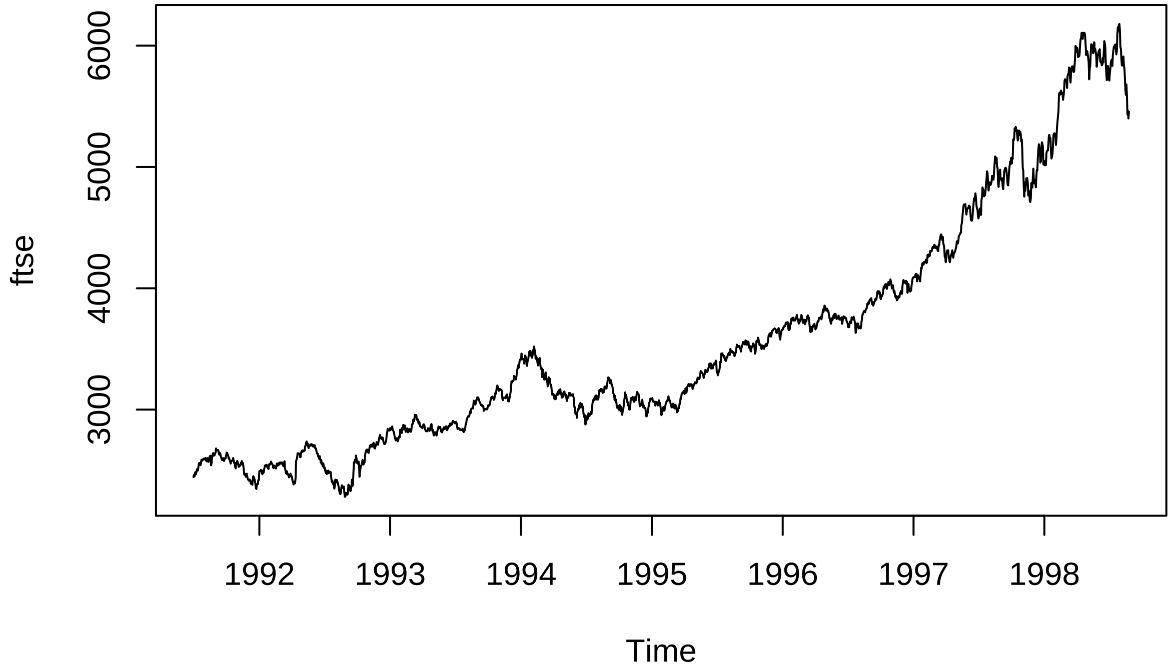
DataAnalyticsLab3

Siddharth.S.Chandran - 18BCE1003

26/02/2021

1. loading the dataset and visualizing it

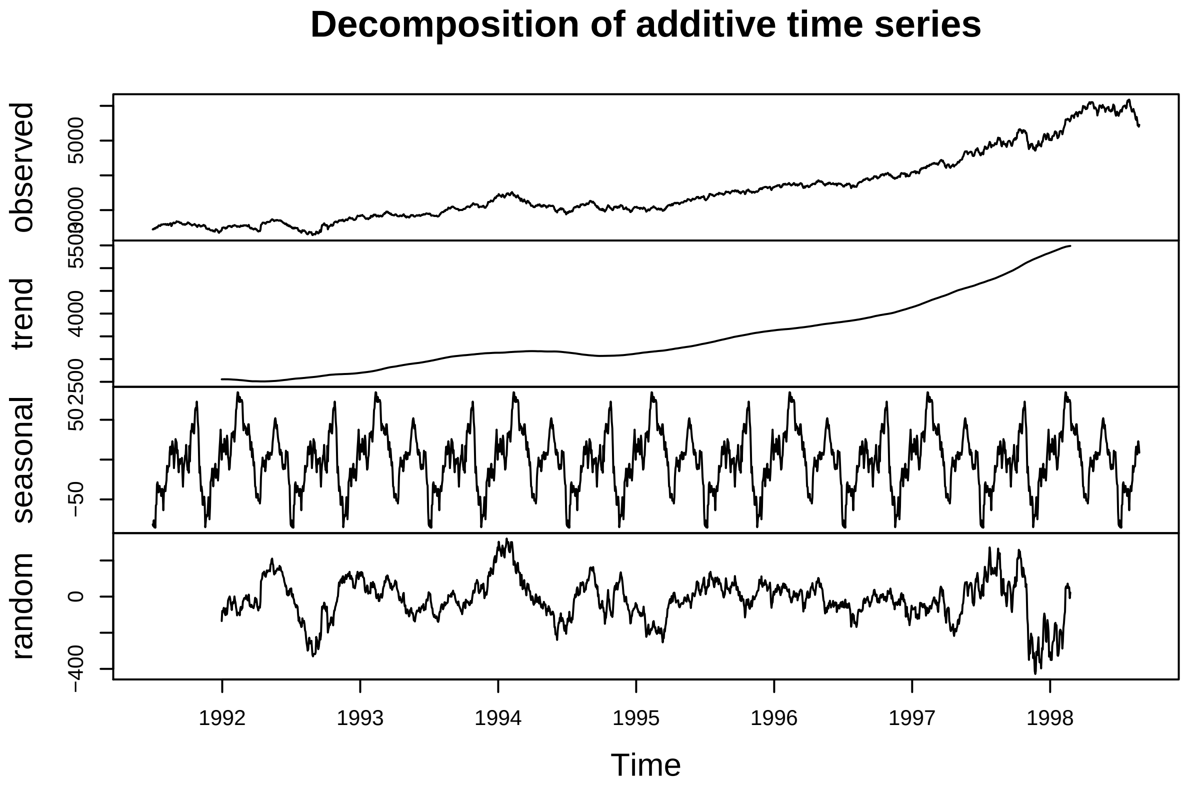
```
data(EuStockMarkets)
ftse=(EuStockMarkets[,4])
plot(ftse)
```



We can see that there is a clear upward trend, even