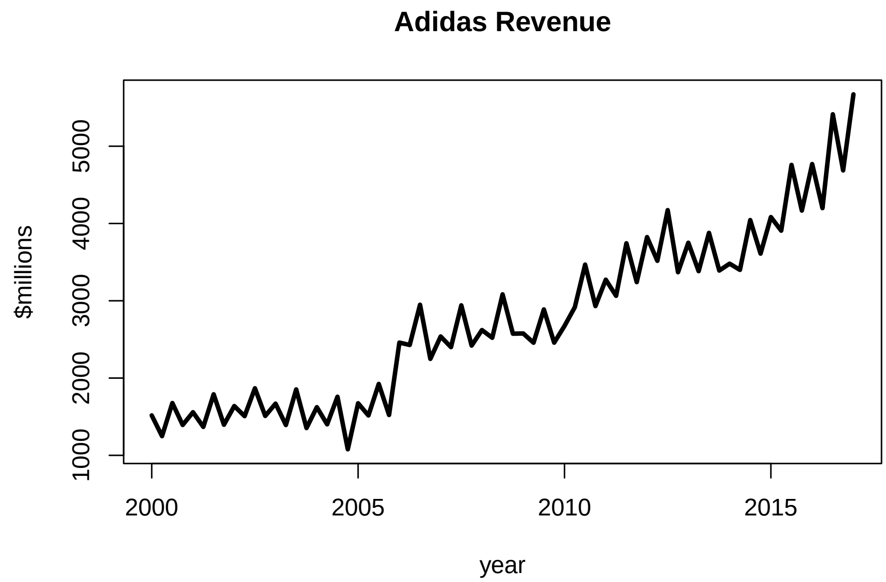
MA641 Project

ADIDAS AG is a multinational corporation, founded and head-quartered in Herzogenaurach, Germany, that designs and manufactures shoes, clothing and accessories. It is the largest sportswear manufacturer in Europe, and the second largest in the world, after Nike. In the fiscal year of 2016, ADIDAS generated a total revenue of $19,068 million, increased by 18 percent. In its annual report, ADIDAS projects an annual growth in sale between 11 percent and 13 percent For this paper, I used the past quarterly sales data of ADIDAS to forecast its future sales from Q2 of 2017 to Q4 of 2018, applying Time Series Theories to predict whether the goal of ADIDAS can be reached.

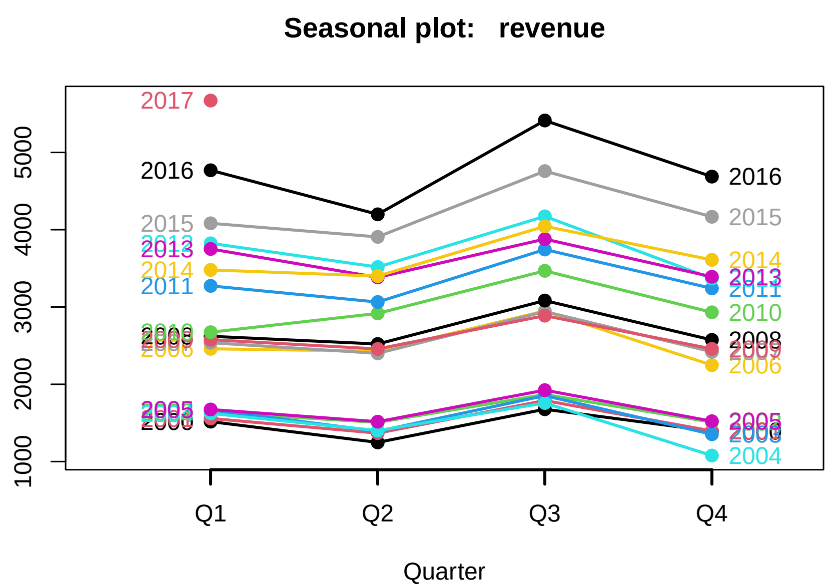
```
adidas_revenue <- read.csv("C:/Users/manal/Desktop/adidas_revenue1.csv")

### rename data frame
df=adidas_revenue

### define data as time series
df<-ts(df, start=c(2000,1), end=c(2017,1), frequency = 4)

### extract Adidas revenue data
y=df[,2]
y
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2000 1517 1248 1677 1393
## 2001 1558 1368 1790 1396
## 2002 1638 1507 1868 1510
## 2003 1669 1392 1853 1353
## 2004 1623 1401 1758 1078
## 2005 1674 1516 1924 1522
## 2006 2459 2428 2949 2248
## 2007 2538 2400 2941 2420
## 2008 2621 2521 3083 2574
## 2009 2577 2457 2888 2458
## 2010 2674 2917 3468 2931
## 2011 3273 3064 3744 3241
## 2012 3824 3517 4173 3369
## 2013 3751 3383 3879 3391
## 2014 3480 3400 4044 3610
## 2015 4083 3907 4758 4167
## 2016 4769 4199 5413 4687
## 2017 5671
```

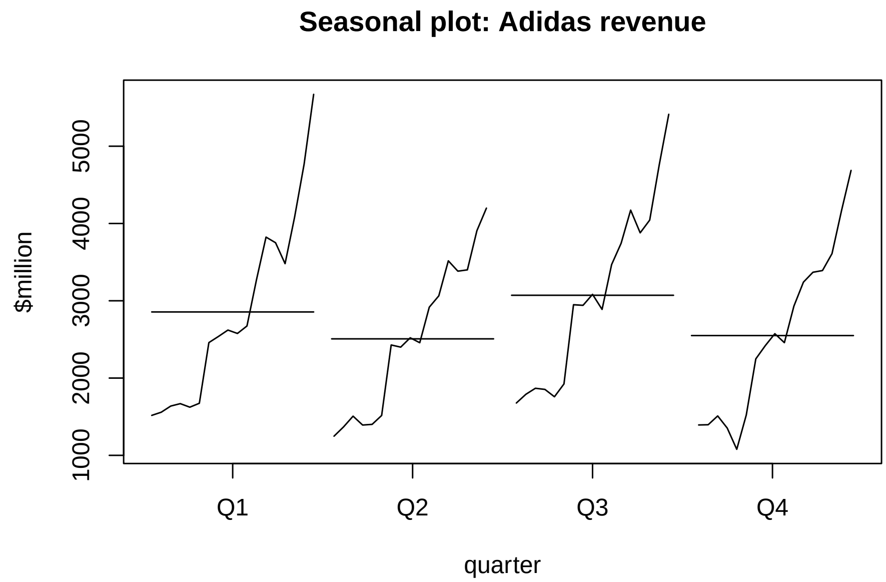
```
plot(y,main="Adidas Revenue", xlab="year", ylab="$millions", lwd=3)
```




```
seasonplot(y,main="Seasonal plot:   revenue",year.labels = TRUE,
           year.labels.left = TRUE,col=1:20, pch=19,lwd=2)
```



