```r
library('expsmooth')
library('fpp2')
library('fitdistrplus')
library('logspline')
library('xts')
library('forecast');
library('fma')
library('lmtest')
library('tseries')
library('Quandl')
library('fpp');
library('urca')
library('TSA')
```

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

```r
myts.train <- window(usgdp, end=c(2005,1))
myts.test  <- window(usgdp, start=c(2005,2))
head(myts.train)
```

```
##        Qtr1   Qtr2   Qtr3   Qtr4
## 1947 1570.5 1568.7 1568.0 1590.9
## 1948 1616.1 1644.6
```

```r
tail(myts.train)
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 2003                         10502.6
## 2004 10612.5 10704.1 10808.9 10897.1
## 2005 10999.3
```

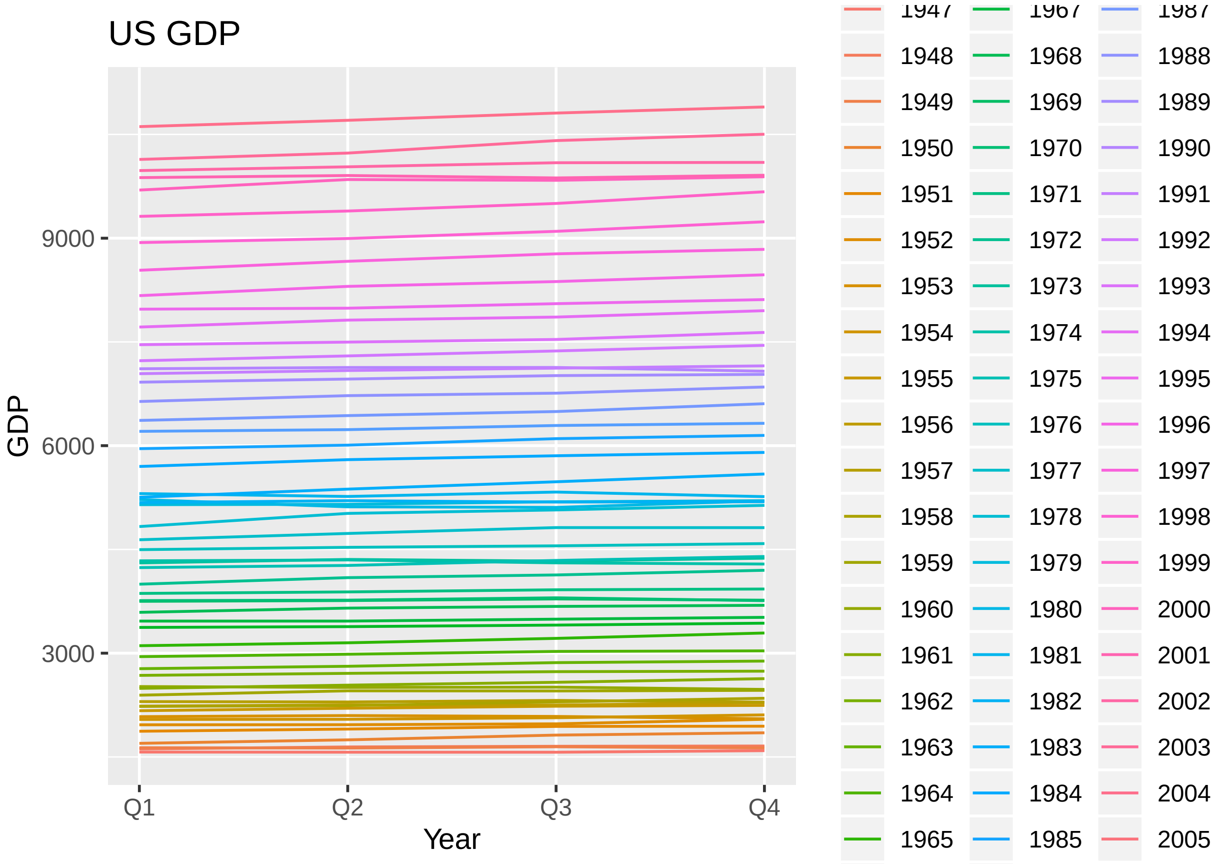
```r
head(myts.test)
```

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

```r
x <- ts(myts.train, frequency=365/90)
fit <- tbats(x)
seasonal <- !is.null(fit$seasonal)
print(paste('Seasonality = ',seasonal))
```

```
## [1] "Seasonality =  FALSE"
```

```r
ggseasonplot(myts.train)+
  ggtitle("US GDP") +
  xlab("Year") +
  ylab("GDP")
```



