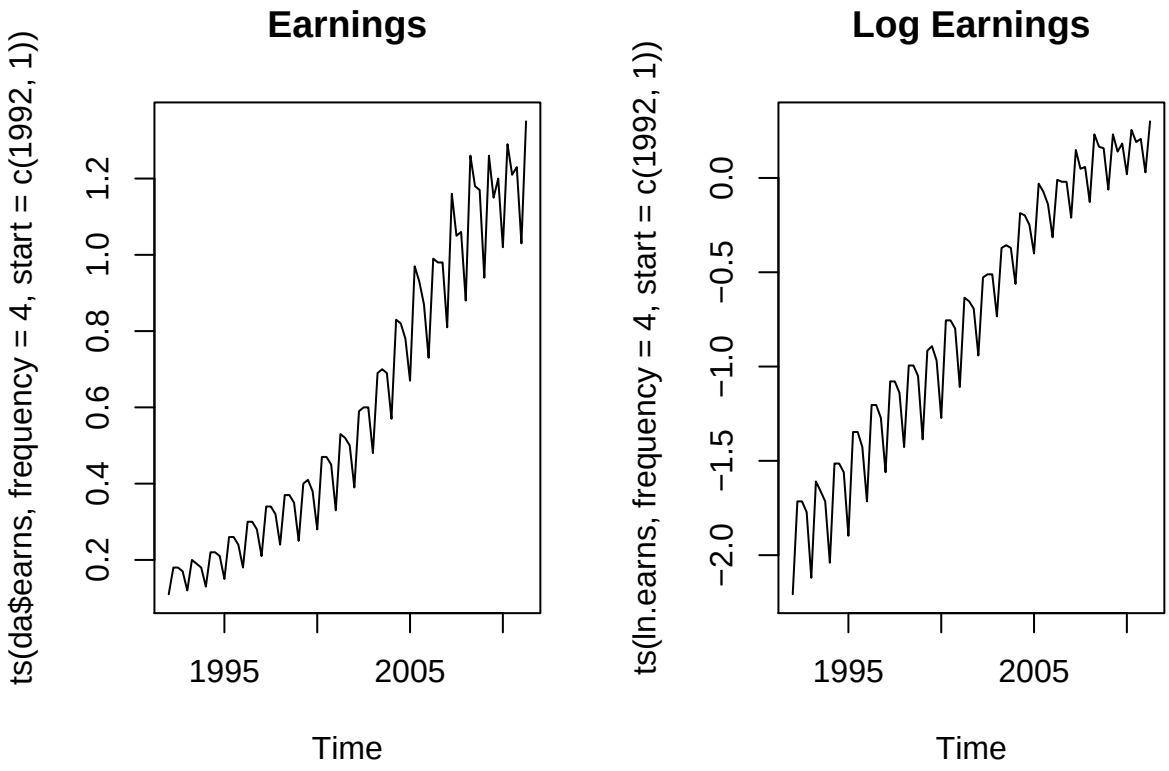
```
##   day mon year earns
## 1  30   1 1992  0.11
## 2  23   4 1992  0.18
## 3  21   7 1992  0.18
## 4  20  10 1992  0.17
## 5   1   2 1993  0.12
## 6  29   4 1993  0.20
```

```
dim(da) # 78 x 4
```

```
## [1] 78  4
```

Take log transformation if necessary.

```
ln.earns <- log(da$earns)
par(mfrow=c(1,2))
plot(ts(da$earns,frequency = 4, start=c(1992,1)), type="l", main="Earnings")
plot(ts(ln.earns,frequency = 4, start=c(1992,1)), type="l", main="Log Earnings")
```



