

Assignment4

Manish Kumar

Feb 1, 2023

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-earn-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

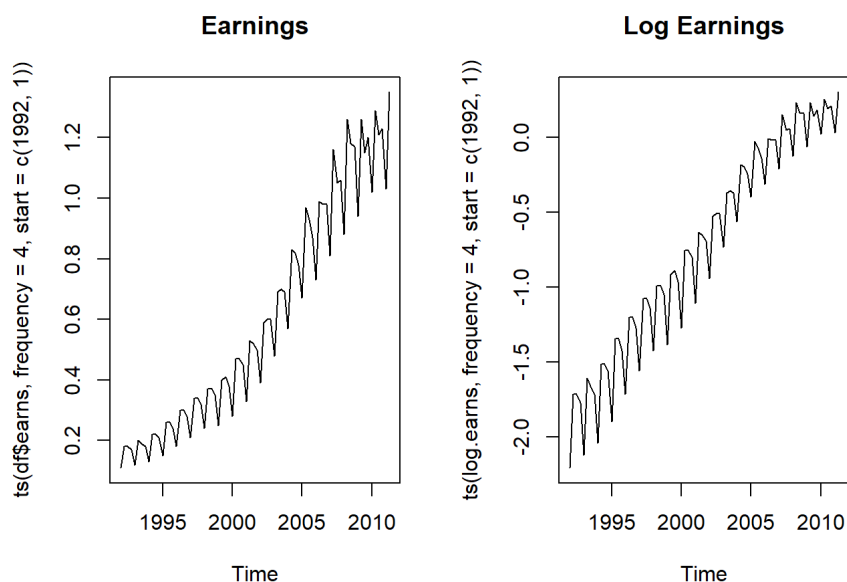
```
# Load the dataset
df = read.table(file="q-jnj-earn-9211.txt", header=T)
head(df)
```

	day <int>	mon <int>	year <int>	earn <dbl>
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

6 rows

Take log transformation if necessary.

```
# Take a log and compare the results in a plot with raw values
log.earn <- log(df$earn)
par(mfrow=c(1,2))
plot(ts(df$earn, frequency=4, start=c(1992,1)), type="l", main="Earnings")
plot(ts(log.earn, frequency=4, start=c(1992,1)), type="l", main="Log Earnings")
```



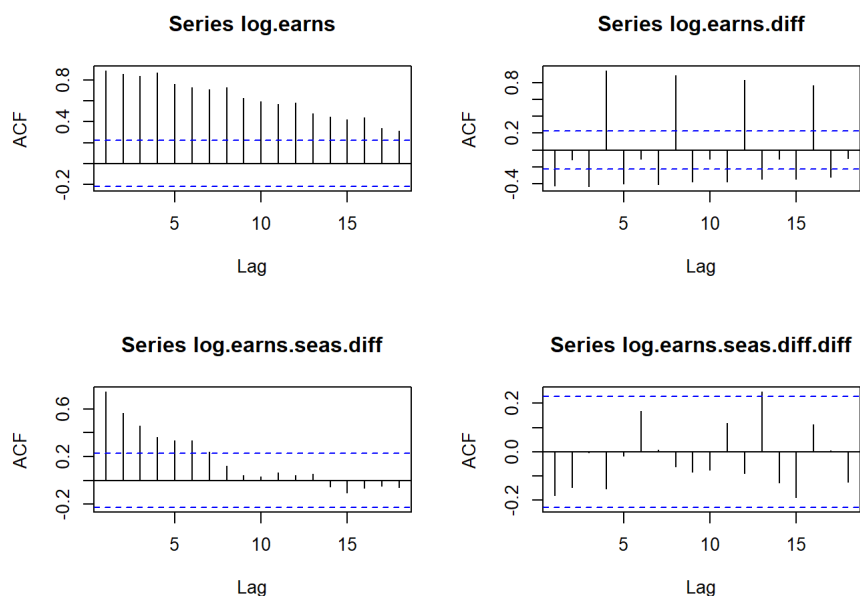
Taking the log has helped in slightly straightening the slope of the timeseries plot. Also, `log.earn` is not stationary.

