Time Series Assignment 3

Matthew Dunne

July 9, 2018

Question 1:

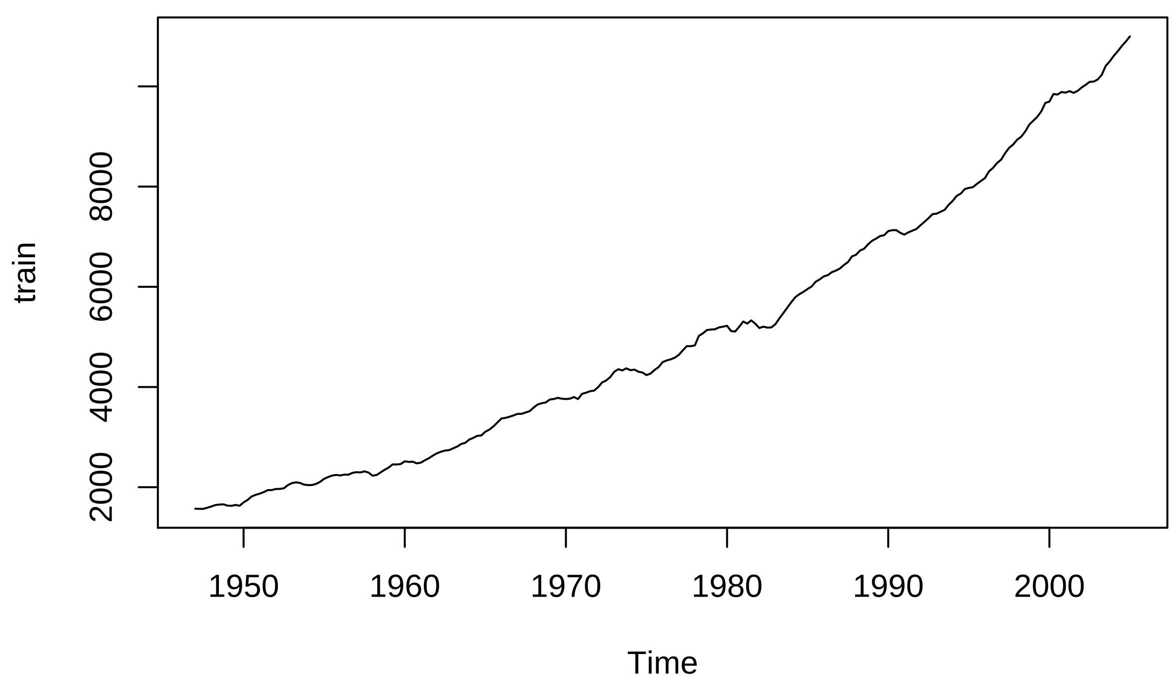
Load the usgdp.rda dataset and split it into a training dataset (1947Q1 - 2005Q1) and a test dataset (2005Q2 - 2006Q1)

```
load("C:/Users/mjdun/Desktop/Time Series/Assignments/usgdp.rda")
train<-window(usgdp, c(1947, 1), c(2005, 1))
test<-window(usgdp, c(2005, 2), c(2006, 1))
suppressMessages(library(fpp))
suppressMessages(library(TSA))
```

Question 2:

Plot the training dataset. Is Box-Cox transformation necessary for this data?

```
plot(train)
```