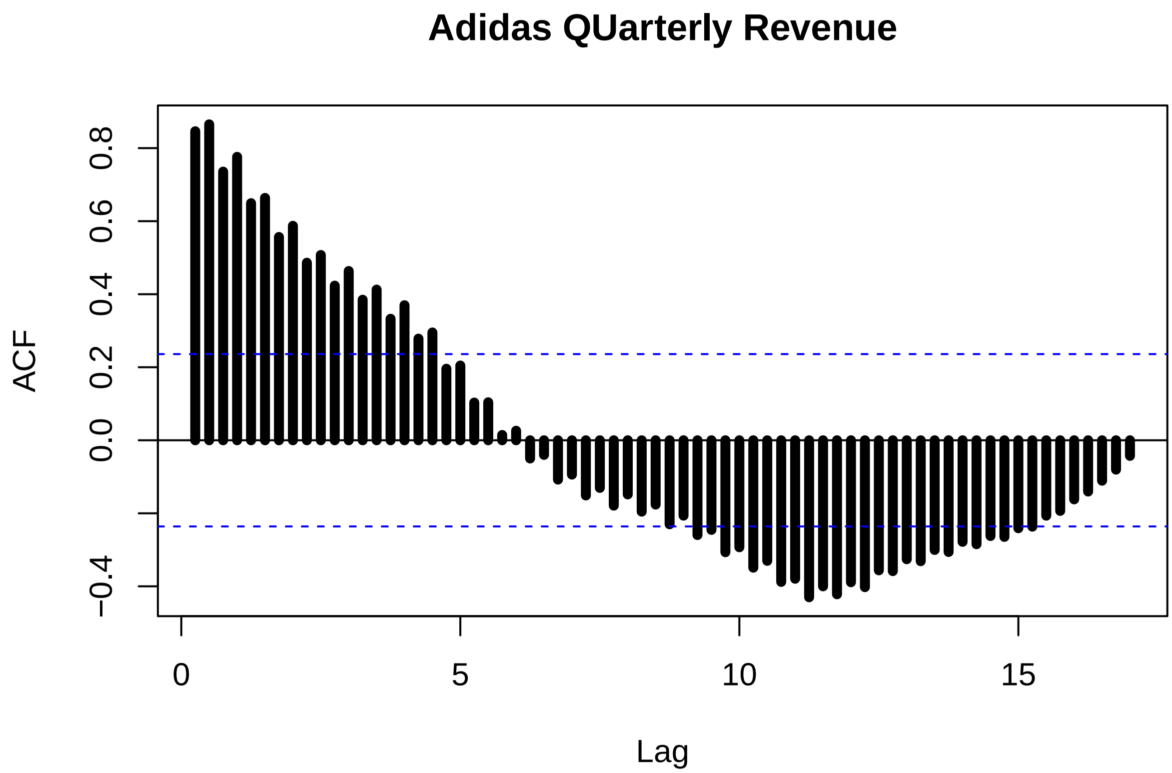

```
monthplot(y, main="Seasonal plot: Adidas revenue", xlab="quarter", ylab = "$million")
```



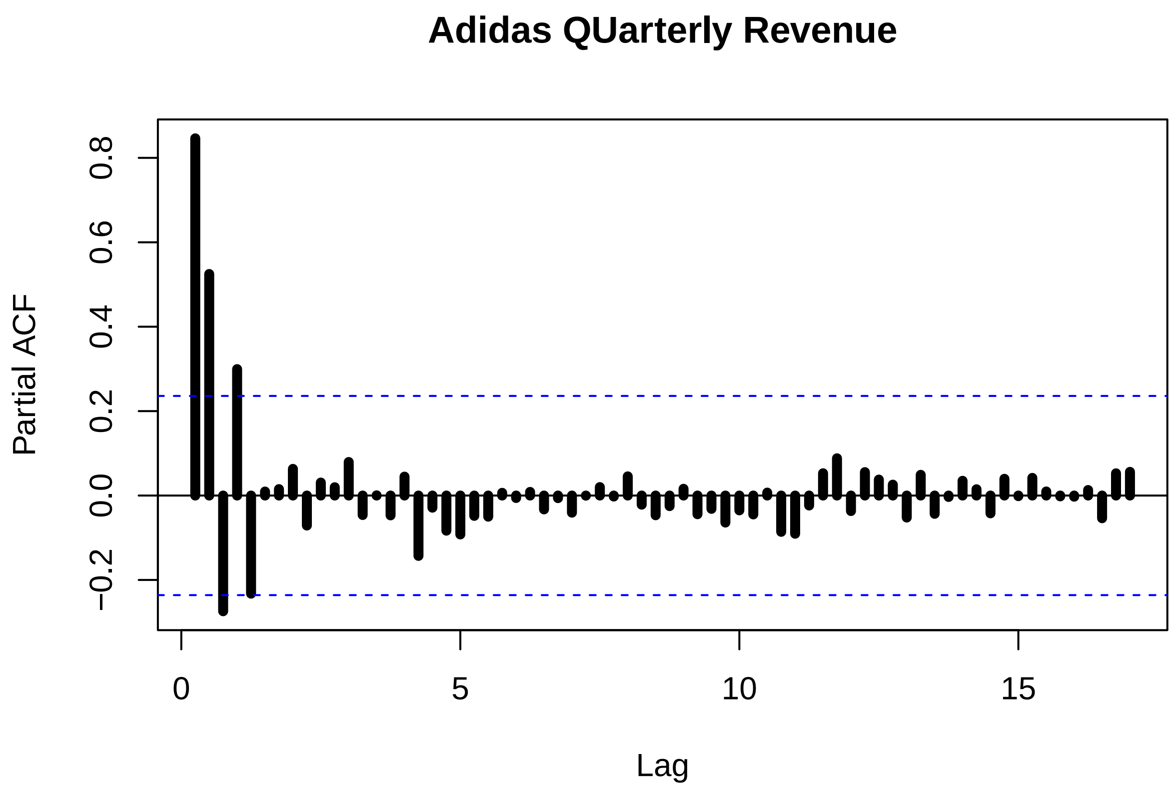
The quarterly sales of ADIDAS have strong growing trends throughout the years except for a visible decrease in the year of 2004. There are also some fluctuations every year which indicates a possible seasonal pattern that increases in size as the level of the series increase. The centered moving average plot removes the influence of seasonality and makes the increasing trend more obvious These pattern indicate that a good forecast of this series would need to capture both the trend and seasonality.