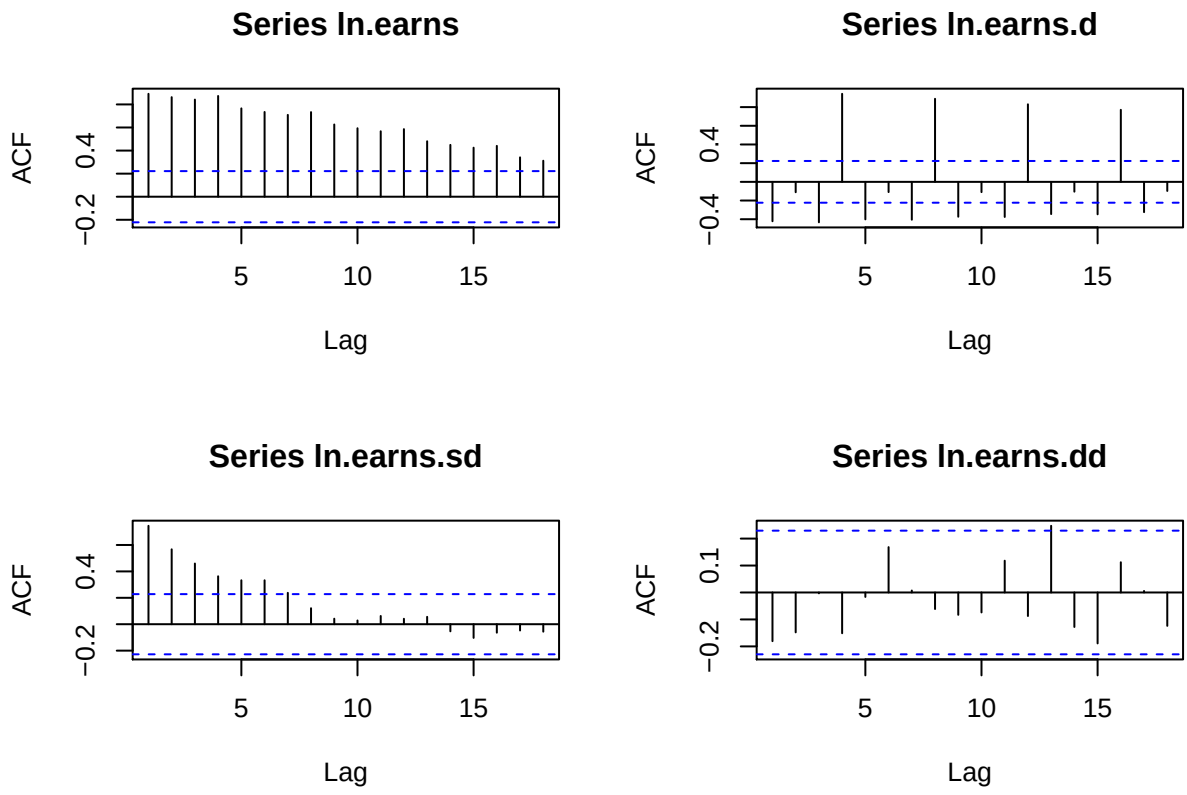
Using log transformation seems to stablize the variance and smooth out(straighten) the slope of the data.

Create a time series model for the data

From the data we can see the seasonality every quater(every 4th lag).

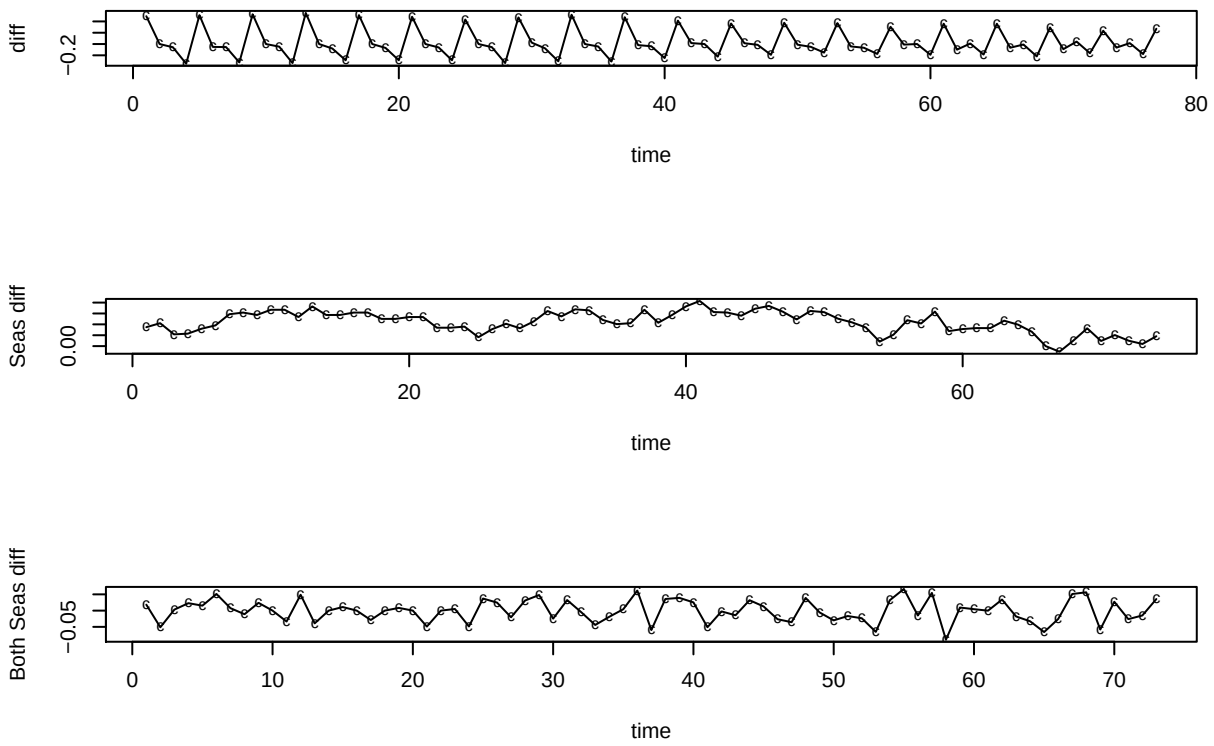
```
# Take the difference of the log transformed earnings
ln.earns.d <- diff(ln.earns)
# add 4 quarter seasonality
ln.earns.sd <- diff(ln.earns, 4)
# combine seasonal and regular differences
ln.earns.dd <- diff(ln.earns.sd )


# Look at acf plots
par(mfrow=c(2,2))
acf(ln.earns)
acf(ln.earns.d)
acf(ln.earns.sd)
acf(ln.earns.dd)
```



