MSCA 31006 Assignment3

Duo Zhou

10/25/2020

Instructions:

. Total number of points is 36. The assignmnet's final grade will be multiplied by 1/6 to calculate its weight on the final grade.

. Mark the question number and your final answer clearly (use a textbox.)

. Remember to show and explain your work (If you can't explain it, you don't understand it.)

. Please submit your solution through Canvas.

For this exercise, we will use the Quarterly US GDP 1947Q1 - 2006Q1 dataset from the FPP package (Data set: usgdp.rda).

(1 points) Question 1:

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

```r
data(usgdp)
(usgdp_train<-ts(usgdp[1:233],start=c(1947,1),frequency = 4))
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 1947  1570.5  1568.7  1568.0  1590.9
## 1948  1616.1  1644.6  1654.1  1658.0
## 1949  1633.2  1628.4  1646.7  1629.9
## 1950  1696.8  1747.3  1815.8  1848.9
## 1951  1871.3  1903.1  1941.1  1944.4
## 1952  1964.7  1966.0  1978.8  2043.8
## 1953  2082.3  2098.1  2085.4  2052.5
## 1954  2042.4  2044.3  2066.9  2107.8
## 1955  2168.5  2204.0  2233.4  2245.3
## 1956  2234.8  2252.5  2249.8  2286.5
## 1957  2300.3  2294.6  2317.0  2292.5
## 1958  2230.2  2243.4  2295.2  2348.0
## 1959  2392.9  2455.8  2453.9  2462.6
## 1960  2517.4  2504.8  2508.7  2476.2
## 1961  2491.2  2538.0  2579.1  2631.8
## 1962  2679.1  2708.4  2733.3  2740.0
## 1963  2775.9  2810.6  2863.5  2885.8
## 1964  2950.5  2984.8  3025.5  3033.6
## 1965  3108.2  3150.2  3214.1  3291.8
```

```
## 1966  3372.3  3384.0  3406.3  3433.7
## 1967  3464.1  3464.3  3491.8  3518.2
## 1968  3590.7  3651.6  3676.5  3692.0
## 1969  3750.2  3760.9  3784.2  3766.3
## 1970  3760.0  3767.1  3800.5  3759.8
## 1971  3864.1  3885.9  3916.7  3927.9
## 1972  3997.7  4092.1  4131.1  4198.7
## 1973  4305.3  4355.1  4331.9  4373.3
## 1974  4335.4  4347.9  4305.8  4288.9
## 1975  4237.6  4268.6  4340.9  4397.8
## 1976  4496.8  4530.3  4552.0  4584.6
## 1977  4640.0  4731.1  4815.8  4815.3
## 1978  4830.8  5021.2  5070.7  5137.4
## 1979  5147.4  5152.3  5189.4  5204.7
## 1980  5221.3  5115.9  5107.4  5202.1
## 1981  5307.5  5266.1  5329.8  5263.4
## 1982  5177.1  5204.9  5185.2  5189.8
## 1983  5253.8  5372.3  5478.4  5590.5
## 1984  5699.8  5797.9  5854.3  5902.4
## 1985  5956.9  6007.8  6101.7  6148.6
## 1986  6207.4  6232.0  6291.7  6323.4
## 1987  6365.0  6435.0  6493.4  6606.8
## 1988  6639.1  6723.5  6759.4  6848.6
## 1989  6918.1  6963.5  7013.1  7030.9
## 1990  7112.1  7130.3  7130.8  7076.9
## 1991  7040.8  7086.5  7120.7  7154.1
## 1992  7228.2  7297.9  7369.5  7450.7
## 1993  7459.7  7497.5  7536.0  7637.4
## 1994  7715.1  7815.7  7859.5  7951.6
## 1995  7973.7  7988.0  8053.1  8112.0
## 1996  8169.2  8303.1  8372.7  8470.6
## 1997  8536.1  8665.8  8773.7  8838.4
## 1998  8936.2  8995.3  9098.9  9237.1
## 1999  9315.5  9392.6  9502.2  9671.1
## 2000  9695.6  9847.9  9836.6  9887.7
## 2001  9875.6  9905.9  9871.1  9910.0
## 2002  9977.3 10031.6 10090.7 10095.8
## 2003 10138.6 10230.4 10410.9 10502.6
## 2004 10612.5 10704.1 10808.9 10897.1
## 2005 10999.3
```