Looking at the two plots above, ADIDAS performs differently in different seasons. It is clear that there is a large jump in sales in January each year. The Seasonal graph also shows that there is a big gap between 2005 and 2006, meaning there was a huge increase between these two years.

```
acf(y, lwd=5, main="Adidas QUarterly Revenue", lag.max = 100)
```




```r
pacf(y, lwd=5, main="Adidas QUarterly Revenue", lag.max = 100)
```



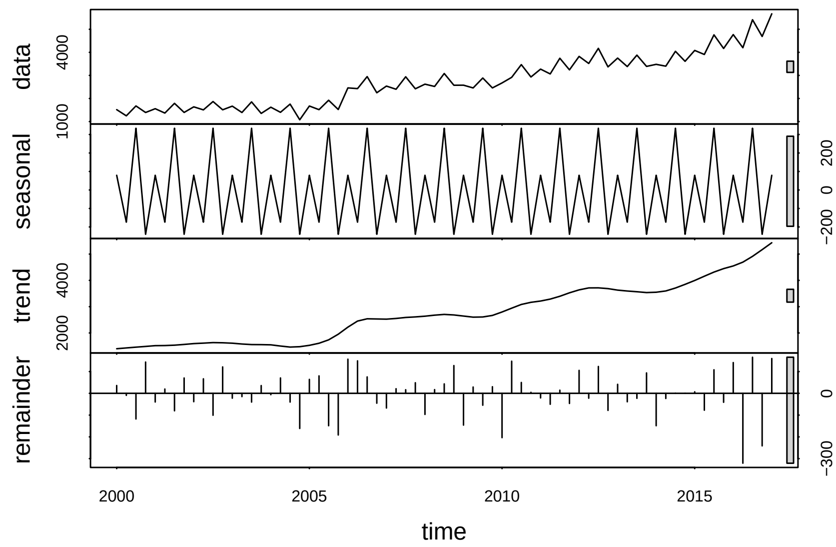

```
eacf(y)
```

```
## 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 x x x x x x x x x  x  x  x
## 2 x x o x o x o x o x o  o  o  x
## 3 x x o x o x o x o x o  x  o  x
## 4 x o o o o x o o o x x  x  o  o
## 5 o x o o o o o o o o o  x  o  o
## 6 o o o o o o o o o o o  x  o  o
## 7 x x o o o o o o o o o  x  o  o
```

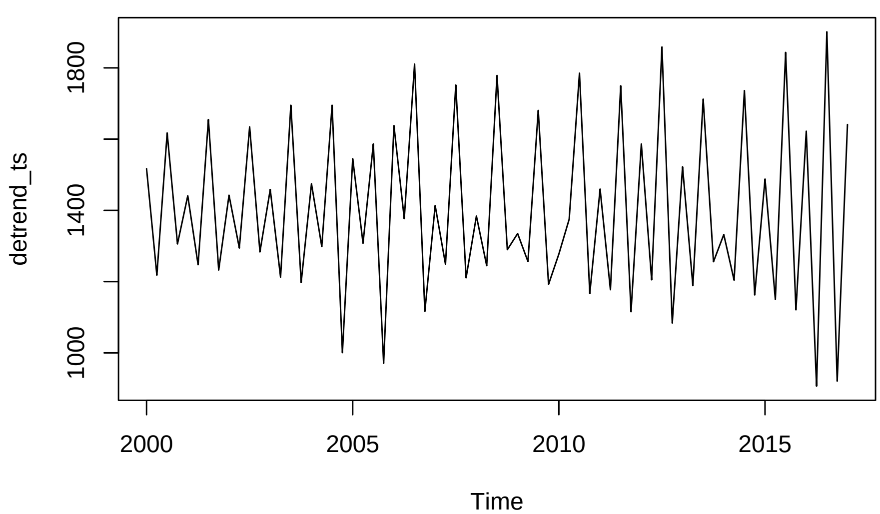
Methodology. Exploratory Data Analysis: Here, we plot the time series plot, auto-correlation function(ACF) plot, partial auto-correlation function(PACF) plot, and EACF of the time series.

The ACF graph indicates a strong trend component of the dataset. Although the tend pattern is getting weak as the number of lags increases, it is still significant when the lag is as large as 20. Additionally, r4, r8, r12 and r16 is slightly higher than their neighbors. This is because the seasonal pattern of the data: the peaks tend to be four quarters apart (which is the Q3 of each year)

```
plot(stl(y, s.window = "periodic"))
```




```
trend = stl(y, s.window = "periodic")$time.series[,2]
detrend_ts = y - (trend - trend[1])
plot(detrend_ts)
```



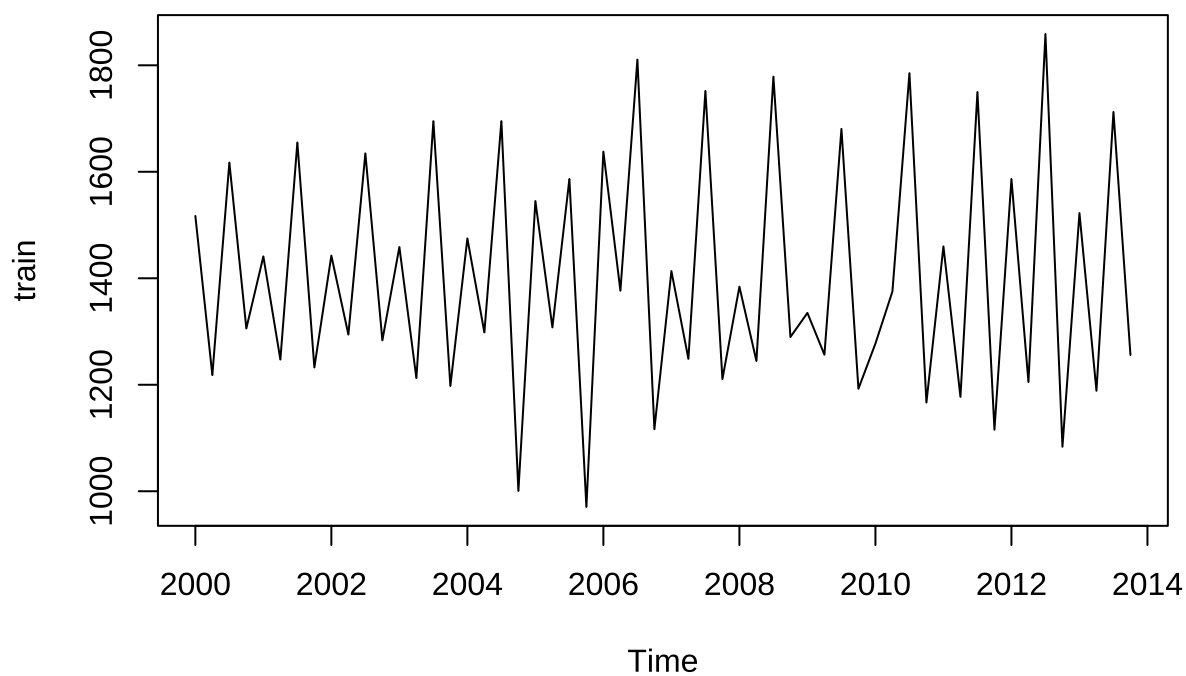

```
train<-window(detrend_ts,start=c(2000,1), end=c(2013,4))
test<-window(detrend_ts,start=c(2014,1))
```

we apply detrending to remove trends from the TS and then use Augmented Dicky Fuller test to test for stationarity.

```r
adf.test(train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train
## Dickey-Fuller = -4.0832, Lag order = 3, p-value = 0.01242
## alternative hypothesis: stationary
```

```r
plot(train)
```