Make the time series plots after differencing.

```
# Obtain time plots
par(mfcol=c(3,1))
plot(ln.earns.d, xlab="time", ylab="diff", type="l")
points(ln.earns.d,pch="c1",cex=0.7)
plot(ln.earns.sd, xlab="time", ylab="Seas diff", type="l")
points(ln.earns.sd,pch="c1",cex=0.7)
plot(ln.earns.dd, xlab="time", ylab="Both Seas diff", type="l")
points(ln.earns.dd,pch="c1",cex=0.7)
```



Here we see that the log of earnings (ln.earns) is not stationary. Regular differencing (ln.earns.d) removed growth and stressed seasonality. The ACF of ln.earns.d is high for lags which are multiles of 4. ACF decays slowly.

Looks like seasonality at 4.

The third ACF plot, is after taking quaterly seasonality into consideration (ln.earns.sd). Here we see exponential decay in the ACF plot.

Although taking both differences (ln.earns.dd) removed both seasonality and nonstationarity, it seems to be going too far, we don't even have significant correlation at the first lag.

Let's do ADF tests on the original data and all the differencings to see which one gives the staionalty.

Identify AR order for these data using ar.

```
m1=ar(ln.earns,method='mle')
m2=ar(ln.earns.d,method='mle')
m3=ar(ln.earns.sd,method='mle')
m4=ar(ln.earns.dd,method='mle')
m1$order
```

```
## [1] 11
```

```
m2$order
```

```
## [1] 4
```

```
m3$order
```

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

```
m4$order
```

```
## [1] 2
```

ADF test with k equals to selected AR orders

```
adf.test(ln.earns,k=11,alternative="stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ln.earns
## Dickey-Fuller = -0.24874, Lag order = 11, p-value = 0.99
## alternative hypothesis: stationary
```

```
adf.test(ln.earns.d,k=4,alternative="stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ln.earns.d
## Dickey-Fuller = -2.9554, Lag order = 4, p-value = 0.1854
## alternative hypothesis: stationary
```

```
adf.test(ln.earns.sd,k=1,alternative="stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ln.earns.sd
## Dickey-Fuller = -3.0778, Lag order = 1, p-value = 0.136
## alternative hypothesis: stationary
```

```
adf.test(ln.earns.dd,k=2,alternative="stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ln.earns.dd
## Dickey-Fuller = -6.0651, Lag order = 2, p-value = 0.01
## alternative hypothesis: stationary
```

Validating number of differences with direct analysis of roots

```
# Create coefficients of the characteristic polynomial as:
p1=c(1,-m1$ar)
p2=c(1,-m2$ar)
p3=c(1,-m3$ar)
p4=c(1,-m4$ar)
#Then find its roots.
r1=polyroot(p1)
r1
```

```
##  [1] -0.003508+1.002002i -1.001392+0.000000i -0.003508-1.002002i
##  [4]  1.006118+0.028131i  0.733168+1.099753i -0.670903+1.154737i
##  [7] -0.670903-1.154737i  1.006118-0.028131i -1.512736-0.000000i
## [10]  0.733168-1.099753i  3.448843+0.000000i
```

```
r2=polyroot(p2)
r2
```

```
## [1] -0.002603+1.002761i -1.001540-0.000000i -0.002603-1.002761i
## [4]  1.321434-0.000000i
```

```
r3=polyroot(p3)
r3
```

```
## [1] 1.325133+0i
```

```
r4=polyroot(p4)
r4
```

```
## [1] -0.568636+2.23641i -0.568636-2.23641i
```

Find real and imaginary parts of the roots and their modula.

```
r1Re<-Re(r1)
r1Im<-Im(r1)
Mod(r1)
```

```
##  [1] 1.002008 1.001392 1.002008 1.006511 1.321738 1.335488 1.335488 1.006511
##  [9] 1.512736 1.321738 3.448843
```

```
r2Re<-Re(r2)
r2Im<-Im(r2)
Mod(r2)
```

```
## [1] 1.002765 1.001540 1.002765 1.321434
```

```
r3Re<-Re(r3)
r3Im<-Im(r3)
Mod(r3)
```