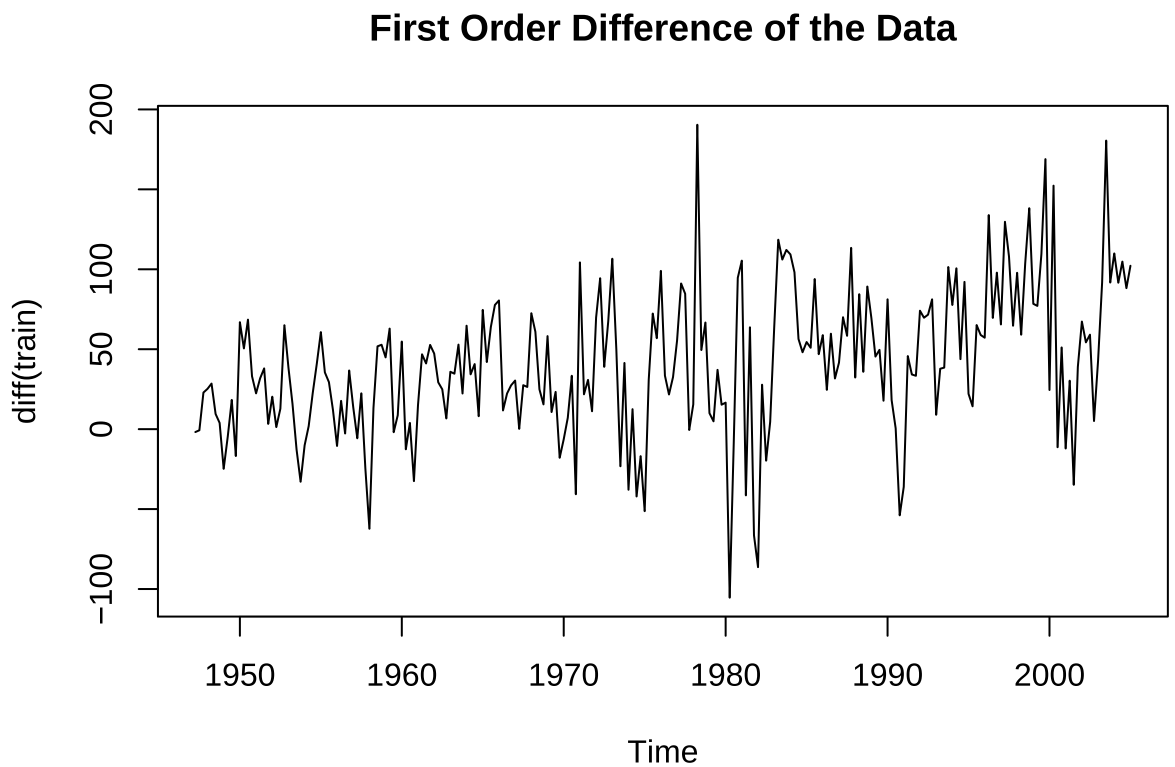


The Box-Cox transformation is not necessary for the data. While there is a clear trend upwards, the variance does not change. The Box-Cox transformation is used when the *variation* increases with the level of the series.

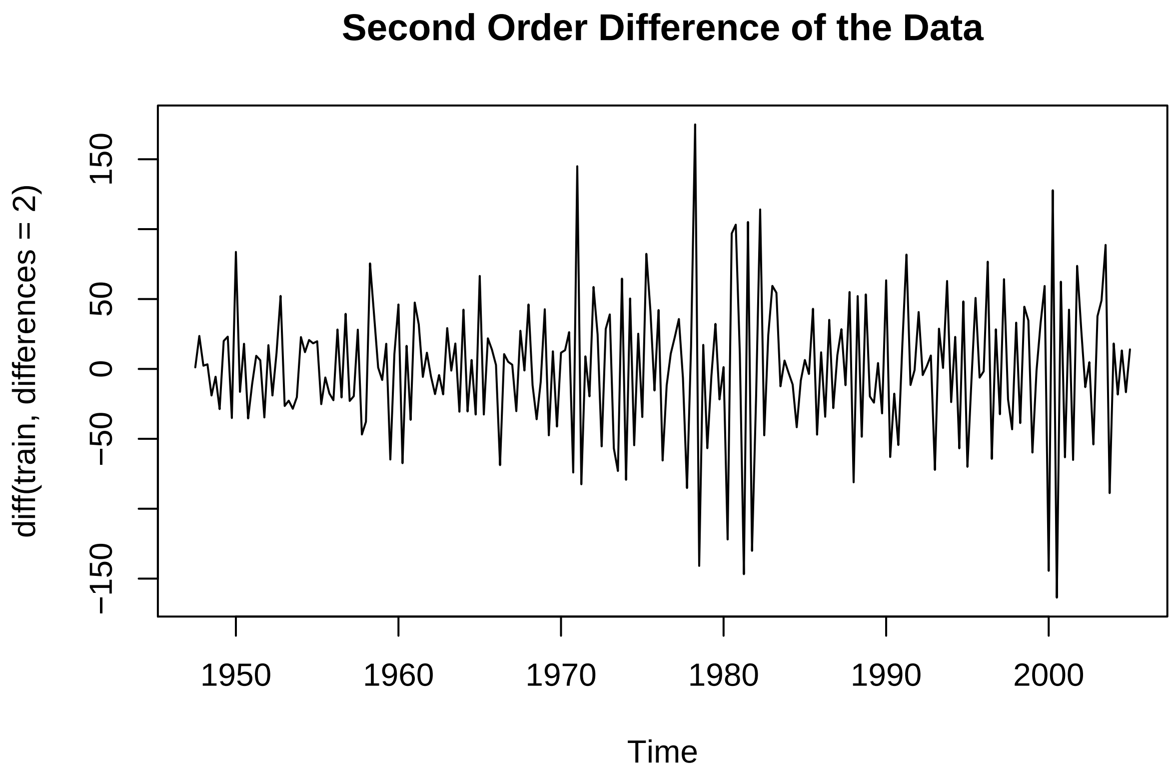
Question 3:

Plot the 1st and 2nd order difference of the data. Apply KPSS Test for Stationarity to determine which difference order results in a stationary dataset.

```
plot(diff(train), main="First Order Difference of the Data")
```




```
plot(diff(train, differences = 2), main="Second Order Difference of the Data")
```



There is a slight upward trend in the 1st order difference. Applying the KPSS Test for stationarity, we see a small p-value of 0.01 which is indicative of non-stationarity.

```
first<-diff(train)
kpss.test(first, null = "Level")
```

```
## Warning in kpss.test(first, null = "Level"): p-value smaller than printed
## p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  first
## KPSS Level = 1.8306, Truncation lag parameter = 3, p-value = 0.01
```

Applying the KPSS to the 2nd order data, we get a p-value of 0.1 which suggests that the data is now stationary.

```
second<-diff(train, differences=2)
kpss.test(second)
```

```
## Warning in kpss.test(second): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  second
## KPSS Level = 0.010349, Truncation lag parameter = 3, p-value = 0.1
```

Question 4:

Fit a suitable ARIMA model to the transformed data using the auto.arima() function. Report the resulting p, d, q and the coefficient values.

```
fit<-auto.arima(train, d=2)
fit
```

```
## Series: train
## ARIMA(2,2,2)
##
## Coefficients:
##           ar1     ar2      ma1      ma2
##       -0.1138  0.3059  -0.5829  -0.3710
## s.e.   0.2849  0.0895   0.2971   0.2844
##
## sigma^2 estimated as 1591:  log likelihood=-1178.16
## AIC=2366.32   AICc=2366.59   BIC=2383.53
```

The model returns an ARIMA(2, 2, 2) which means p=2, d=2, and q=2. Running the auto.arima with the default value for d instead of 2 allows the algorithm to choose a d based on the KPSS test. When I run it with the default value it also returns a value of d=2.

Question 5:

Compute the sample Extended ACF (EACF) and use the Arima() function to try some other plausible models by experimenting with the orders chosen. Limit your models to q, p <=2 and d<=2. Use the model summary() function to compare the Corrected Akaike information criterion (i.e., AICc) values (Note: Smaller values indicated better models).

The EACF function on the 2nd Order data is:

```
eacf(second)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o x o o o o x x o o  x  o  o
## 1 x x o o o o o x x o o  x  o  o
## 2 x x x o o o o o o o o  x  o  o
## 3 x x o o o o o o o o o  x  o  o
## 4 x x o o o o o o o o o  x  o  o
## 5 x x x o o o o x o o o  x  o  o
## 6 x o x x x o x x o o o  x  o  o
## 7 o x x x x o x o o o o  x  o  o
```

Limiting our q and p to <=2 we are left to choose between ARIMA(0, 2, 1) and ARIMA(1, 2, 2).

```
fit2<-Arima(train, c(0,2,1))
fit3<-Arima(train, c(1,2,2))
```

The AICc for the ARIMA(0, 2, 1) is:

```
fit2$aicc
```

```
## [1] 2383.029
```

And the AICc for the ARIMA(1, 2, 2) is:

```r
fit2$aicc
```

```
## [1] 2383.029
```

which is slightly lower.