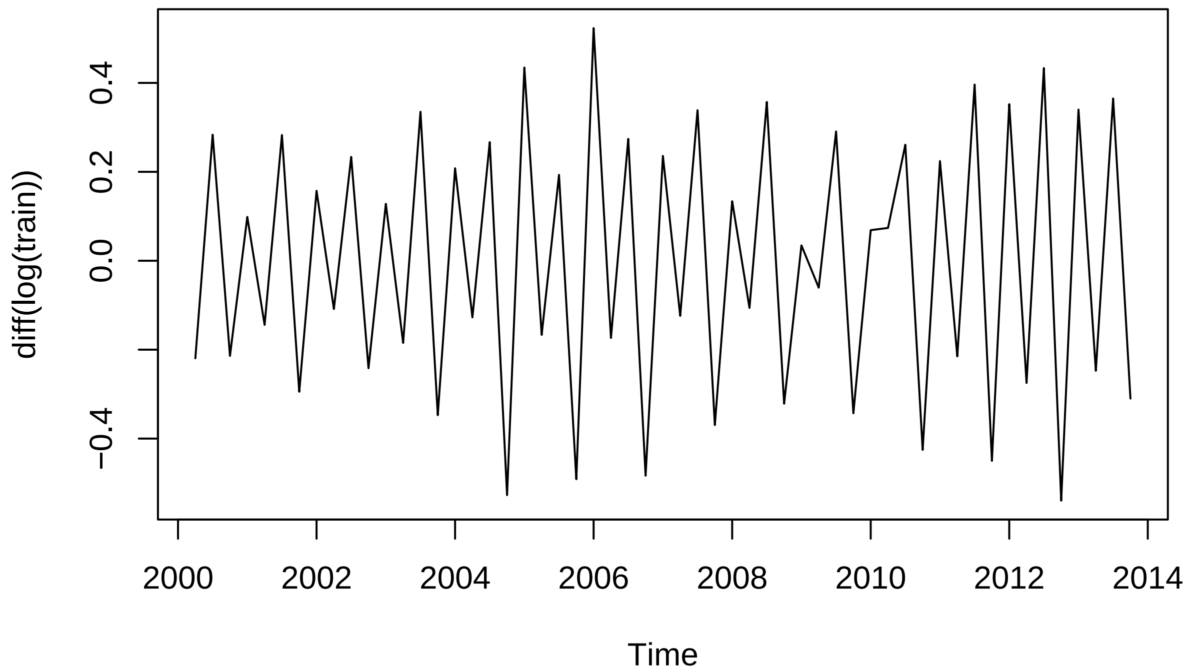



```r
adf.test(diff(log(train)))
```

```
## Warning in adf.test(diff(log(train))): p-value smaller than printed p-value
```

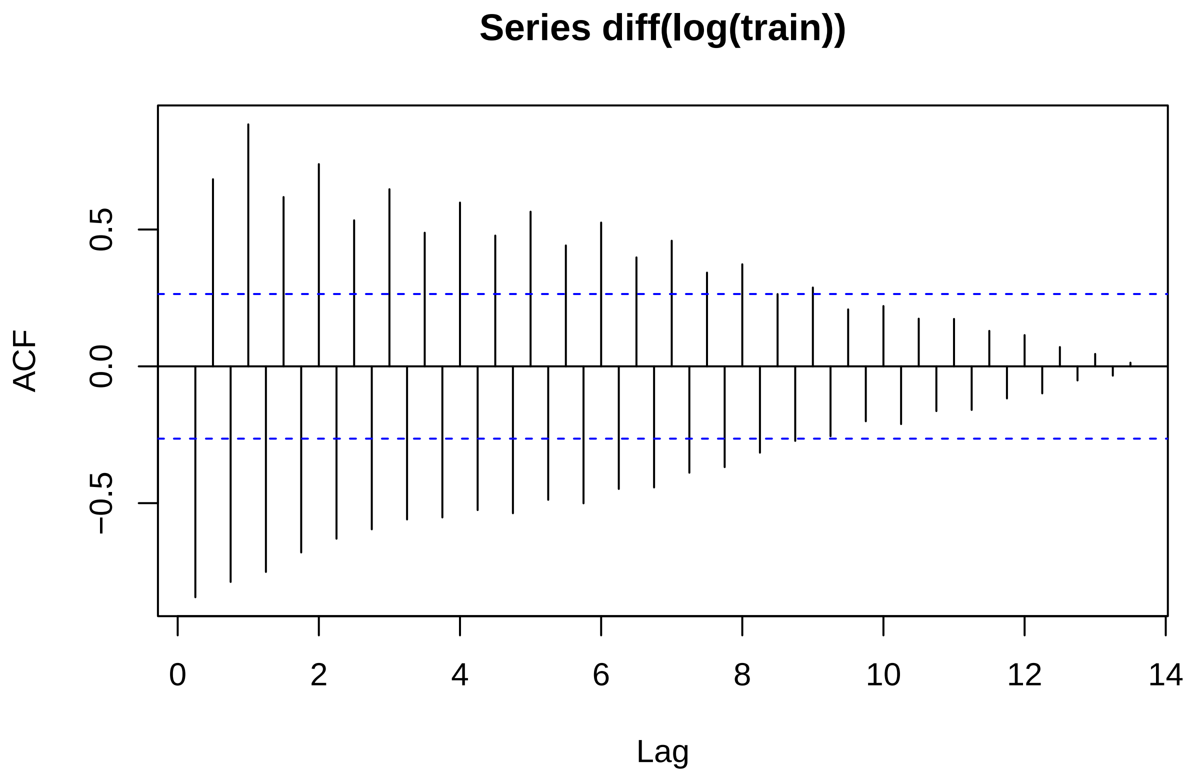
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(log(train))
## Dickey-Fuller = -6.7252, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
plot(diff(log(train)))
```

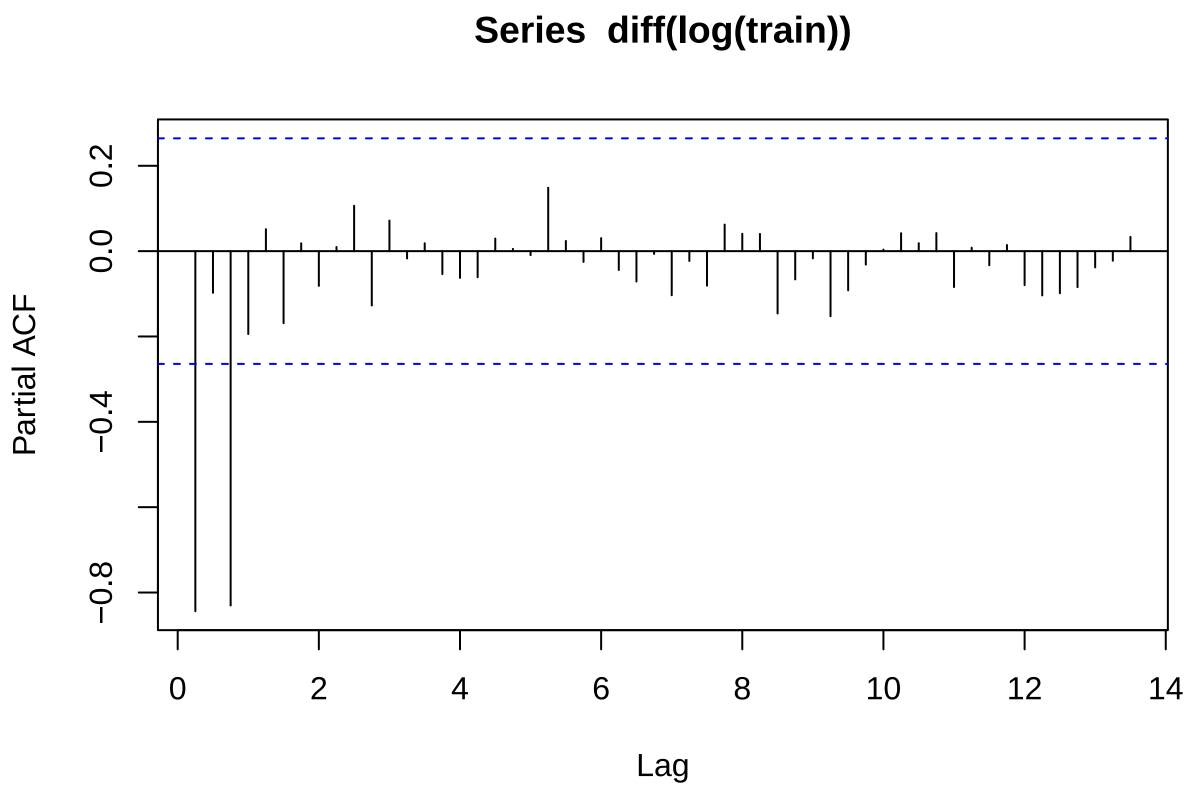


Stationarity: Here, we test for stationarity of the TS by Augmented Dicky Fuller test. If the TS is stationary we proceed to the next step, otherwise if the TS is non-stationary, we apply differencing,transforming,to remove stationarity from the TS and then use Augmented Dicky Fuller test to test for stationarity again On comparing the plot of train and diff(log(train)) we can see that the latter plot is centered around 0 with +-0.4 variance. Hence we are choosing this transformation.