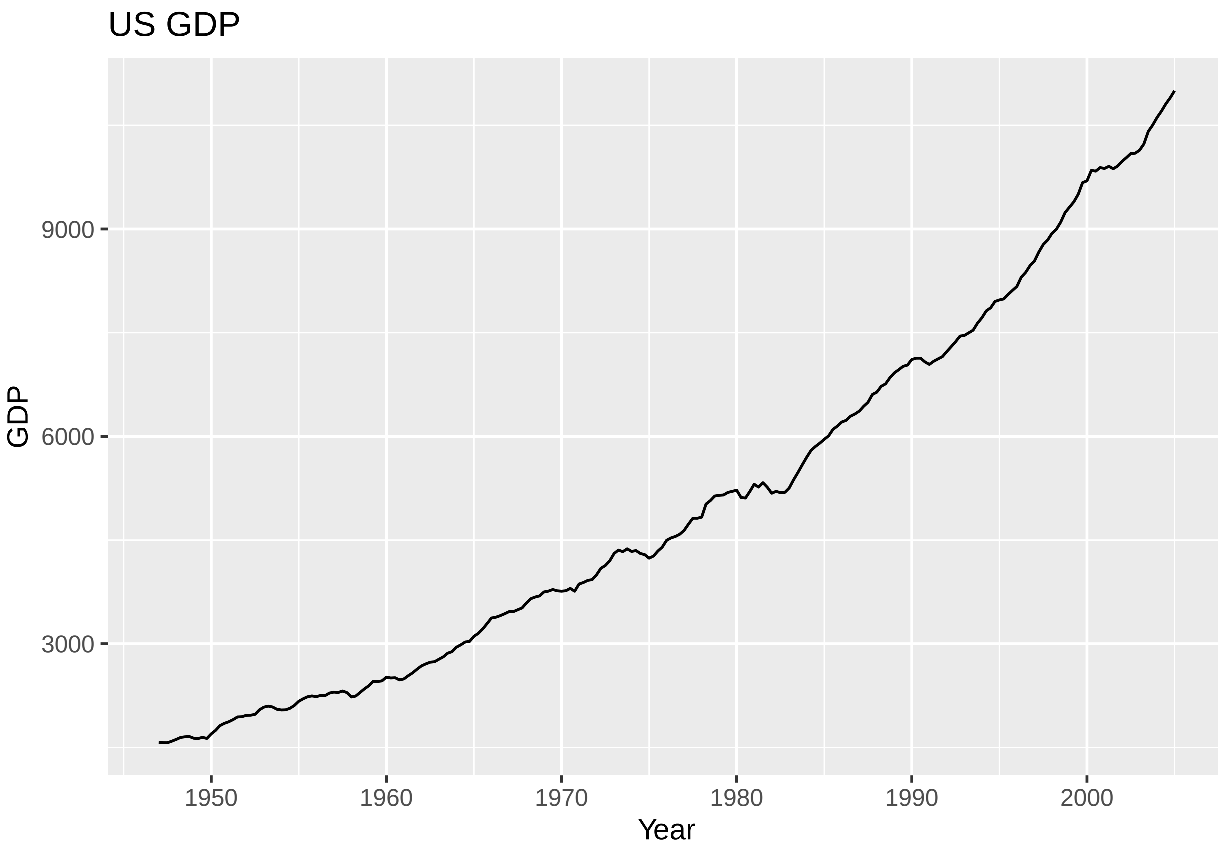

```
Features:

- Trend as there is increase

- No seasonal pattern occurs.I used the tbats model.   It will handle quarter seasonality
and will automatically determine if a seasonal pattern is present,I aslo used ggseasonplot
and it show no Seasonality
```

I Plotted the training dataset and test if Box-Cox transformation necessary for this data?

```r
autoplot(myts.train) +
  ggtitle("US GDP") +
  xlab("Year") +
  ylab("GDP")
```




```
I don't belive Box-Cox transformation is nessesry for this data because data show no
variation that increases with the level of the series
```

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

```r
ndiffs(myts.train)
```