though the index has experienced a sharp fall after 1998, we are intrested in forecasting if this fall is temperory or not that is predicting the rise and falls after 1998

We split the series into components

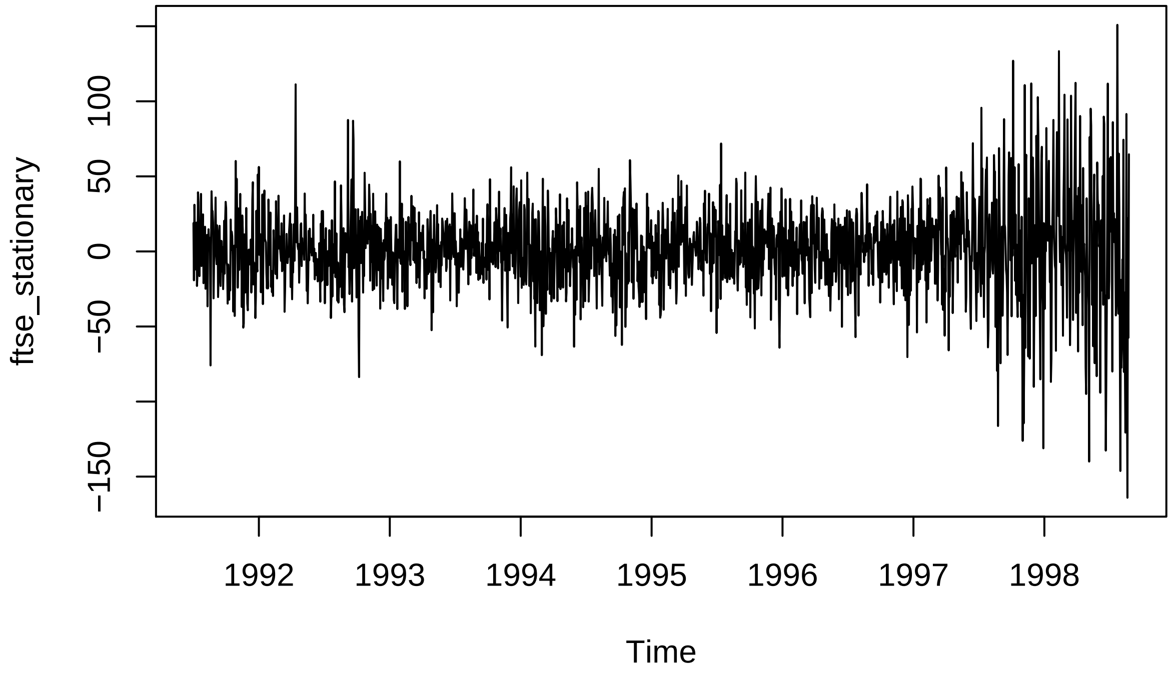
```
components.ts = decompose(ftse)
plot(components.ts)
```