```
acf(diff(log(train)), lag.max=100)
```




```
pacf(diff(log(train)), lag.max=100)
```




```
eacf(log(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 o x o x o x o x o x o  x  o  x
## 2 o x o x o x o x o x o  x  o  x
## 3 x o o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 x o o o o o o o o o o  o  o  o
## 7 x o o o o o o o o o o  o  o  o
```

Here, we find the models by checking the number of bars in ACF plot to find theMA(q) and number of bars in PACF to find the AR(p) and then cross check it with EACF to find ARMA(p,q) model.

```
qtr_forecast <- Arima(log(train), order=c(1,0,0),
                      seasonal=list(order=c(1,1,0), period=4), method="CSS-ML")
summary(qtr_forecast)
```

```
## Series: log(train)
## ARIMA(1,0,0)(1,1,0)[4]
##
## Coefficients:
```

```
##           ar1      sar1
##       -0.1564  -0.1857
## s.e.   0.1526   0.1500
##
## sigma^2 = 0.005082:  log likelihood = 64.51
## AIC=-123.01   AICc=-122.51   BIC=-117.16
##
## Training set error measures:
##                        ME       RMSE        MAE          MPE      MAPE
## Training set 7.196602e-05 0.06736216 0.05224137 -0.004742922 0.7259858
##                   MASE        ACF1
## Training set 0.9324321 -0.02832863
```

```
qtr_forecast <- Arima(log(train), order=c(2,0,0),
                      seasonal=list(order=c(2,1,0), period=4), method="CSS-ML")
summary(qtr_forecast)
```

```
## Series: log(train)
## ARIMA(2,0,0)(2,1,0)[4]
##
## Coefficients:
##           ar1      ar2     sar1     sar2
##       -0.1795  -0.1913  -0.2321  -0.0777
## s.e.   0.1642   0.1450   0.1609   0.1425
##
## sigma^2 = 0.005061:  log likelihood = 65.61
## AIC=-121.21   AICc=-119.91   BIC=-111.45
##
## Training set error measures:
##                         ME       RMSE        MAE         MPE      MAPE     MASE
## Training set -0.0004015498 0.06586122 0.05044917 -0.01179306 0.7009798 0.900444
##                     ACF1
## Training set -0.05263869
```

```
qtr_forecast <- Arima(log(train), order=c(1,0,2),
                      seasonal=list(order=c(1,1,2), period=4), method="CSS-ML")
summary(qtr_forecast)
```

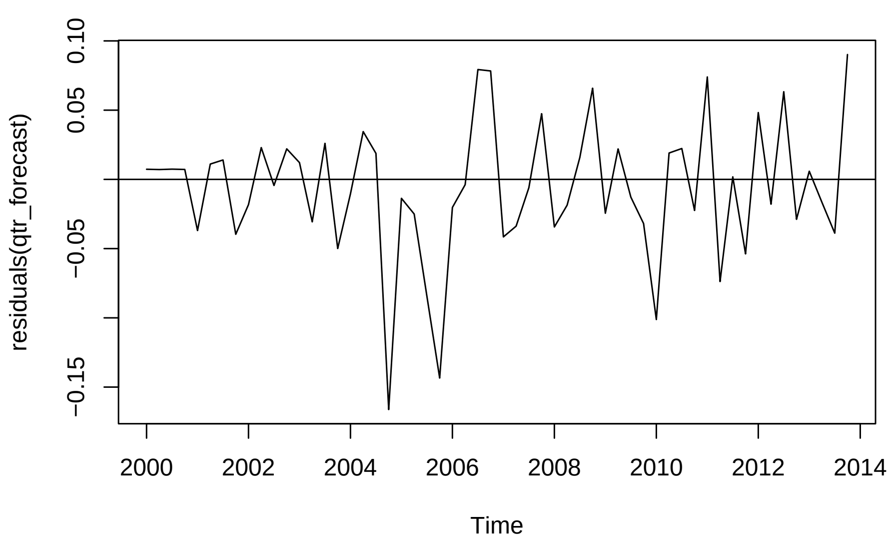
```
## Series: log(train)
## ARIMA(1,0,2)(1,1,2)[4]
##
## Coefficients:
##          ar1      ma1      ma2    sar1     sma1     sma2
##       0.3306  -0.7780  -0.2220  0.3486  -0.8239  -0.1761
## s.e.  0.2213   0.2376   0.2146  0.2411   0.2587   0.2083
##
## sigma^2 = 0.002991:  log likelihood = 74.31
## AIC=-134.61   AICc=-132.07   BIC=-120.95
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE      MAPE      MASE
## Training set -0.006723581 0.04956536 0.03673744 -0.0994821 0.5116394 0.6557097
##                      ACF1
## Training set -0.001735337
```