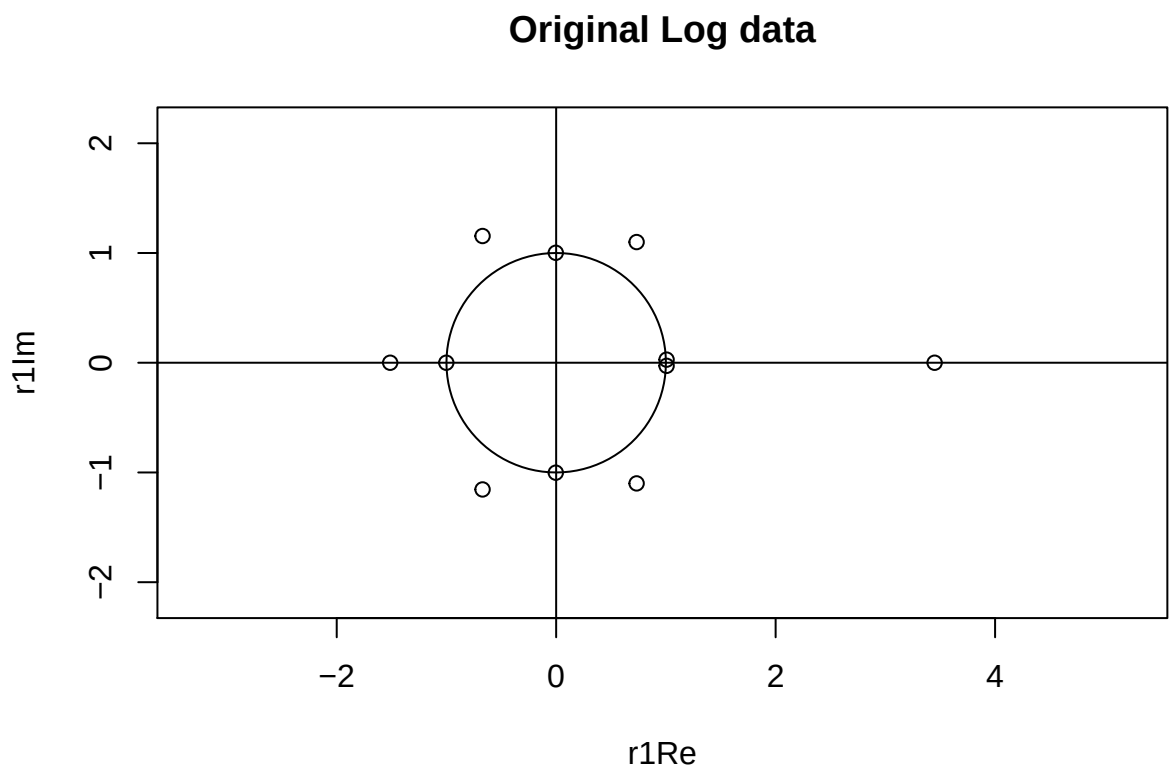
```
## [1] 1.325133
```

```
r4Re<-Re(r4)
r4Im<-Im(r4)
Mod(r4)
```

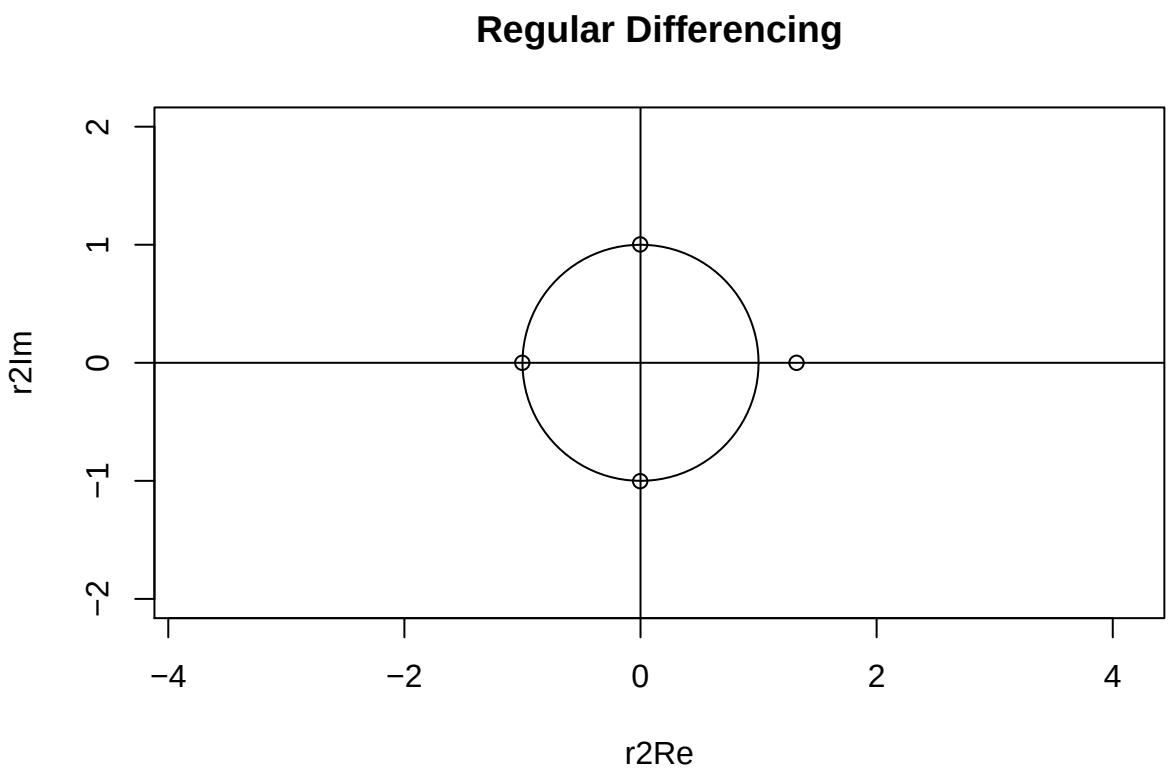
```
## [1] 2.307569 2.307569
```

Vizualize the roots and the unit circle.