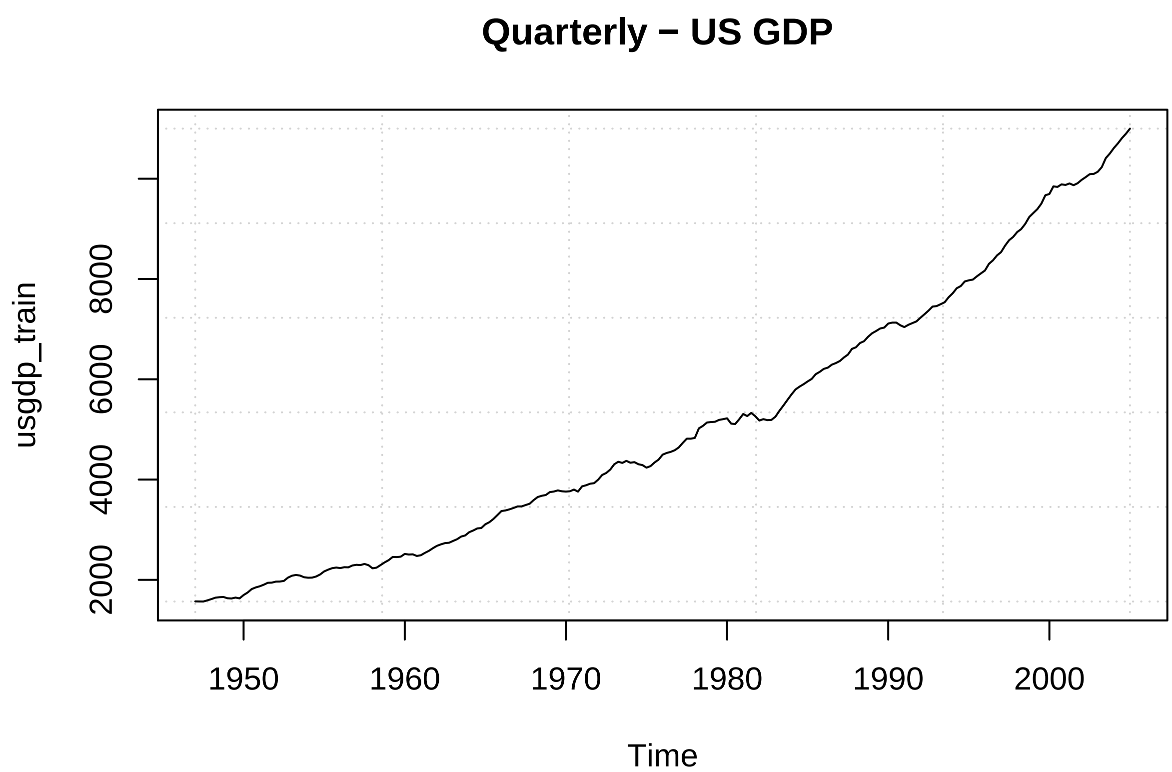
```
(usgdp_test<-ts(usgdp[234:237],start=c(2005,2),frequency = 4))
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 2005         11089.2 11202.3 11248.3
## 2006 11403.6
```