```r
qtr_forecast <- Arima(log(train), order=c(2,0,2),
                      seasonal=list(order=c(2,1,2), period=4), method="CSS-ML")
summary(qtr_forecast)
```

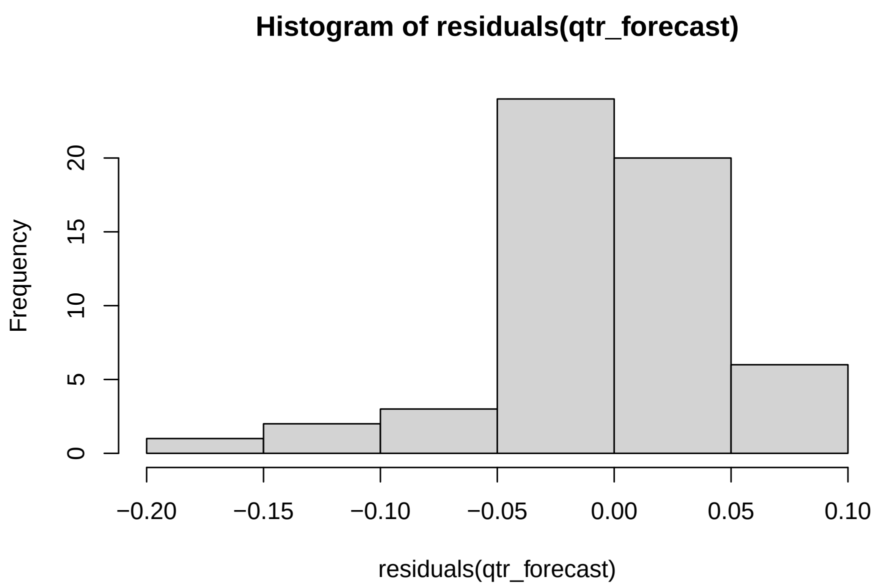
```
## Series: log(train)
## ARIMA(2,0,2)(2,1,2)[4]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1    sar2     sma1     sma2
##       0.6425  -0.1656  -1.1381  0.1383  -0.4644  0.5262  -0.0262  -0.9735
## s.e.  1.0375   0.5100   1.0340  1.0269   0.1847  0.1664   0.2500   0.2480
##
## sigma^2 = 0.003094:  log likelihood = 74.43
## AIC=-130.87   AICc=-126.58   BIC=-113.31
##
## Training set error measures:
##                       ME     RMSE        MAE        MPE      MAPE      MASE
## Training set -0.00678742 0.049305 0.03621685 -0.1000019 0.5051683 0.6464179
##                     ACF1
## Training set -0.02046693
```

Parameter Estimation : we compare multiple models based on the AIC and BIC scores, log likelihood values, Adjusted Pearson Goodness-of-Fit Test, Ljung-Box Test, ARCH LM Tests etc. and select the best ARIMA model from them.

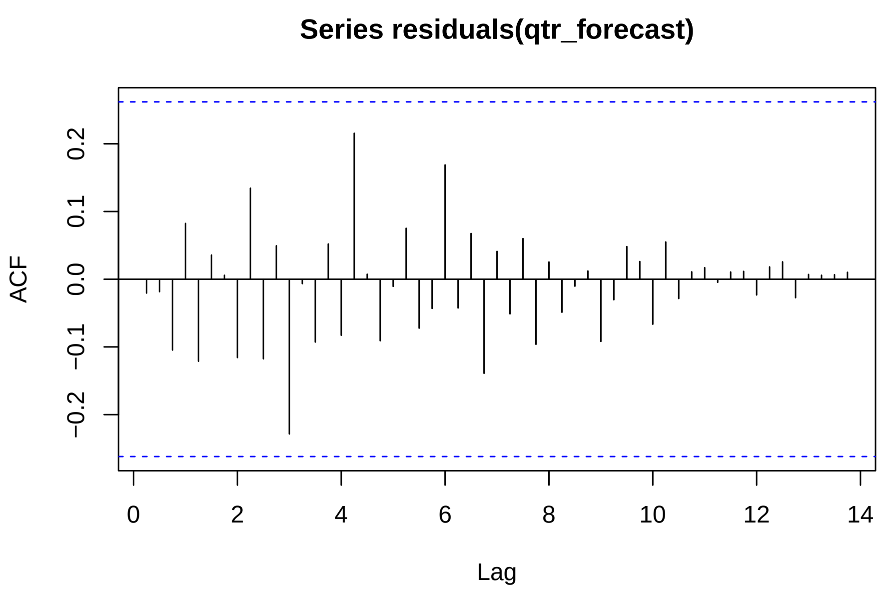
```r
qtr_forecast <- Arima(log(train), order=c(2,0,2),
                      seasonal=list(order=c(2,1,2), period=4), method="CSS-ML")
plot(residuals(qtr_forecast))
abline(h=0)
```



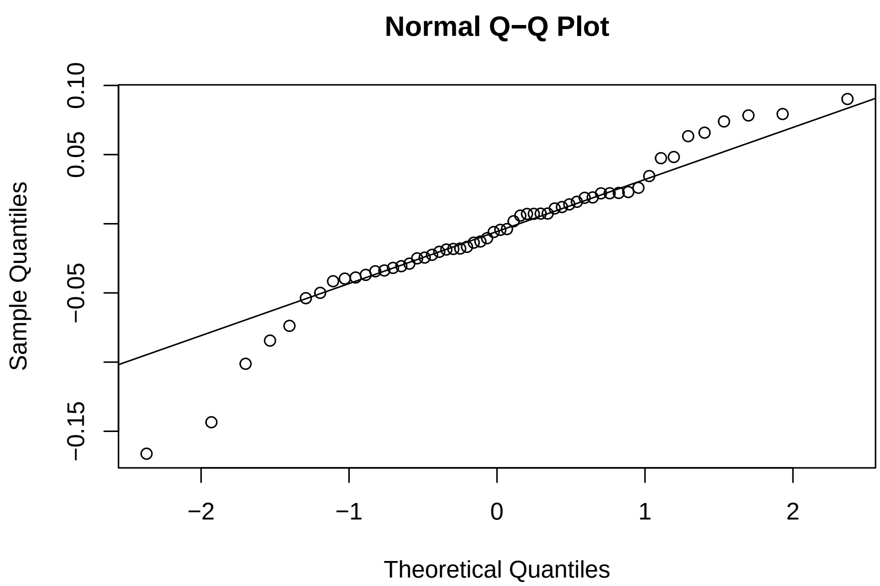
```r
hist(residuals(qtr_forecast))
```




```r
acf(residuals(qtr_forecast), lag.max=100)
```



```
qqnorm(residuals(qtr_forecast));qqline(residuals(qtr_forecast))
```




```
shapiro.test(residuals(qtr_forecast))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(qtr_forecast)
## W = 0.9472, p-value = 0.01597
```