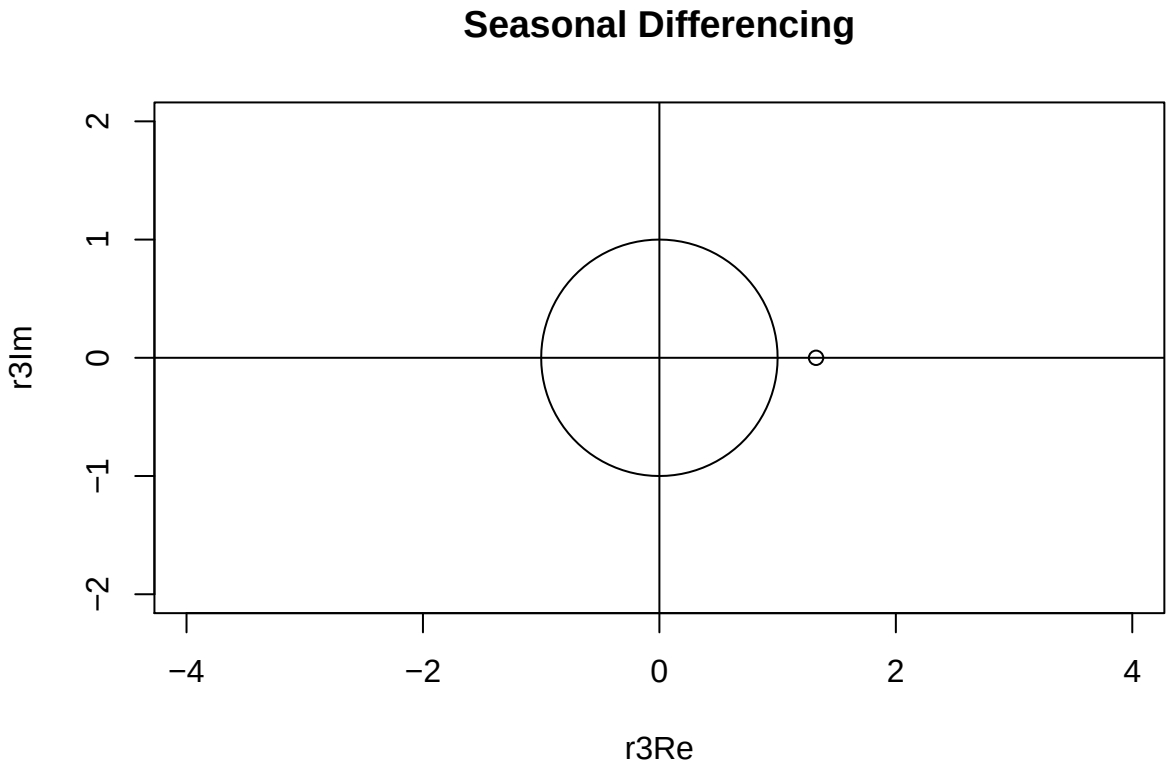
```
plot(r1Re,r1Im,asp=1,xlim=c(min(r1Re-1),max(r1Re+1)),ylim=c(min(r1Im-1),max(r1Im+1)),
     main='Original Log data')
draw.circle(0,0,radius=1)
abline(v=0)
abline(h=0)
```