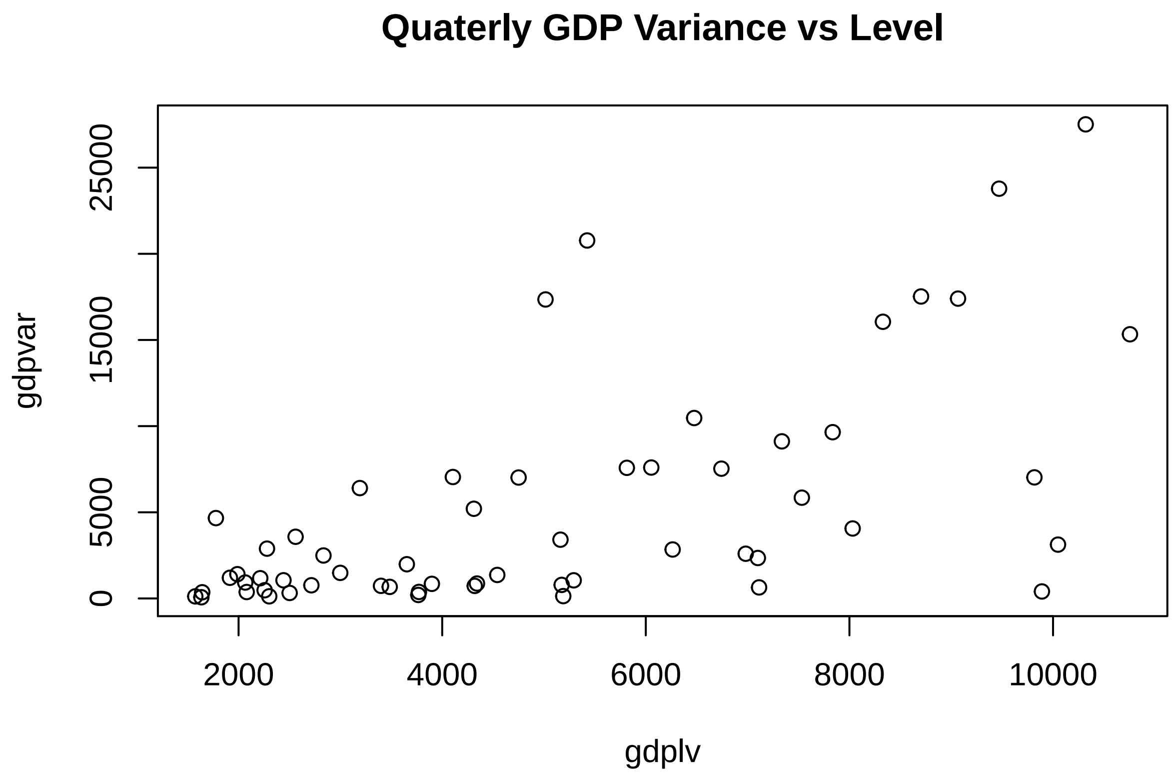
(5 points) Question 2:

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

```
plot(usgdp_train, main="Quarterly - US GDP ",panel.first = grid())
```