```
Box.test(residuals(qtr_forecast), type='Ljung')
```

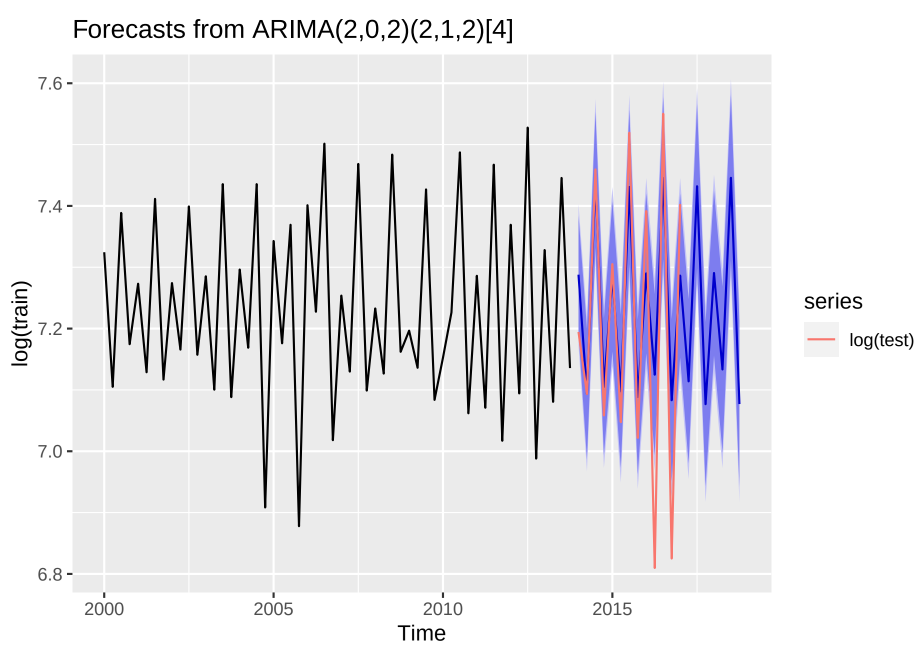
```
##
##  Box-Ljung test
##
## data:  residuals(qtr_forecast)
## X-squared = 0.024738, df = 1, p-value = 0.875
```

According to shapiro-wilk test p value is less than 0.05 hence it is normal, but qq plot says that it has alot of outliers. Here, we use ARIMA models here which are selected by taking difference of log of TS and then using this new TS to find the best model based on ACF, PACF and EACF plots. Residual Analysis : Here, we analyse the residuals of the bets model by making ACF plot, Histogram, QQ plot, Residual plot and check for Shapiro-Wilk test and Ljung-Box test.

```
library(ggplot2)
qtr_forecast <- Arima(log(train), order=c(2,0,2),
                      seasonal=list(order=c(2,1,2), period=4), method="CSS-ML")
summary(qtr_forecast)
```

```
## Series: log(train)
## ARIMA(2,0,2)(2,1,2)[4]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1    sar2     sma1     sma2
##       0.6425  -0.1656  -1.1381  0.1383  -0.4644  0.5262  -0.0262  -0.9735
## s.e.  1.0375   0.5100   1.0340  1.0269   0.1847  0.1664   0.2500   0.2480
##
## sigma^2 = 0.003094:  log likelihood = 74.43
## AIC=-130.87   AICc=-126.58   BIC=-113.31
##
## Training set error measures:
##                        ME      RMSE        MAE        MPE      MAPE      MASE
## Training set -0.00678742 0.049305 0.03621685 -0.1000019 0.5051683 0.6464179
##                     ACF1
## Training set -0.02046693
```

```
qtr_forecast.new <- forecast(qtr_forecast, h=20, level=c(90,95))
autoplot(qtr_forecast.new,) + autolayer(log(test))
```



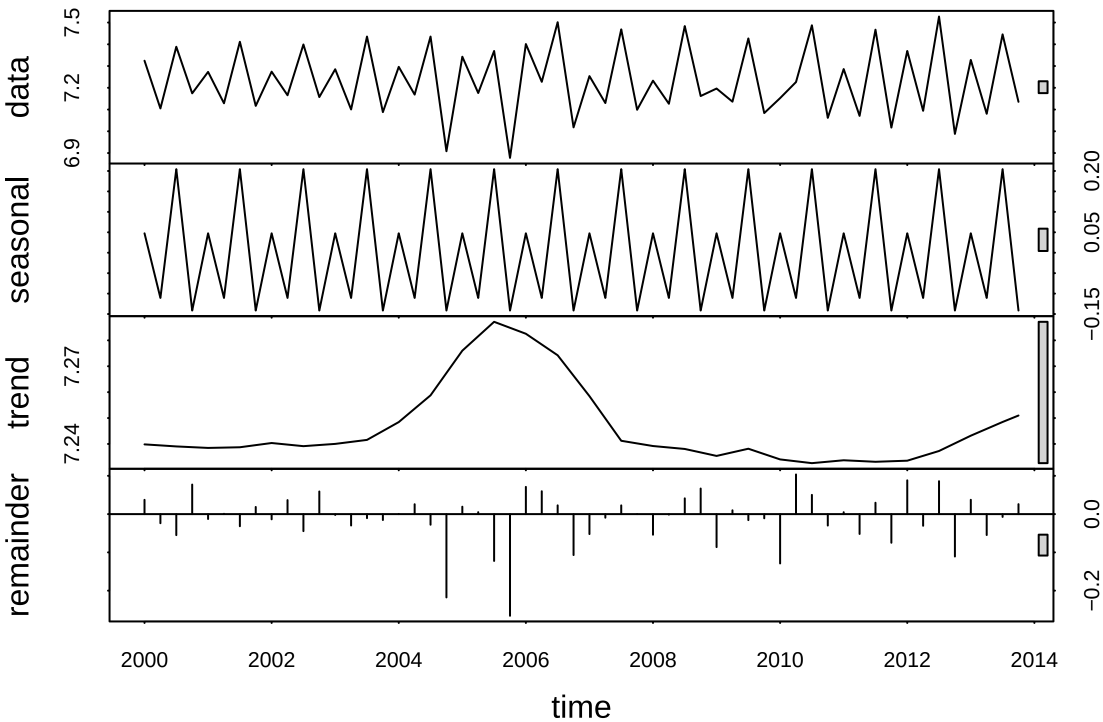

```
summary(qtr_forecast.new)
```

```
##
## Forecast method: ARIMA(2,0,2)(2,1,2)[4]
##
## Model Information:
## Series: log(train)
## ARIMA(2,0,2)(2,1,2)[4]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sar1    sar2     sma1     sma2
##       0.6425  -0.1656  -1.1381  0.1383  -0.4644  0.5262  -0.0262  -0.9735
```