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

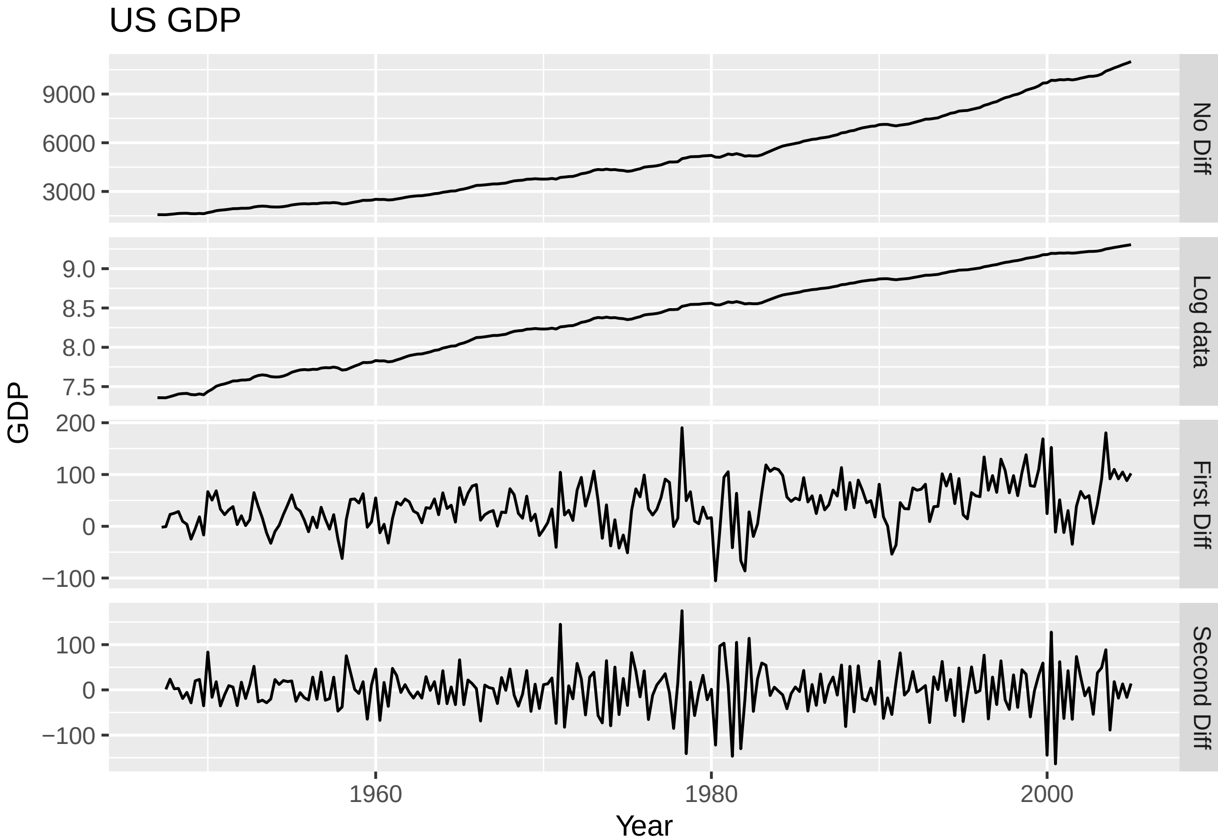
```r
DY <- diff(myts.train)
ndiffs(DY)
```