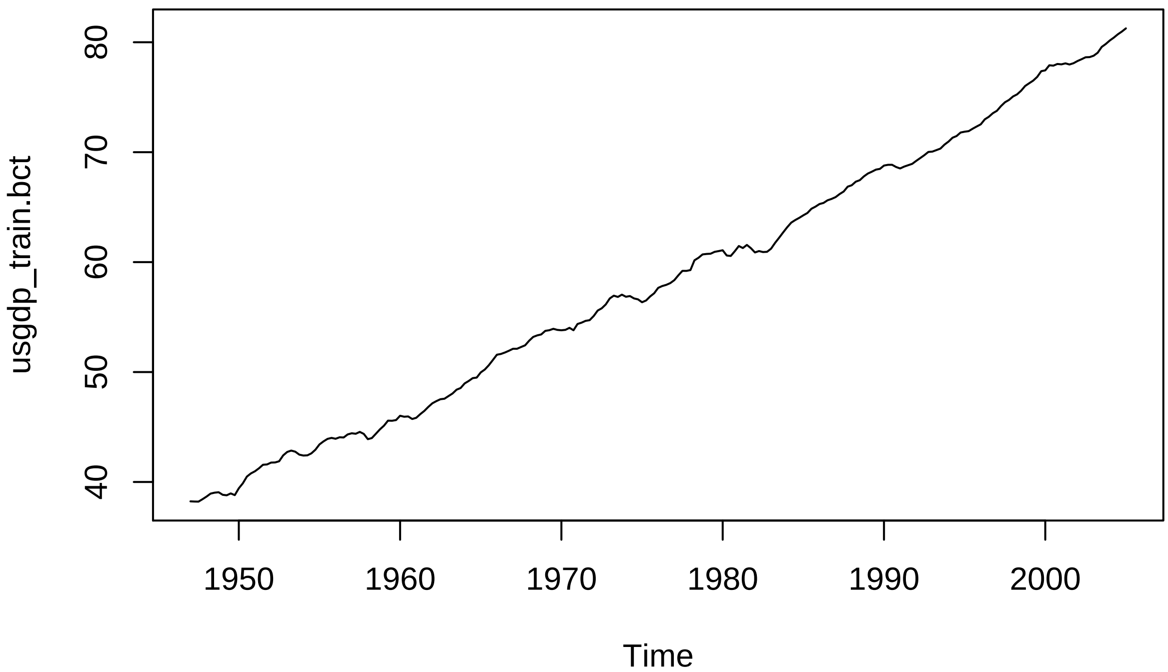

```
gdpvar<-c()
gdplv<-c()
for (i in 1:as.integer(233/4)){
    gdpvar[i]<-var(usgdp_train[seq(4*i,4*i-3,-1)])
    gdplv[i]<-mean(usgdp_train[seq(4*i,4*i-3,-1)])
}
plot(x=gdplv,y=gdpvar,main='Quaterly GDP Variance vs Level')
```




```
BoxCox.lambda(usgdp_train)
```

```
## [1] 0.3689656
```

```
usgdp_train.bct<-BoxCox(usgdp_train,lambda = 0.3689656)
usgdp_test.bct<-BoxCox(usgdp_test,lambda = 0.3689656)
plot(usgdp_train.bct)
```



From the plot, We can see that variance changes with level and suggested lamda for BoxCox transformation is 0.3689656. But this is not enough to conclude that whether Box-Cox Transformation is necessary for the data. I will produce best arima models for both original data and box-cox transformed data and make the conclusion based on the sum of squared error from the forecast of each model.

```
# There does not appear to be a seasonal pattern, so we fit the best model without seasonality
fit.uchanged<-auto.arima(usgdp_train, seasonal = F)
fit.bct<-auto.arima(usgdp_train.bct,seasonal = F)
fc.uc<-forecast(fit.uchanged,h=4)$mean
fc.bct<- bimixt::boxcox.inv(forecast(fit.bct,h=4)$mean,lambda = 0.3689656)
sse.uc<-sum((fc.uc-usgdp_test)**2)
sse.bct<-sum((fc.bct-usgdp_test)**2)
cat('Forecast SSE of Unchanged Data:',sse.uc,'\n')
```

```
## Forecast SSE of Unchanged Data: 12759.38
```

```
cat('Forecast SSE of Box-Cox Transformed Data:',sse.bct)
```

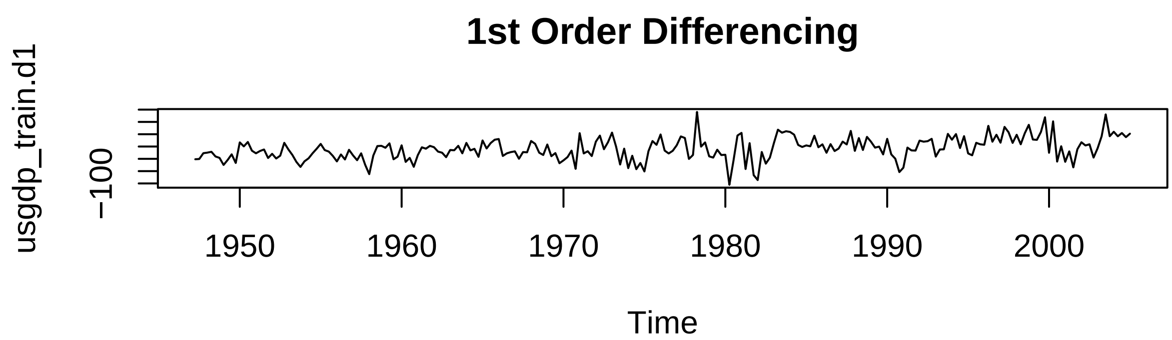
```
## Forecast SSE of Box-Cox Transformed Data: 15844.55
```