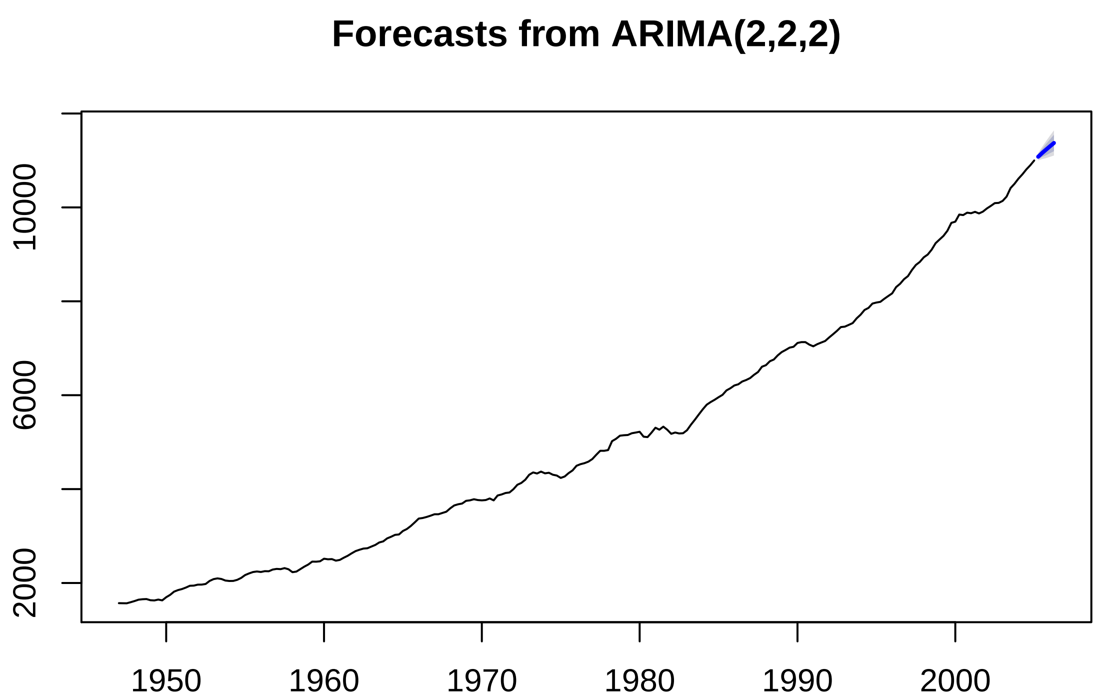
Question 6:

Use the model chosen in Question 4 to forecast the GDP for 2005Q2 - 2006Q1 (Test Period).

In question 4 we had ARIMA(2, 2, 2). There are five periods in our Test Period and so we will forecast ahead by 5.

Here is the plot, which is somewhat difficult to see:

```r
plot(forecast(fit,h=5))
```



And the actual forecast values with confidence intervals:

```r
forecast(fit,h=5)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2005 Q2       11079.04 11027.93 11130.15 11000.87 11157.21
## 2005 Q3       11157.71 11073.75 11241.68 11029.31 11286.12
## 2005 Q4       11229.64 11111.69 11347.59 11049.26 11410.03
## 2006 Q1       11302.01 11154.80 11449.23 11076.87 11527.15
## 2006 Q2       11372.27 11197.16 11547.37 11104.47 11640.06
```

Question 7:

Compare your forecasts with the actual values using error = actual - estimate and plot the errors. (Note: Use the forecast $mean element for the forecast estimate)

```
#forecasted values
forecast<-forecast(fit,h=5)$mean
#actual values from test - forecasted values
errors<-test-forecast
errors
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2005           10.16263  44.58517  18.65855
## 2006 101.58798
```

Question 8:

Calculate the sum of squared error.

```
sum(errors^2)
```

```
## [1] 12759.38
```