```
## s.e.   1.0375   0.5100   1.0340  1.0269   0.1847  0.1664   0.2500   0.2480
##
## sigma^2 = 0.003094:  log likelihood = 74.43
## AIC=-130.87   AICc=-126.58   BIC=-113.31
##
## Error measures:
##                       ME     RMSE        MAE        MPE      MAPE      MASE
## Training set -0.00678742 0.049305 0.03621685 -0.1000019 0.5051683 0.6464179
##                     ACF1
## Training set -0.02046693
##
## Forecasts:
##         Point Forecast    Lo 90    Hi 90    Lo 95    Hi 95
## 2014 Q1       7.287912 7.191012 7.384812 7.172449 7.403376
## 2014 Q2       7.093411 6.987136 7.199687 6.966776 7.220047
## 2014 Q3       7.442833 7.332061 7.553605 7.310840 7.574826
## 2014 Q4       7.105219 6.993712 7.216727 6.972350 7.238089
## 2015 Q1       7.285057 7.162582 7.407533 7.139119 7.430996
## 2015 Q2       7.097570 6.972754 7.222386 6.948843 7.246298
## 2015 Q3       7.430922 7.305023 7.556820 7.280904 7.580939
## 2015 Q4       7.088512 6.962503 7.214521 6.938364 7.238661
## 2016 Q1       7.290021 7.159497 7.420546 7.134492 7.445551
## 2016 Q2       7.124813 6.993253 7.256373 6.968050 7.281576
## 2016 Q3       7.445325 7.313281 7.577369 7.287985 7.602665
## 2016 Q4       7.083203 6.951124 7.215282 6.925821 7.240585
## 2017 Q1       7.286378 7.152863 7.419893 7.127285 7.445471
## 2017 Q2       7.113973 6.980103 7.247843 6.954457 7.273489
## 2017 Q3       7.432099 7.298108 7.566090 7.272439 7.591759
## 2017 Q4       7.076766 6.942857 7.210675 6.917204 7.236328
## 2018 Q1       7.290656 7.155877 7.425435 7.130057 7.451255
## 2018 Q2       7.133344 6.998272 7.268417 6.972395 7.294293
## 2018 Q3       7.445826 7.310638 7.581014 7.284739 7.606912
## 2018 Q4       7.076964 6.941818 7.212111 6.915927 7.238002
```

Taking logs to reduce the variability of the model, we can see that the forecast is flatter then the original ARIMA model. The plot does not catch the trend of data, which can influence the accuracy.

```r
y.stl <- stl(log(train), t.window=15, s.window="periodic", robust=TRUE)
plot(y.stl)
```

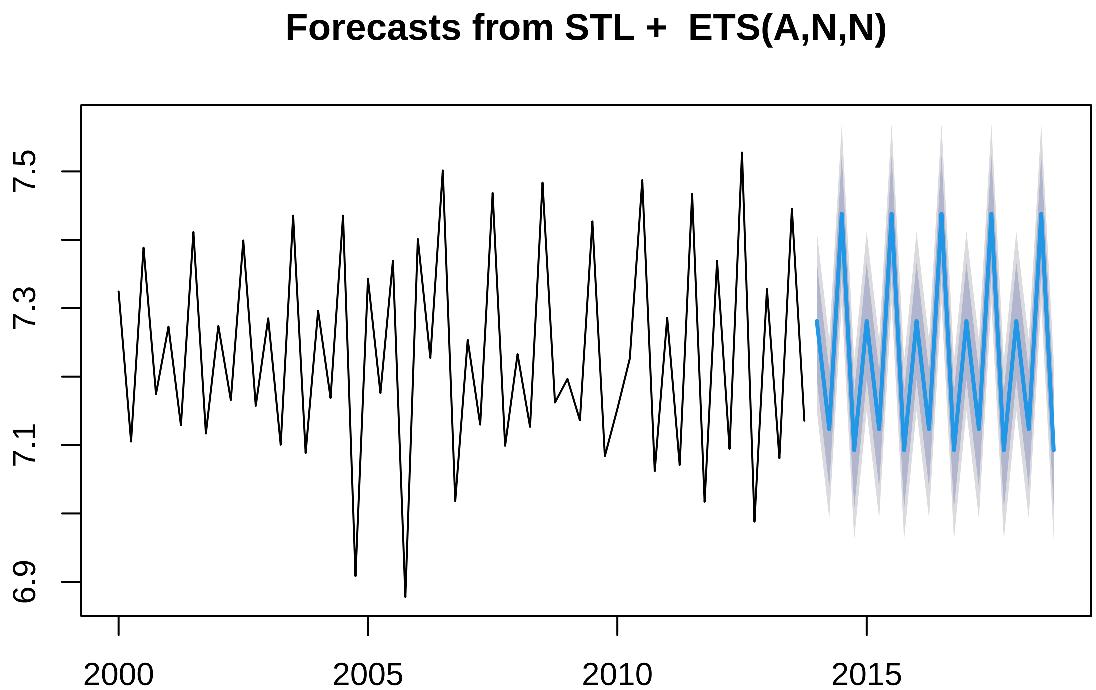



```
fit.stl <- forecast(y.stl, method="ets", h=20)
summary(fit.stl)
```

```
##
## Forecast method: STL +  ETS(A,N,N)
##
## Model Information:
## ETS(A,N,N)
##
## Call:
##  ets(y = na.interp(x), model = etsmodel, allow.multiplicative.trend = allow.multiplicative.trend)
##
##   Smoothing parameters:
##     alpha = 1e-04
##
##   Initial states:
##     l = 7.2338
##
##   sigma:  0.0668
##
##       AIC      AICc       BIC
## -73.66638 -73.20484 -67.59032
##
## Error measures:
##                         ME        RMSE         MAE          MPE        MAPE       MASE
```

```
## Training set -8.436276e-06 0.06561291 0.0487748 -0.0106131 0.6785731 0.870559
##                    ACF1
## Training set -0.2478556
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2014 Q1       7.281206 7.195576 7.366835 7.150247 7.412165
## 2014 Q2       7.123194 7.037565 7.208824 6.992236 7.254153
## 2014 Q3       7.438207 7.352578 7.523836 7.307248 7.569166
## 2014 Q4       7.092413 7.006784 7.178043 6.961455 7.223372
## 2015 Q1       7.281206 7.195576 7.366835 7.150247 7.412165
## 2015 Q2       7.123194 7.037565 7.208824 6.992236 7.254153
## 2015 Q3       7.438207 7.352578 7.523836 7.307248 7.569166
## 2015 Q4       7.092413 7.006784 7.178043 6.961455 7.223372
## 2016 Q1       7.281206 7.195576 7.366835 7.150247 7.412165
## 2016 Q2       7.123194 7.037565 7.208824 6.992236 7.254153
## 2016 Q3       7.438207 7.352578 7.523836 7.307248 7.569166
## 2016 Q4       7.092413 7.006784 7.178043 6.961455 7.223372
## 2017 Q1       7.281206 7.195576 7.366835 7.150247 7.412165
## 2017 Q2       7.123194 7.037565 7.208824 6.992236 7.254153
## 2017 Q3       7.438207 7.352578 7.523836 7.307248 7.569166
## 2017 Q4       7.092413 7.006784 7.178043 6.961455 7.223372
## 2018 Q1       7.281206 7.195576 7.366835 7.150247 7.412165
## 2018 Q2       7.123194 7.037565 7.208824 6.992236 7.254153
## 2018 Q3       7.438207 7.352578 7.523836 7.307248 7.569166
## 2018 Q4       7.092413 7.006784 7.178043 6.961455 7.223372
```

```r
plot(fit.stl)
lines(y)
```



The model uses STL and ETS to forecast.Forecasts of STL objects are obtained by applying a non-seasonal forecasting method to the seasonally adjusted data and re-seasonalizing using the last year of the seasonal component.The plot shows that the forecast follows the seasonal component pretty well but does not include trend component. For ETS Method, the best model is generated by system itself. Based on the title of plots, we can see that the best model found by system is (A, N, N). It means that the model has addictive error, which does not conform with the strong trend component.

Prediction : Since AIC, BIC values of ARIMA model are better than STL with ETS model,Hence, we choose ARIMA has the best model to forecast the revenue of Adidas