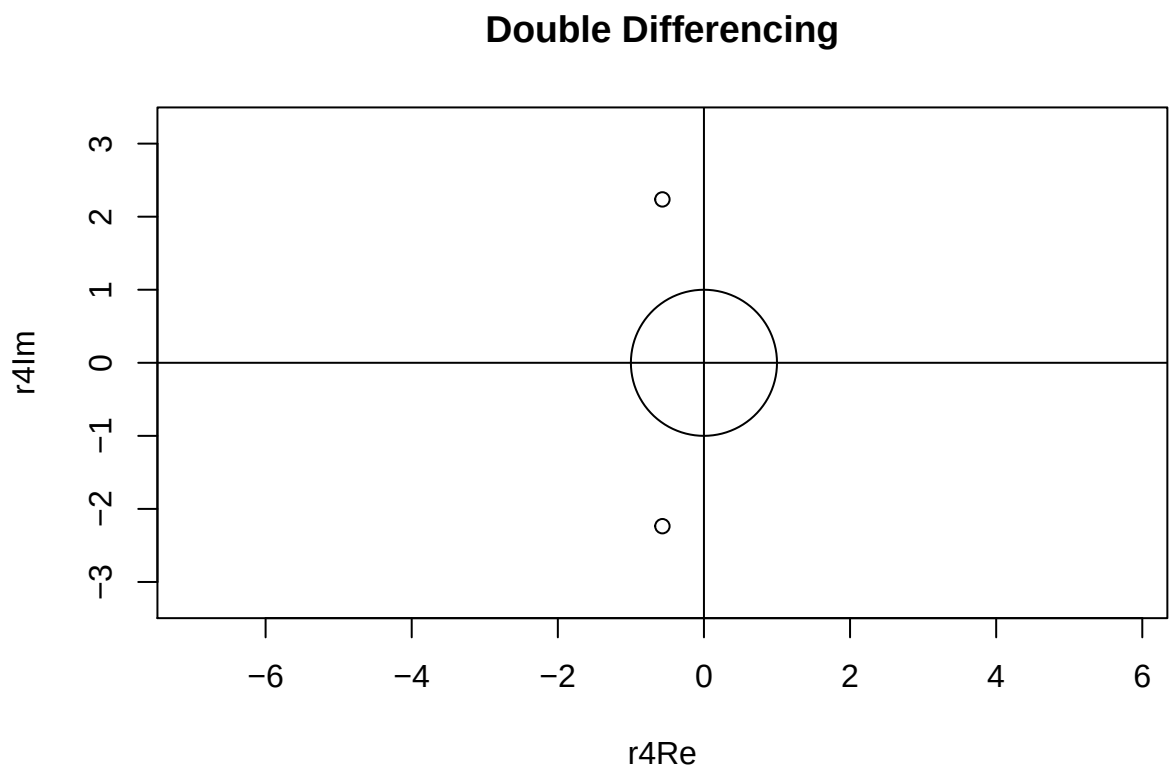

```
plot(r2Re,r2Im,asp=1,xlim=c(min(r2Re),max(r2Re)),ylim=c(min(r2Im-1),max(r2Im+1)),
     main='Regular Differencing')
draw.circle(0,0,radius=1)
abline(v=0)
abline(h=0)
```




```
plot(r3Re,r3Im,asp=1,xlim=c(-2,2),ylim=c(-2,2),
     main='Seasonal Differencing')
draw.circle(0,0,radius=1)
abline(v=0)
abline(h=0)
```




```
plot(r4Re,r4Im,asp=1,xlim=c(min(r4Re-1),max(r4Re+1)),ylim=c(min(r4Im-1),max(r4Im+1)),
     main='Double Differencing')
draw.circle(0,0,radius=1)
abline(v=0)
abline(h=0)
```



The original data is not stationary and has 4 unit roots.

Regular differencing (ln.earns.d) removed growth and has strong seasonality. The ACF for diff.ln.earns shows high positive autocorrelation at lags which are multiples of 4 (4, 8,12, etc). These data stil produces unit roots.P-value of the ADF test in this case is insignificant.

Seasonal differencing(ln.earns.sd) using 4 period cycle, seems to have removed the seasonality, from the unit look stationary. After looking at the ADF test, we cannot reject the null hypothesis of Unit root. ALthough the unit root graph shows no root close to the unit circle, the actual mod of the root is 1.32, which is still close enough to 1 with selected k=1 under the Augmented Dickey Fuller test. P-value of the ADF test in this case is 0.13.

Double differencing (ln.earns.dd) using the first difference of the seasonal difference removed seasonality and stationarity. The ADF test gives us a significant p-value and we can reject the null hypothesis that there is at least one unit root.

Now we can find the order of the time series model using the double differenced data.

Estimate the model ARIMA(0,1,1).

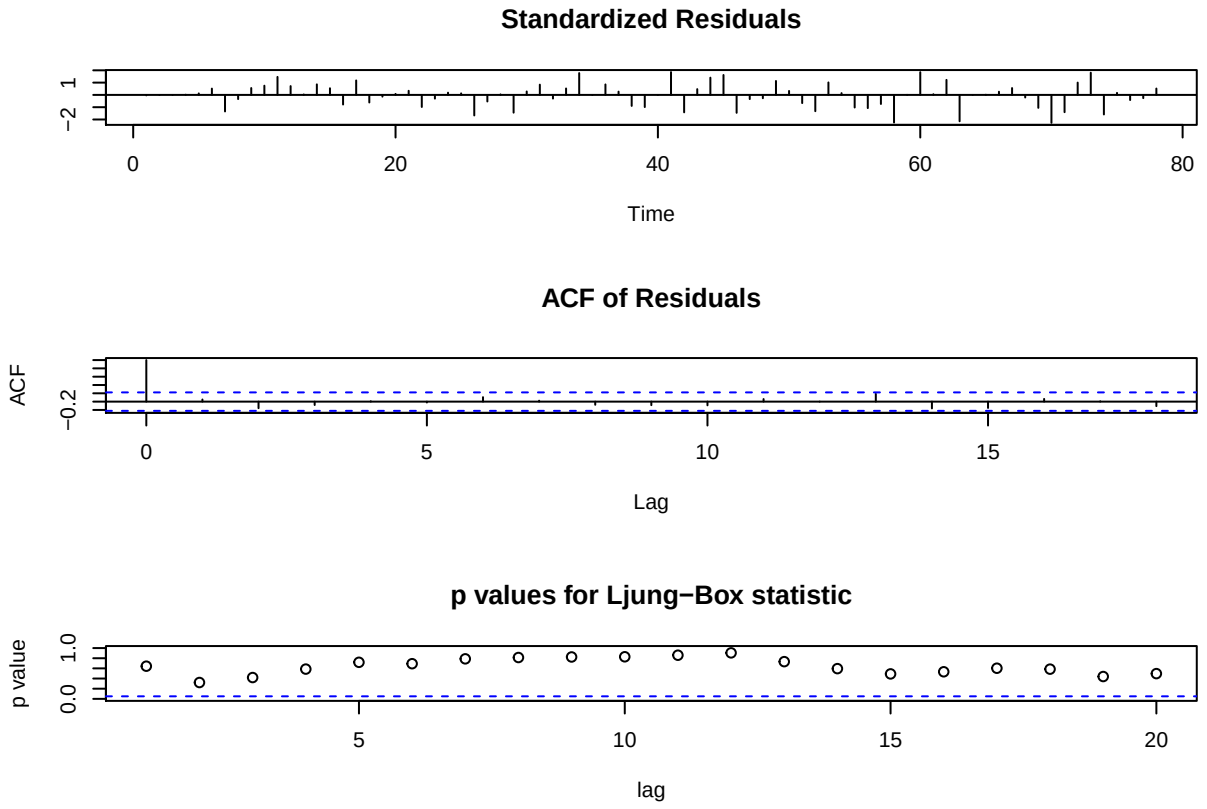
```
dd.mod <- arima(ln.earns,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
dd.mod
```

```
##
## Call:
## arima(x = ln.earns, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 4))
##
```

```
## Coefficients:
##           ma1     sma1
##       -0.3223  -0.2175
## s.e.   0.1372   0.1208
##
## sigma^2 estimated as 0.0011:  log likelihood = 144.9,  aic = -285.81
```