We need to manage the series which are stationary which have constand mean and variance in order to bypass the main concern of time series: the impossibilty of observing at a fixed time t, all the ups and downs of the stock market.

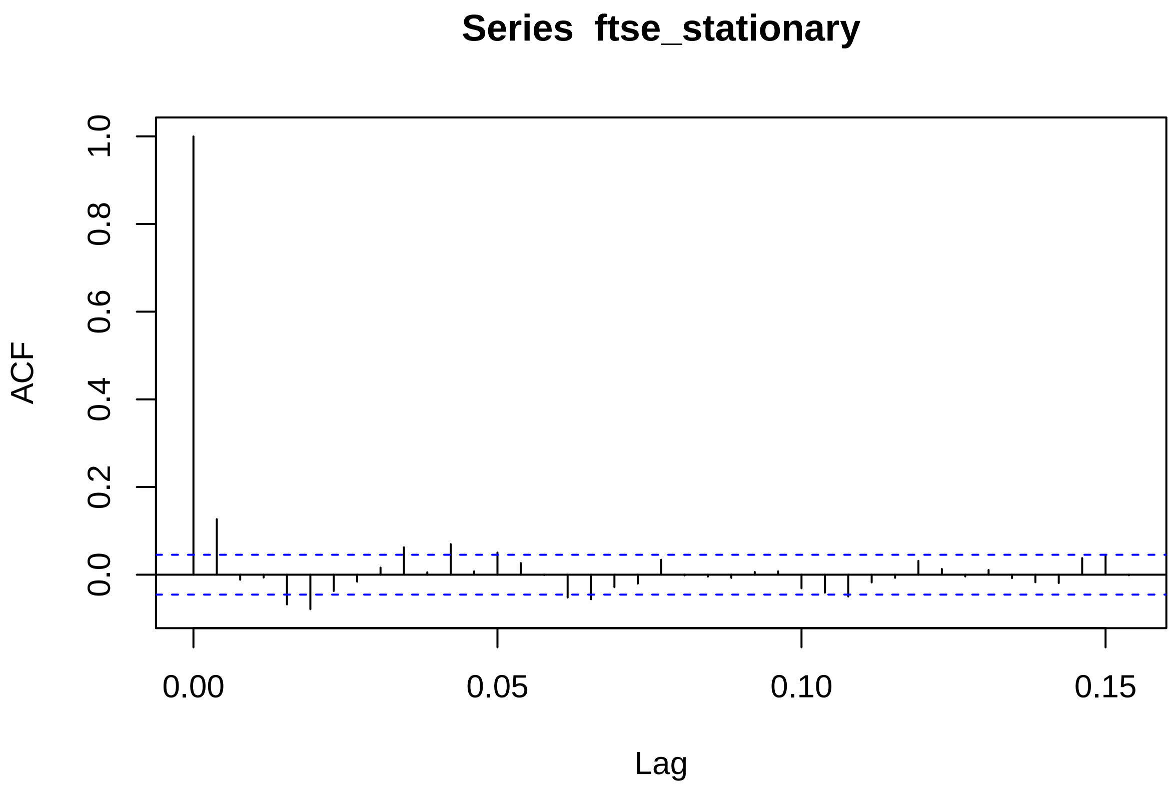
To make a time process stationary we take the first difference , once subtracted the seasonal component

```
x = ftse- components.ts$seasonal
ftse_stationary <- diff(x, differences=1)
plot(ftse_stationary)
```