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

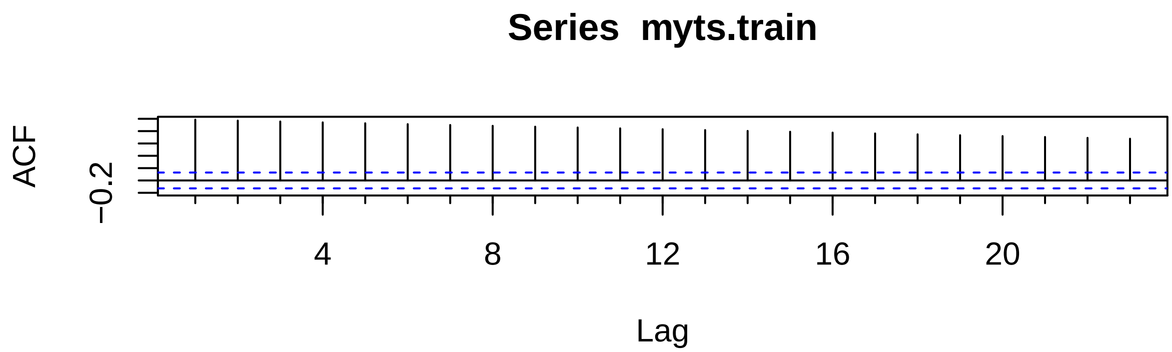
```r
DY2 <- diff(DY)
ndiffs(DY2)
```

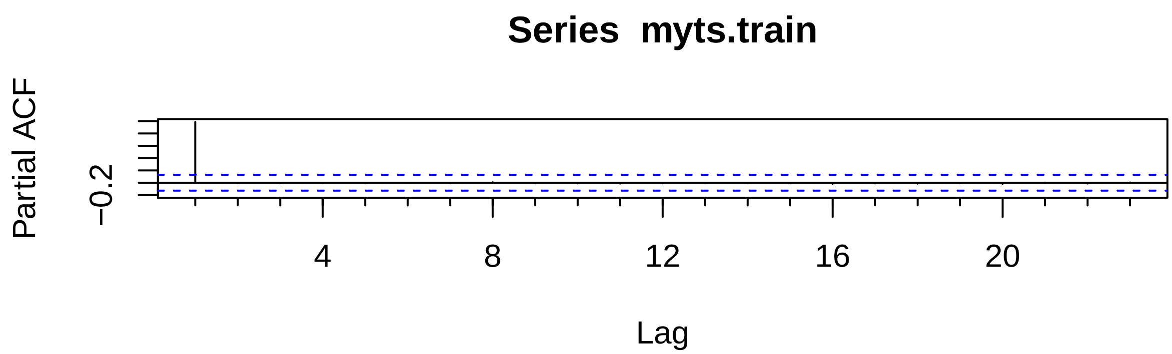
```
## [1] 0
```

```r
cbind("No Diff" = myts.train,
      "Log data" = log(myts.train),
      "First Diff" = DY,
      "Second Diff" = DY2) %>%
autoplot(facets=TRUE) +
  ggtitle("US GDP") +
  xlab("Year") +
  ylab("GDP")
```




```
par(mfrow=c(2,1))
Acf(myts.train)
Pacf(myts.train)
```





As seen from the ACF graph, there are significant lags. PACF tells a slight different story.

```
(LB_test <- Box.test(myts.train,lag=20, type='Ljung-Box'))
```

```
##
##  Box-Ljung test
##
## data:  myts.train
## X-squared = 3589.3, df = 20, p-value < 2.2e-16
```

```
(adf_test <- adf.test(myts.train,alternative = 'stationary')) # p-value < 0.05 indicates the TS is stat
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  myts.train
## Dickey-Fuller = 0.22209, Lag order = 6, p-value = 0.99
## alternative hypothesis: stationary
```

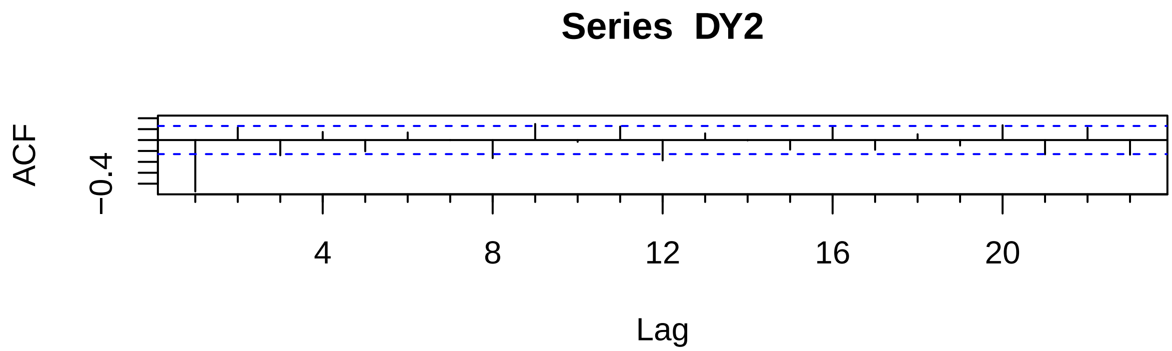
```
(kpss.test(myts.train))#low p-value indicate not trend stationary (non-stationary)
```