Create a time series model for the data

```
log.earnings.diff = diff(log.earnings)

# From the previous plots, we could see that there was a seasonality of 4. Add it to earnings difference
log.earnings.seas.diff = diff(log.earnings, 4)
log.earnings.seas.diff.diff = diff(log.earnings.seas.diff)

# Compare the acf plots
par(mfrow=c(2,2))
acf(log.earnings)
acf(log.earnings.diff)
acf(log.earnings.seas.diff)
acf(log.earnings.seas.diff.diff)
```



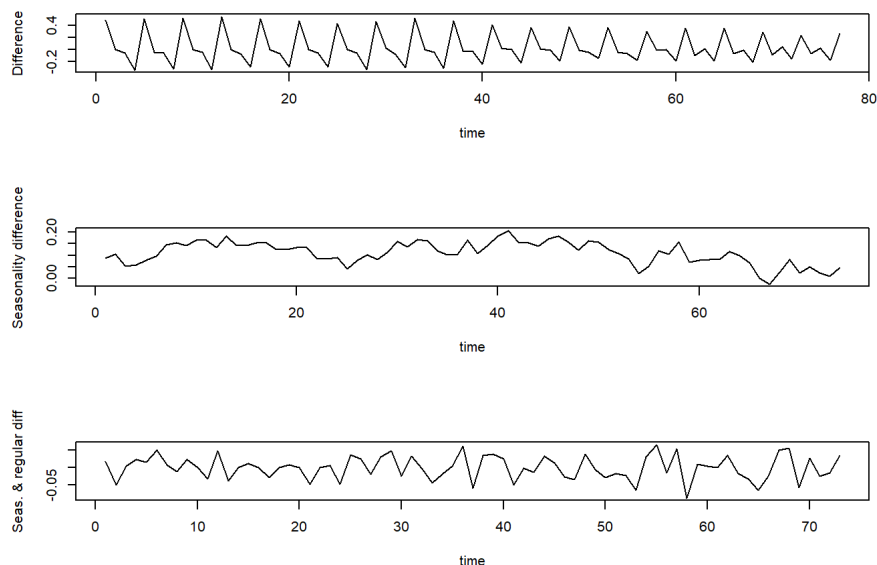
We previously saw that `log.earnings` was not stationary. However, after taking a difference of values (`log.earnings.diff`), it removed the growth and highlighted seasonality. Every 4th bar is positive and large (second plot) so the seasonality is 4.

Further, there's no sharp decay in ACF plots.

Next, we added a seasonality of 4 to the earnings and took the difference again (`log.earnings.seas.diff`). The third ACF plot show an exponential decay.

To deal with this, we can take another difference (`log.earnings.seas.diff.diff`) which seem to have made it stationary and removed seasonality (4th plot). But it seems to be going too far, we don't even have significant correlation at the first lag. This is also shown in the below timeseries plots for the same.

```
# Make timeseries plots
par(mfcol=c(3,1))
plot(log.earnings.diff, xlab="time", ylab="Difference", type="l")
plot(log.earnings.seas.diff, xlab="time", ylab="Seasonality difference", type="l")
plot(log.earnings.seas.diff.diff, xlab="time", ylab="Seas. & regular diff", type="l")
```



Let's do ADF tests to check whether the series are stationary

First, find AR order for these data.

```
ar_model1 = ar(log.earns, method='mle')
adf.test(log.earns, k=ar_model1$order, alternative="stationary")
```

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

```
ar_model2 = ar(log.earns.diff, method='mle')
adf.test(log.earns.diff, k=ar_model2$order, alternative="stationary")
```

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

```
ar_model3 = ar(log.earns.seas.diff, method='mle')
adf.test(log.earns.seas.diff, k=ar_model3$order, alternative="stationary")
```

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

```
ar_model4 = ar(log.earns.seas.diff.diff, method='mle')
adf.test(log.earns.seas.diff.diff, k=ar_model4$order, alternative="stationary")
```

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

ADF test for `log.earns.seas.diff.diff` shows a significant p-value, meaning we can reject the null hypothesis.