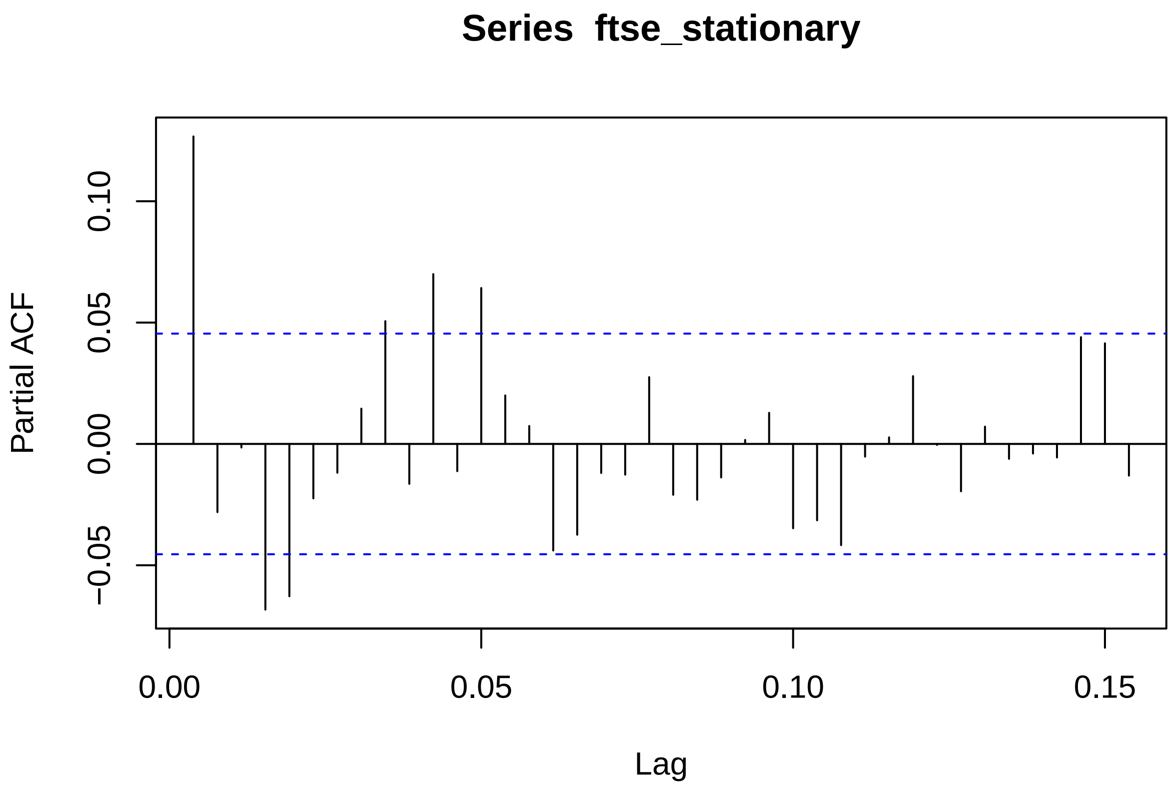


The graph looks more stationary now. Now we inspect the auto correlation function and partial autocorrelation function of the series

```r
acf(ftse_stationary,lag.max = 40)
```



```
pacf(ftse_stationary,lag.max = 40)
```