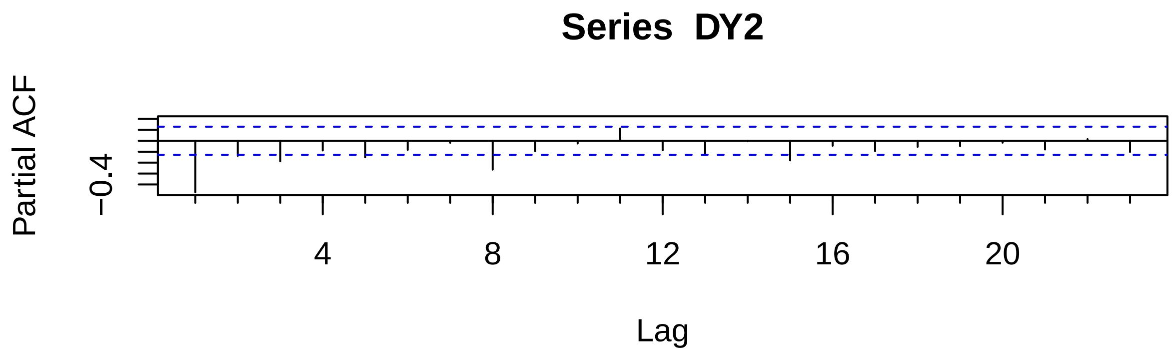
```
##
##  KPSS Test for Level Stationarity
##
## data:  myts.train
## KPSS Level = 4.5854, Truncation lag parameter = 4, p-value = 0.01
```

While using Ljung-Box testing stationarity, it shows a very small p-value which indicates that the time series is stationary. let's do same steps for dataset after applying 2nd Diff

```
par(mfrow=c(2,1))
Acf(DY2)
```

```r
Pacf(DY2)
```






```r
(LB_test <- Box.test(DY2,lag=20, type='Ljung-Box'))# small p-value indicates is stationary
```

```
##
##  Box-Ljung test
##
## data:  DY2
## X-squared = 103.56, df = 20, p-value = 2.888e-13
```

```r
(adf_test <- adf.test(DY2,alternative = 'stationary'))# p-value < 0.05 indicates the TS is stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  DY2
## Dickey-Fuller = -7.7362, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
(kpss.test(DY2))#low p-value indicate not trend stationary (non-stationary)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  DY2
## KPSS Level = 0.0116, Truncation lag parameter = 4, p-value = 0.1
```

Now Ljung-Box indicates is stationary (small p-value), adf.test indicates is stationary (p-value < 0.05) and kpss.test indicate trend stationary

```r
ndiffs(myts.train)
```

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

```r
myts.train %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 4.5854
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
myts.train %>% diff() %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 1.6348
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
myts.train %>% diff() %>% diff() %>% ur.kpss() %>% summary()
```

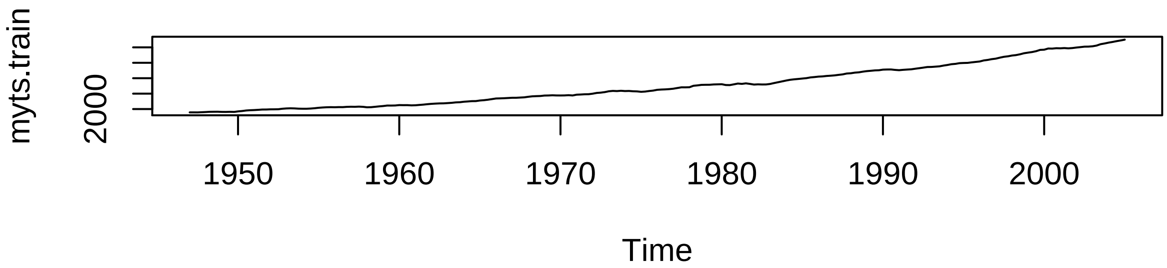
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0116
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

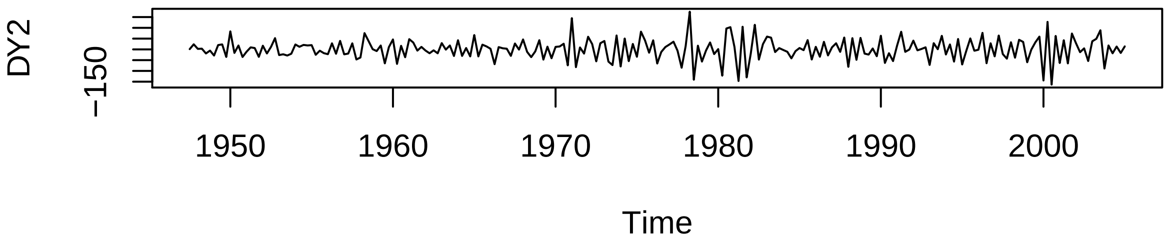
As we saw from the KPSS tests above, two difference is required to make myts.train data stationary.