Perform model check

```
# Test the residuals
tsdiag(dd.mod, gof=20)
```



Adjust Box-Ljung Test for lag=12

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

```
##
##  Box-Ljung test
##
## data:  dd.mod$residuals
## X-squared = 6.2041, df = 12, p-value = 0.9054
```

Calculate adjusted p-value for the same statistic.

Adjested DF for 12 lags should be 12-2=10

```
pv12=1-pchisq(6.2041,10)
pv12
```

```
## [1] 0.797834
```

The Ljung Box test gives us a large p value. We cannot reject the null hypothesis that there is no correlation between the residuals of different lags. Our model has already accounted for all the correlations. No additional correlations in the residuals.

Conclusion: The model is adequate.

```
dd.mod
```

```
##
## Call:
## arima(x = ln.earns, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 4))
##
## Coefficients:
##           ma1     sma1
##       -0.3223  -0.2175
## s.e.   0.1372   0.1208
##
## sigma^2 estimated as 0.0011:  log likelihood = 144.9,  aic = -285.81
```

Write down the model

$(1-B)(1-B_4)\xi_t = (1-0.3223B)(1-0.2175B_4)\epsilon_t, \sigma_\epsilon^2 = 0.0011$

Refit the model using data from 1992 to 2008

```
train<-da$earns[1:68]
ln.train<-log(train)
# USe Double Differencing
train.mod <- arima(ln.train,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
#Use Only Seasonal Differencing
train.mod.1 <- arima(ln.train,order=c(0,0,1),seasonal=list(order=c(0,1,1),period=4))
train.mod
```

```
##
## Call:
## arima(x = ln.train, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 4))
##
## Coefficients:
##           ma1     sma1
##       -0.3419  -0.1849
## s.e.   0.1344   0.1389
##
## sigma^2 estimated as 0.001002:  log likelihood = 128,  aic = -252
```

```
train.mod.1
```

```
##
## Call:
## arima(x = ln.train, order = c(0, 0, 1), seasonal = list(order = c(0, 1, 1),
##     period = 4))
##
## Coefficients:
##          ma1    sma1
##       0.8638  0.6449
## s.e.  0.0584  0.0798
##
## sigma^2 estimated as 0.003129:  log likelihood = 92.28,  aic = -180.56
```

Perform 1 to 10 step forecasts of earnings and obtain a forecast plot

```
pm1 <- predict(train.mod, 10)
pm2<-  predict(train.mod.1, 10)
pred=pm1$pred
pred1=pm2$pred
se=pm1$se
se1=pm2$se

act.da=da$earns # actual obser

fore=exp(pred+se^2/2) #point forecasts, delogged
fore1=exp(pred1+se1^2/2)
v1=exp(2*pred+se^2)*(exp(se^2)-1)
v2=exp(2*pred1+se1^2)*(exp(se1^2)-1)
s1=sqrt(v1) # std of the forecast error
s2=sqrt(v2)
eps=act.da[49:78]
length(eps)
```

```
## [1] 30
```