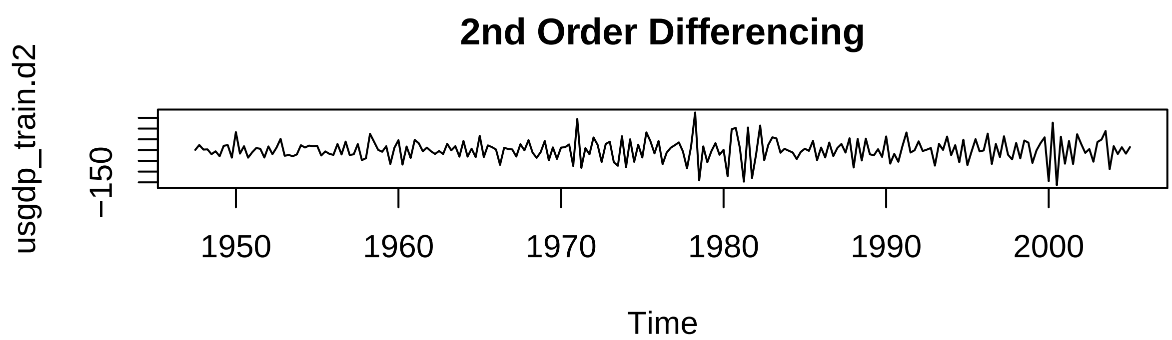
We can see that Forecast SSE from the best ARIMA model of Unchanged Data is smaller than Forecast SSE from the best ARIMA model of Box-Cox Transformed Data. The conclusion is that Box-Cox Transformation is NOT necessary.

(5 points) 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.

```
usgdp_train.d1<-diff(usgdp_train)
usgdp_train.d2<-diff(usgdp_train,differences = 2)
par(mfrow=c(2,1))
plot(usgdp_train.d1,main='1st Order Differencing')
plot(usgdp_train.d2,main='2nd Order Differencing')
```






```
print('1st Order Differencing')
```

```
## [1] "1st Order Differencing"
```

```
tseries::kpss.test(usgdp_train.d1)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  usgdp_train.d1
## KPSS Level = 1.6348, Truncation lag parameter = 4, p-value = 0.01
```

```r
print('2nd Order Differencing')
```

```
## [1] "2nd Order Differencing"
```

```r
tseries::kpss.test(usgdp_train.d2)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  usgdp_train.d2
## KPSS Level = 0.0116, Truncation lag parameter = 4, p-value = 0.1
```

Based on the KPSS test, 1st Order Differencing p-value =0.01, which is less than 0.05 and 2nd Order Differencing p-value-0.1, which is greater than 0.05. With 5% confidence level, we can conclude that the null hythothesis(H0), data is stationary should be rejected for 1st Order Differencing and H0 should be accepted for 2nd Order Differencing. 2nd order differencing results in a stationary dataset.

(5 points) Question 4:

Fit a suitable ARIMA model to the training dataset using the auto.arima() function. Remember to transform the data first if necessary. Report the resulting p, d, q and the coefficients values

```r
# There does not appear to be a seasonal pattern, so we fit the best model without seasonality
(fit<-auto.arima(usgdp_train,seasonal = F))
```

```
## Series: usgdp_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
```

p=2, d=2 and q=2. We see that order of differencing is 2 which is consistent with our conclusion in Q3. $\phi_1 = -0.1138, \phi_2 = 0.3059, \theta_1 = -0.5829, \theta_2 = -0.3710$ and c=0.