I Fitted a suitable ARIMA model to the training dataset using the auto.arima() function.

```r
par(mfrow=c(2,1))
plot(myts.train)
```

```r
plot(DY2)
```






```r
(fit.arima <- auto.arima(myts.train))
```

```
## Series: myts.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=0, q=2 Coefficients -0.1138,  0.3059,  -0.5829,  -0.3710,
```

I Computed the sample Extended ACF (EACF) and use the Arima() function to try some other plausible models by experimenting with the orders chosen. I used the model summary() function to compare the Corrected Akaike information criterion (i.e., AICc) values.

```r
ESACF <- eacf(myts.train)
```

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

```
## 7 x x x o o o o o o o o  x  o  o
```

```
ESACF$eacf
```

```
##            [,1]       [,2]        [,3]        [,4]        [,5]        [,6]
## [1,]  0.9854526  0.9709213  0.95637039  0.94196034  0.92765263  0.91354946
## [2,]  0.2898900  0.2485453  0.04504781  0.04040684 -0.07027550 -0.03483321
## [3,] -0.4704419  0.1176892 -0.13973178  0.07395996 -0.10297731  0.06940009
## [4,] -0.2088022 -0.3470191 -0.09589388 -0.07732555 -0.06272136  0.06990899
## [5,] -0.4109410 -0.3130569  0.13979434 -0.02540268 -0.08836902 -0.01550184
## [6,] -0.3102071 -0.4988335  0.07251747 -0.05015580 -0.03060165  0.01488731
## [7,] -0.3427759 -0.4916090 -0.10360211 -0.04362155  0.02734371  0.02876827
## [8,] -0.3759012  0.2345572 -0.23828848 -0.06471809  0.05319577  0.07197193
##              [,7]        [,8]        [,9]        [,10]       [,11]
## [1,]  0.899539968  0.88595297 0.872428171  0.858791182  0.84496161
## [2,] -0.103467354 -0.16684899 0.008168336 -0.022191460 -0.03415054
## [3,] -0.001485117 -0.16518686 0.147169803 -0.016442615  0.11768340
## [4,] -0.003208908 -0.17594982 0.143392583  0.008695110  0.04700099
## [5,] -0.030321436 -0.11962479 0.115069179  0.091948660  0.10000533
## [6,] -0.026973139 -0.08482391 0.107378366 -0.002154568  0.08776324
## [7,] -0.038000066 -0.09869539 0.107740982 -0.002291578  0.01319021
## [8,] -0.042989721 -0.13497853 0.113673334  0.048783431  0.02375258
##           [,12]       [,13]        [,14]
## [1,]  0.8310623  0.81729703  0.803660452
## [2,] -0.2166426 -0.13597236 -0.139739257
## [3,] -0.1870696  0.06124089 -0.003253400
## [4,] -0.1835256  0.03913603  0.001004924
## [5,] -0.1795976 -0.01564670 -0.087808549
## [6,] -0.1921485 -0.09231279 -0.079441223
## [7,] -0.1904952  0.02990272  0.032981416
## [8,] -0.1938398  0.07794531  0.017598845
```

```
(EACF1 <- Arima(myts.train, order=c(0,2,1)))
```

```
## Series: myts.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
```

```
(EACF2 <- Arima(myts.train, order=c(0,2,2)))
```

```
## Series: myts.train
## ARIMA(0,2,2)
##
## Coefficients:
##           ma1      ma2
##       -0.7431  -0.1915
## s.e.   0.0544   0.0529
##
## sigma^2 estimated as 1693:  log likelihood=-1186.4
```

```
## AIC=2378.79   AICc=2378.9   BIC=2389.12
```

```r
(EACF3 <- Arima(myts.train, order=c(1,2,1)))
```

```
## Series: myts.train
## ARIMA(1,2,1)
##
## Coefficients:
##          ar1      ma1
##       0.3026  -0.9597
## s.e.  0.0662   0.0180
##
## sigma^2 estimated as 1645:  log likelihood=-1183.09
## AIC=2372.18   AICc=2372.29   BIC=2382.51
```

```r
(EACF4 <- Arima(myts.train, order=c(1,2,2)))
```

```
## Series: myts.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
(EACF5 <- Arima(myts.train, order=c(2,2,1)))
```

```
## Series: myts.train
## ARIMA(2,2,1)
##
## Coefficients:
##          ar1     ar2      ma1
##       0.2551  0.1965  -0.9688
## s.e.  0.0663  0.0660   0.0147
##
## sigma^2 estimated as 1592:  log likelihood=-1178.72
## AIC=2365.43   AICc=2365.61   BIC=2379.2
```