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

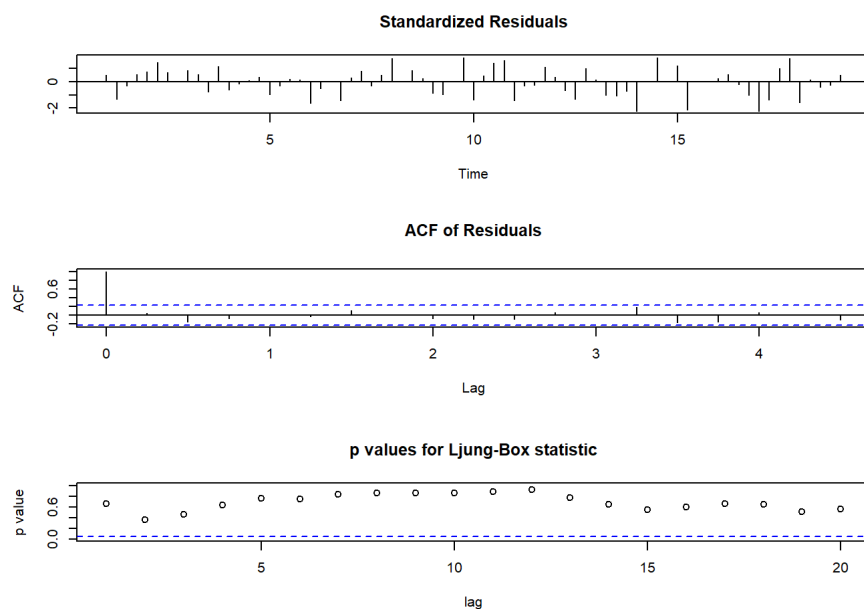
Estimate the model ARIMA(0, 1, 1).

```
arima_model = auto.arima(ts(log.earns.seas.diff.diff, frequency = 4))
arima_model
```

```
## Series: ts(log.earns.seas.diff.diff, frequency = 4)
## ARIMA(0,0,1)(0,0,1)[4] with zero mean
##
## Coefficients:
##      ma1      sma1
##    -0.3223  -0.2175
## s.e.   0.1372   0.1208
##
## sigma^2 = 0.001131: log likelihood = 144.9
## AIC=-283.81   AICc=-283.46   BIC=-276.94
```

Perform model check

```
# Test the residuals
tsdiag(arima_model, gof=20)
```



Adjust Box-Ljung Test for lag=12

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

```
##
## Box-Ljung test
##
## data:  arima_model$residuals
## X-squared = 5.8007, df = 12, p-value = 0.9258
```

p-value is large so we cannot reject the null hypothesis. Therefore, no additional correlations in the residuals.

Conclusion: The model is adequate.

```
arima_model
```

```
## Series: ts(log.earns.seas.diff.diff, frequency = 4)
## ARIMA(0,0,1)(0,0,1)[4] with zero mean
##
## Coefficients:
##      ma1      sma1
##    -0.3223  -0.2175
## s.e.    0.1372   0.1208
##
## sigma^2 = 0.001131: log likelihood = 144.9
## AIC=-283.81   AICc=-283.46   BIC=-276.94
```

Write down the model

$$(1 - B)(1 - B_4)\xi_t = (1 - 0.3223B)(1 - 0.2175B_4)\epsilon_t, \sigma_\epsilon^2 = 0.0012 \quad (1 - \phi_1 B)(1 - \phi_4 B_4)\xi_t = (1 - 0.3223\phi_1 B)(1 - 0.2175\phi_4 B_4)\epsilon_t$$

Refit the model using data from 1992 to 2008

```
df.new = df$earns[1:68]
df.new.log = log(df.new)

df.new.log.diff = diff(df.new.log)

# Create a time series
df.new.log.diff.ts = ts(df.new.log.diff, frequency=4)

# Use auto.arima to suggest best model
best.arima.model = auto.arima(df.new.log.diff.ts, seasonal = T)
summary(best.arima.model)
```

```
## Series: df.new.log.diff.ts
## ARIMA(0,0,1)(0,1,0)[4]
##
## Coefficients:
##          ma1
##        -0.3245
## s.e.    0.1421
##
## sigma^2 = 0.001049: log likelihood = 127.14
## AIC=-250.29   AICc=-250.09   BIC=-246
##
## Training set error measures:
##              ME      RMSE      MAE MPE MAPE      MASE      ACF1
## Training set 0.000291181 0.0311555 0.02427697 NaN  Inf 0.9163081 0.03676114
```

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

```
pred = predict(best.arma.model, 10)$pred
se = predict(best.arma.model, 10)$se

actual.df = df$earnings

fore = exp(pred + se^2/2)
v1 = exp(2*pred + se^2)*(exp(se^2) - 1)
s1 = sqrt(v1)
eps = actual.df[49:78]
length(eps)
```

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

```
tdx = (c(1:30) + 3) / 4 + 2003
upp = c(actual.df[68], fore + 2*s1)
low = c(actual.df[68], fore - 2*s1)
min(low, eps)
```

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

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

```
## [1] 1.603636
```

```
plot(tdx, eps, xlab='year', ylab='earnings', type='l', ylim=c(0.35, 1.8))
points(tdx[19:28], fore, pch='*')
lines(tdx[18:28], upp, lty=2)
lines(tdx[18:28], low, lty=2)
points(tdx[19:28], df$earnings[69:78], pch='o', cex=0.7)
```