The model equation is:

$(1 - (-0.1138)B - (0.3059)B^2)(1 - B)^2 y_t = 0 + (1 + (-0.5829)B + (-0.3710)B^2)\epsilon_t$

(5 points) 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<= 2, 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).

```r
TSA::eacf(usgdp_train.d2)
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

```
## 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
```

```r
(fit1<-Arima(usgdp_train,order=c(0,2,1)))
```

```
## Series: usgdp_train
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.7006
## s.e.   0.0770
##
## sigma^2 estimated as 1741:  log likelihood=-1189.49
## AIC=2382.98   AICc=2383.03   BIC=2389.86
```

```r
(fit2<-Arima(usgdp_train,order=c(1,2,2)))
```

```
## Series: usgdp_train
## ARIMA(1,2,2)
##
## Coefficients:
##          ar1      ma1     ma2
##       0.6424  -1.3239  0.3441
## s.e.  0.1177   0.1344  0.1279
##
## sigma^2 estimated as 1614:  log likelihood=-1180.33
## AIC=2368.66   AICc=2368.83   BIC=2382.42
```

```r
fit
```

```
## Series: usgdp_train
## ARIMA(2,2,2)
##
## Coefficients:
##           ar1     ar2      ma1      ma2
```

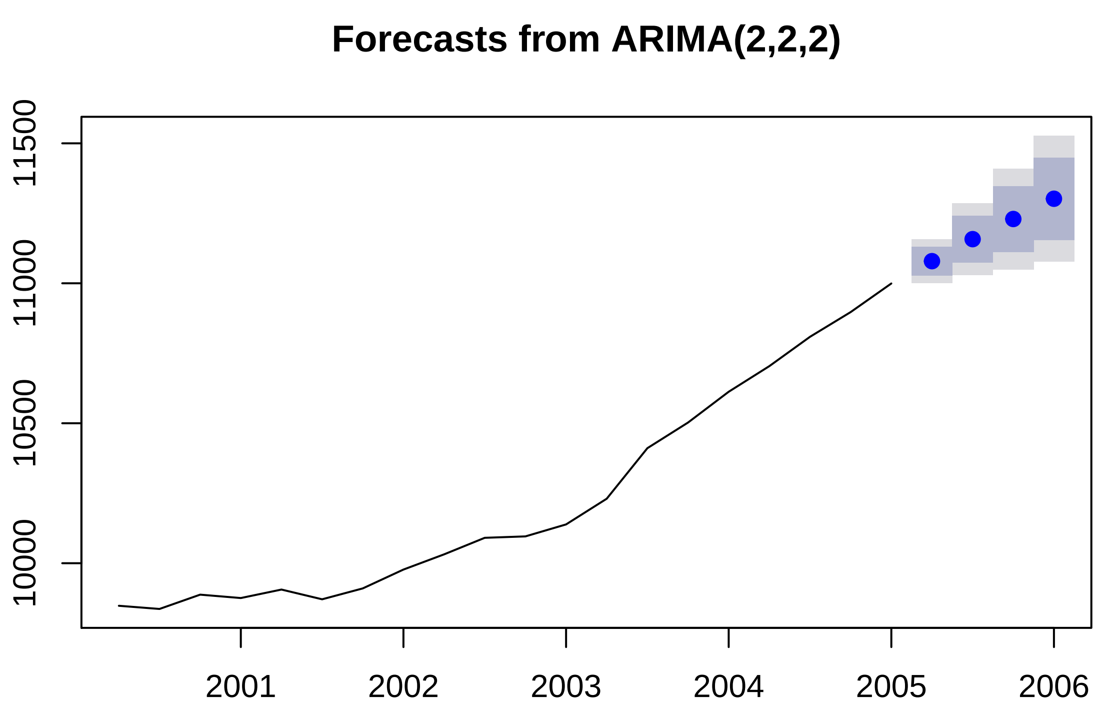
```
##       -0.1138  0.3059  -0.5829  -0.3710
## s.e.   0.2849  0.0895   0.2971   0.2844
##
## sigma^2 estimated as 1591:  log likelihood=-1178.16
## AIC=2366.32   AICc=2366.59   BIC=2383.53
```

The EACF of 2nd order differencing data suggests that ARIMA(0,2,1) and ARIMA(1,2,2) are also significant. Comparing these two model with the best model auto.aroma selected, ARIMA(2,2,2), we can see that ARIMA(2,2,2) has the smallest AICc, hence, ARIMA(2,2,2) is the best model.

(5 points) Question 6:

Use the model chosen in Question 4 to forecast and plot the GDP forecasts with 80 and 95% confidence levels for 2005Q2 - 2006Q1 (Test Period).

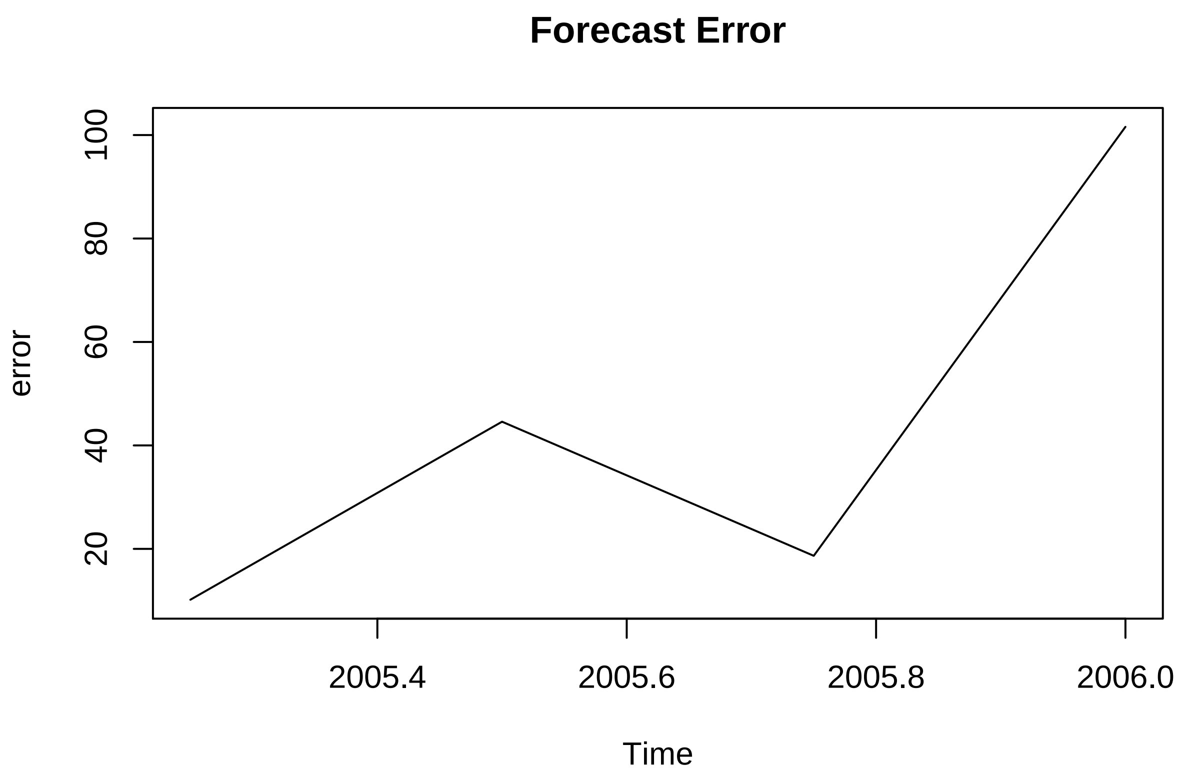
```r
plot(forecast(fit,h=4),include=20)
```



(5 points) 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)

```r
error<- usgdp_test-forecast(fit.uchanged,h=4)$mean
plot(error, main='Forecast Error')
```



(5 points) Question 8:

Calculate the sum of squared error.

```r
sse<-sum(error**2)
cat('Forecast SSE of ARIMA(2,2,2) is', sse,'\n')
```

```
## Forecast SSE of ARIMA(2,2,2) is 12759.38
```