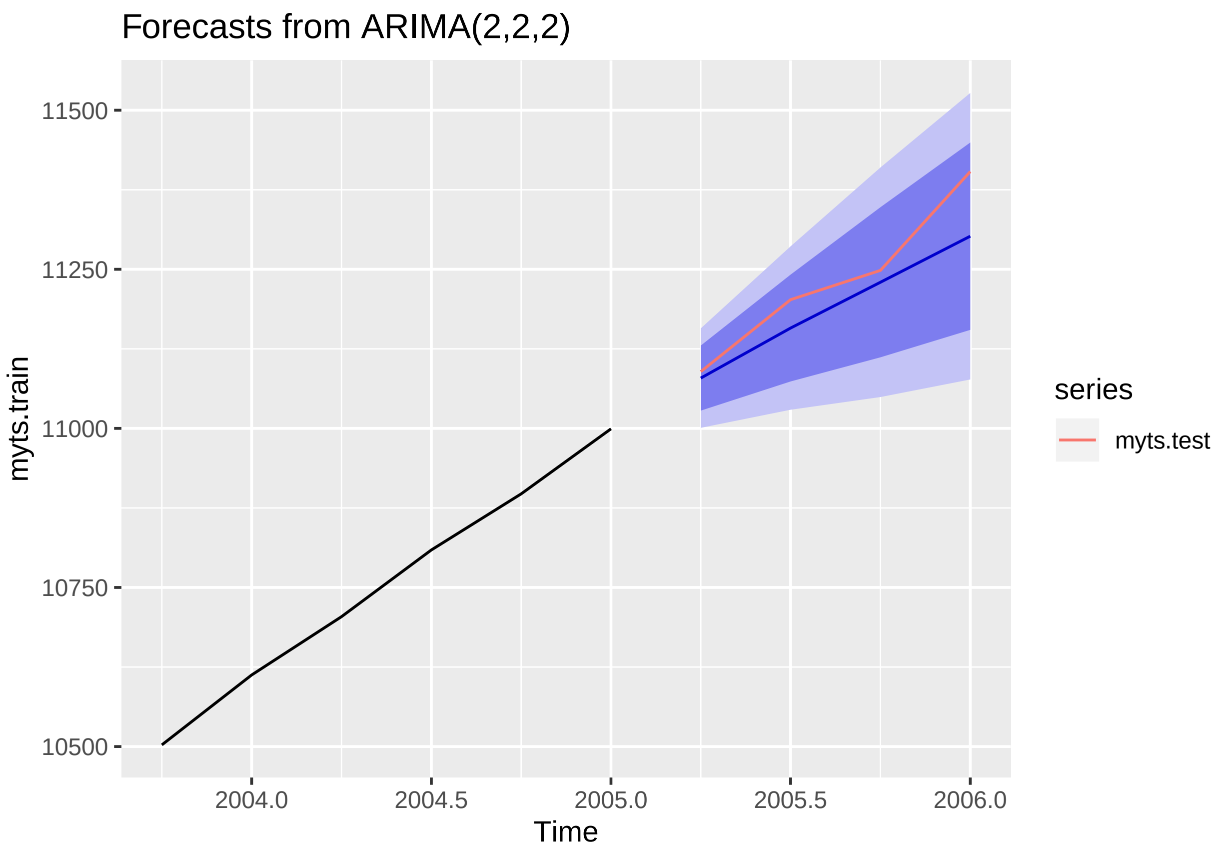
```r
(EACF6 <- Arima(myts.train, order=c(2,2,2)))
```

```
## Series: myts.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
```