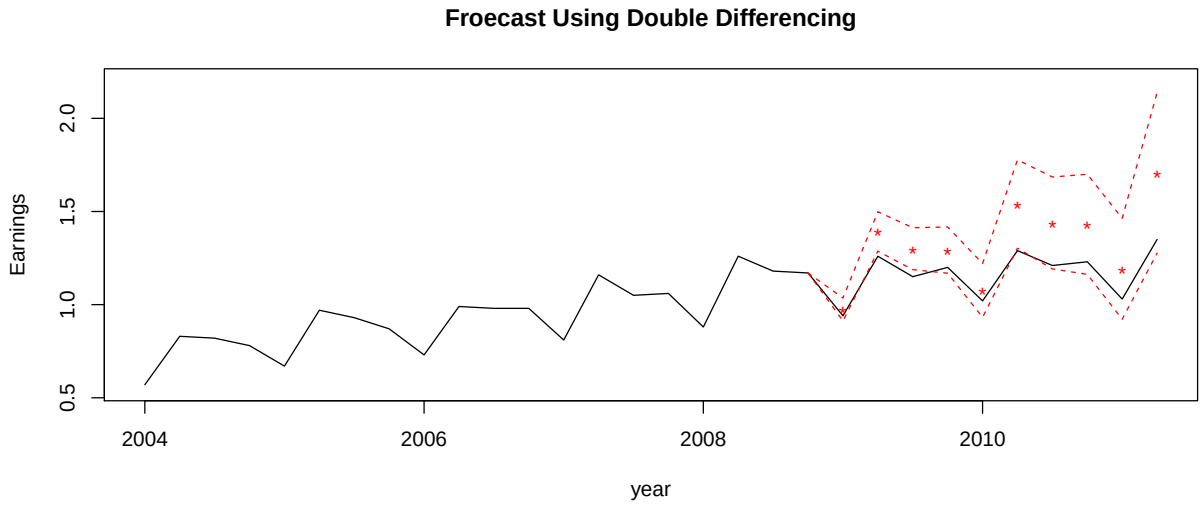
```
tdx=(c(1:30)+3)/4+2003
upp=c(act.da[68],fore+2*s1) # upper band (+2*std)
low=c(act.da[68],fore-2*s1) # lower band (-2*std)
upp1=c(act.da[68],fore1+2*s2) # upper band (+2*std)
low1=c(act.da[68],fore1-2*s2) # lower band (-2*std)
min(low,eps,low1)
```

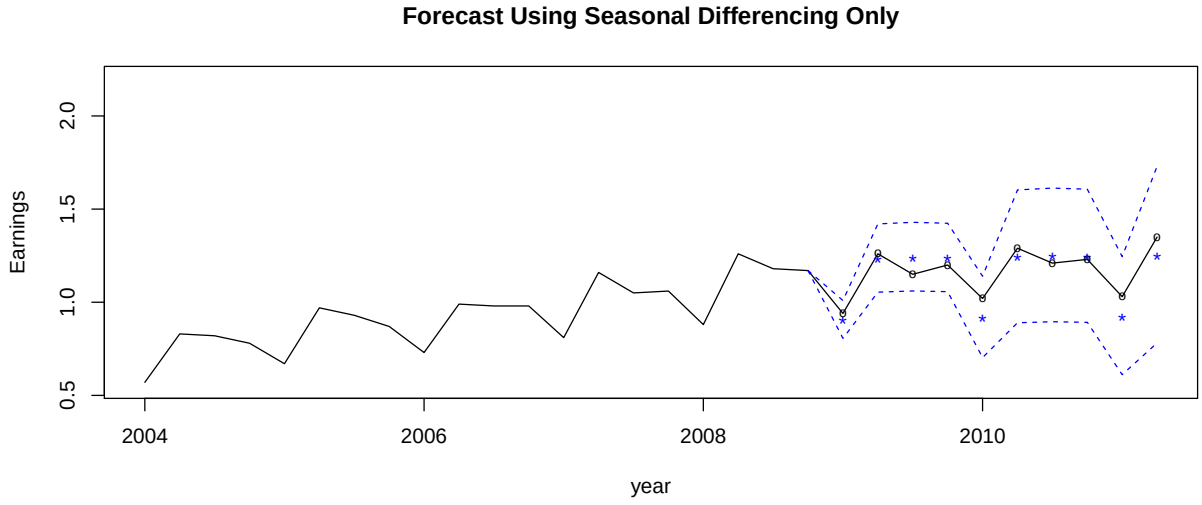
```
## [1] 0.57
```

```
max(upp,eps,upp1)
```

```
## [1] 2.13872
```

```
par(mfcol=c(2,1))
plot(tdx,eps,xlab='year',ylab='Earnings',type='l',ylim=c(0.55,2.2),
     main='Froecast Using Double Differencing')
points(tdx[21:30],fore,pch='*',col='red')
lines(tdx[20:30],upp,lty=2, col='red')
lines(tdx[20:30],low,lty=2, col='red')
plot(tdx,eps,xlab='year',ylab='Earnings',type='l',ylim=c(0.55,2.2),
     main='Forecast Using Seasonal Differencing Only')
points(tdx[21:30],fore1,pch='*',col='blue')
lines(tdx[20:30],upp1,lty=2, col='blue')
lines(tdx[20:30],low1,lty=2, col='blue')
points(tdx[21:30],act.da[69:78],pch='o',cex=0.7)
```





Based on the forecast plots, the seasonal differencing seems to produce better forecast than the double differencing model.