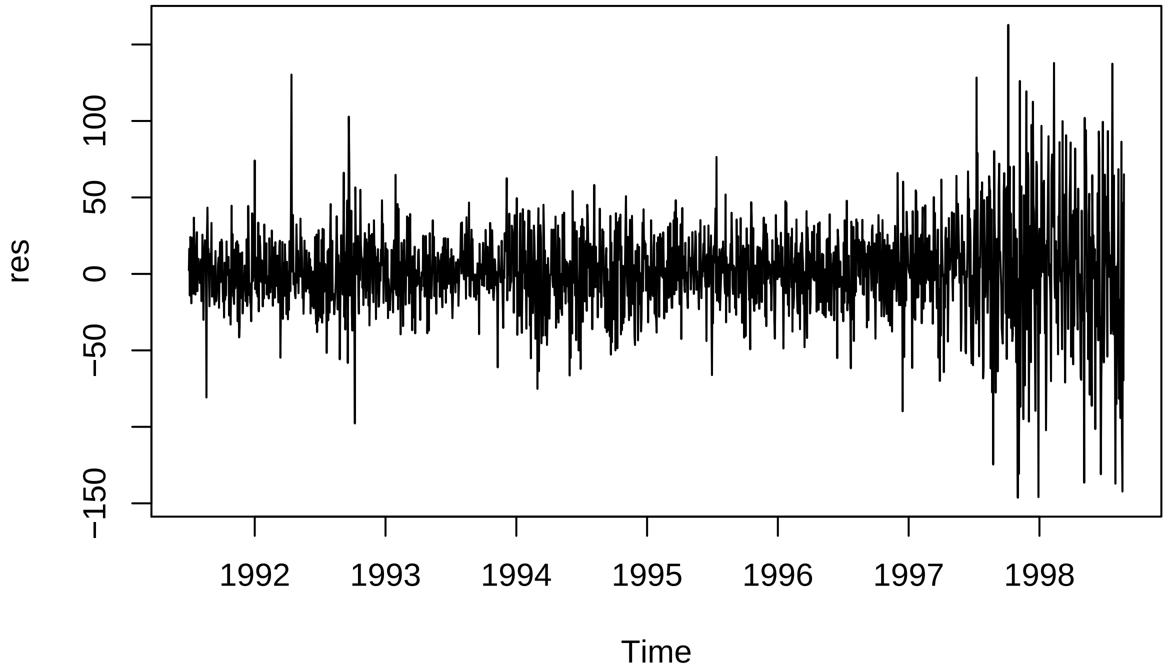
The shapes of the graphs ACF and PACF seem to suggest a Moving Average process

Let's also train a simple arima with low p,q,d values

```
fitARIMA = arima(ftse, order=c(1,1,1),seasonal = list(order = c(1,0,0), period = 12),method="ML")
```

After training the model we ll be looking into its residuals, for the model to be accurate the residuals need to be stationary.

```
res=fitARIMA$residuals
plot(res)
```



They look pretty stationary, but we need to test it. To do that, I will employ the Ljung-Box test, whose hypotheses are:

H0: The data are independently distributed

H1: The data are not independently distributed (hence, they exhibit serial correlation)

Let's examine the output:

```
library(FitAR)
```

```
## Warning: package 'FitAR' was built under R version 4.0.3

## Loading required package: lattice

## Loading required package: leaps

## Warning: package 'leaps' was built under R version 4.0.3

## Loading required package: ltsa

## Warning: package 'ltsa' was built under R version 4.0.3

## Loading required package: bestglm

## Warning: package 'bestglm' was built under R version 4.0.3
```

```
Box.test(res,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 0.0099996, df = 1, p-value = 0.9203
```

We can see that the p value excludes the possibility of serial correlations, as its greater than any significant level of alpha. So the first model was good enough to build a model returning independent residuals. Let's take a better model using auto.arima