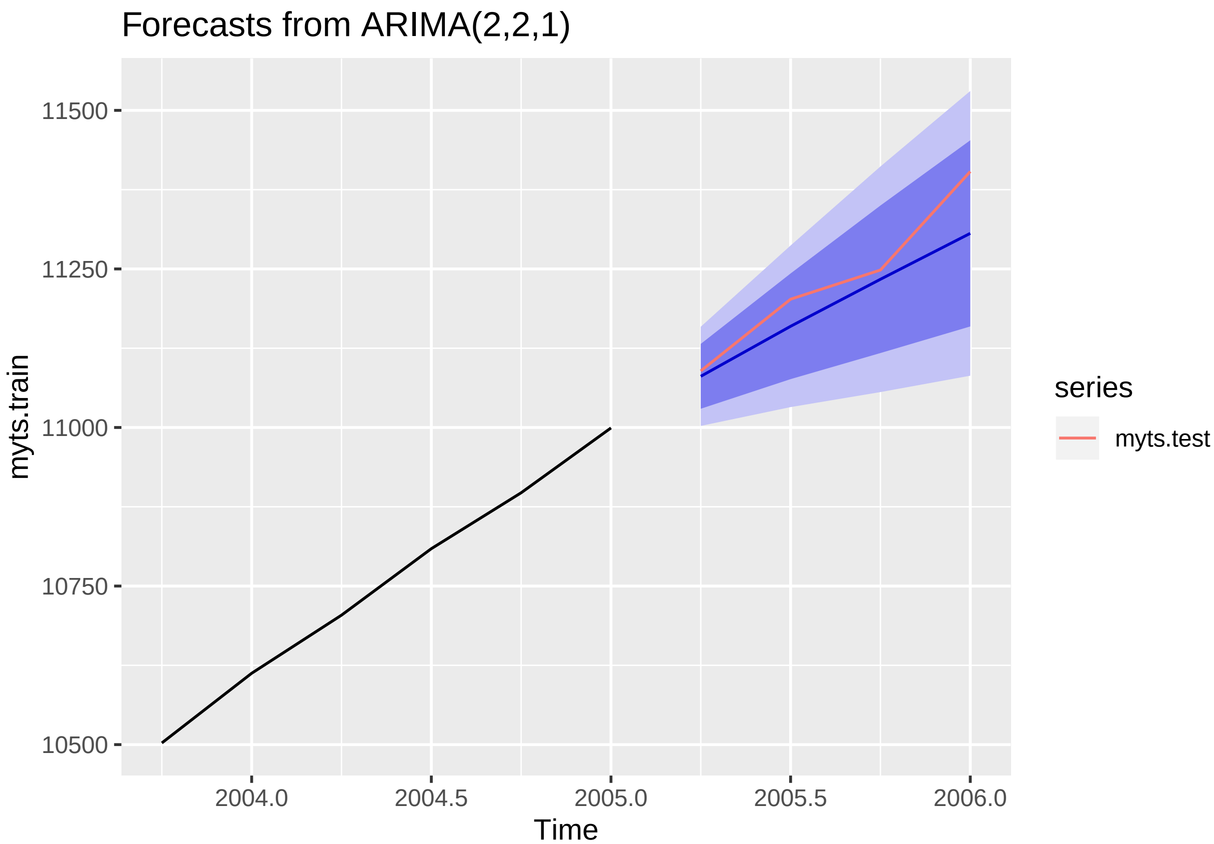
Order=c(2,2,1) provide slightly better model than auto.arima which (2,2,2)

I used the best model to forecast and plot the GDP forecasts with 80 and 95 % confidence levels for 2005Q2 - 2006Q1 (Test Period).

```
autoplot(forecast(fit.arima,h=length(myts.test)),include=6)+ autolayer(myts.test)
```




```
autoplot(forecast(EACF5,h=length(myts.test)),include=6)+ autolayer(myts.test)
```



I

```
exclude some early data and include last 6Q to increase the size of the polt so prediction
become clear
```

I compared my forecasts with the actual values using error = actual - estimate and plot the errors.

```r
fit.arima.forecast <- forecast(fit.arima,h=length(myts.test))
myts.test-fit.arima.forecast$mean
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2005            10.16263  44.58518  18.65855
## 2006 101.58798
```

```r
EACF5.forecast <- forecast(EACF5,h=length(myts.test))
myts.test-EACF5.forecast$mean
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2005           8.49132 42.73454 14.61519
## 2006 97.50587
```

```
Order=c(2,2,1) provide less error than auto.arima which (2,2,2)
```

I calculated the sum of squared error.

```r
accuracy(fit.arima.forecast,myts.test)
```

```
##                     ME     RMSE      MAE        MPE      MAPE      MASE
## Training set  4.304891 39.36568 29.47422 0.09278553 0.6958960 0.1700480
## Test set     43.748585 56.47870 43.74858 0.38659119 0.3865912 0.2524023
```

```
##                      ACF1 Theil's U
## Training set -0.01735678        NA
## Test set     -0.29396381   0.56802
```

```r
accuracy(EACF5.forecast,myts.test)
```

```
##                     ME     RMSE      MAE       MPE      MAPE      MASE
## Training set  4.272536 39.46556 29.48761 0.0920902 0.7003115 0.1701253
## Test set     40.836729 53.89652 40.83673 0.3607575 0.3607575 0.2356026
##                       ACF1 Theil's U
## Training set -0.002588972        NA
## Test set     -0.322725699 0.5425548
```

It is clear that the model with (2,2,1) is better