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:FitAR':
##
##     BoxCox
```

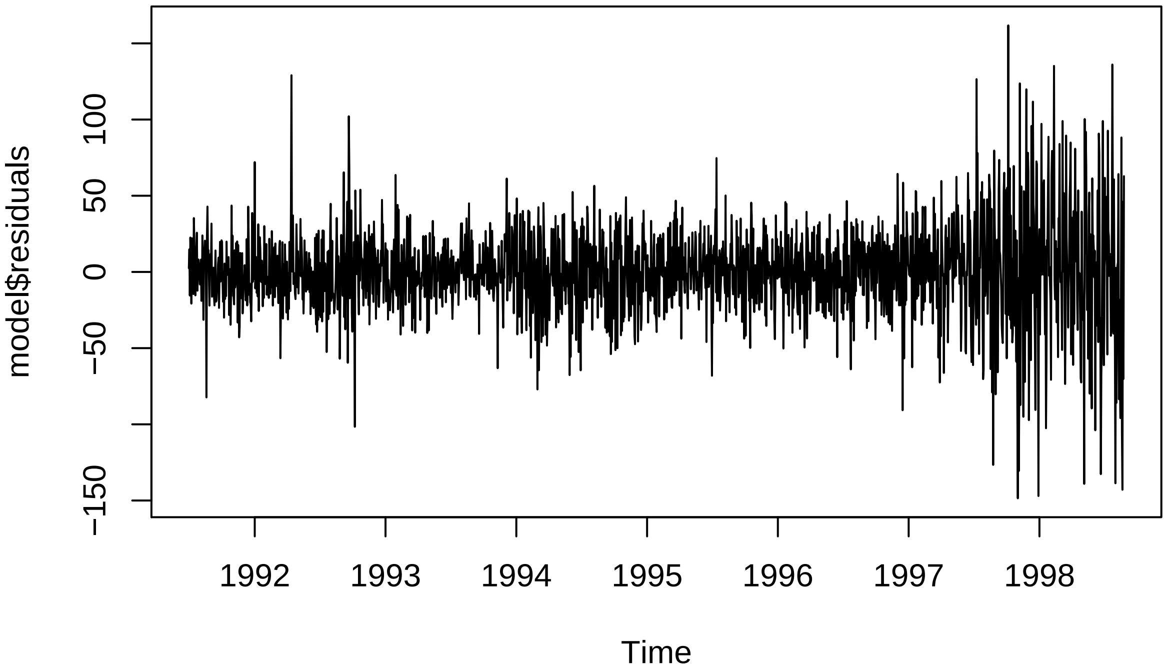
```
model<-auto.arima(ftse, trace=TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)(1,0,1)[260] with drift         : Inf
##  ARIMA(0,1,0)             with drift         : 17994.59
##  ARIMA(1,1,0)(1,0,0)[260] with drift         : Inf
##  ARIMA(0,1,1)(0,0,1)[260] with drift         : Inf
##  ARIMA(0,1,0)                                : 17997.78
##  ARIMA(0,1,0)(1,0,0)[260] with drift         : Inf
##  ARIMA(0,1,0)(0,0,1)[260] with drift         : Inf
##  ARIMA(0,1,0)(1,0,1)[260] with drift         : Inf
##  ARIMA(1,1,0)             with drift         : 17968.54
##  ARIMA(1,1,0)(0,0,1)[260] with drift         : Inf
##  ARIMA(1,1,0)(1,0,1)[260] with drift         : Inf
##  ARIMA(2,1,0)             with drift         : 17970.46
##  ARIMA(1,1,1)             with drift         : 17969.53
##  ARIMA(0,1,1)             with drift         : 17966.98
##  ARIMA(0,1,1)(1,0,0)[260] with drift         : Inf
##  ARIMA(0,1,1)(1,0,1)[260] with drift         : Inf
##  ARIMA(0,1,2)             with drift         : 17968.91
##  ARIMA(1,1,2)             with drift         : Inf
##  ARIMA(0,1,1)                                : 17969.15
```

```
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,1,1)             with drift         : 17973.84
##
##  Best model: ARIMA(0,1,1)             with drift
```

This model is an ARIMA(0,1,1) with drift which is the most accurate one

```
plot(model$residuals)
```



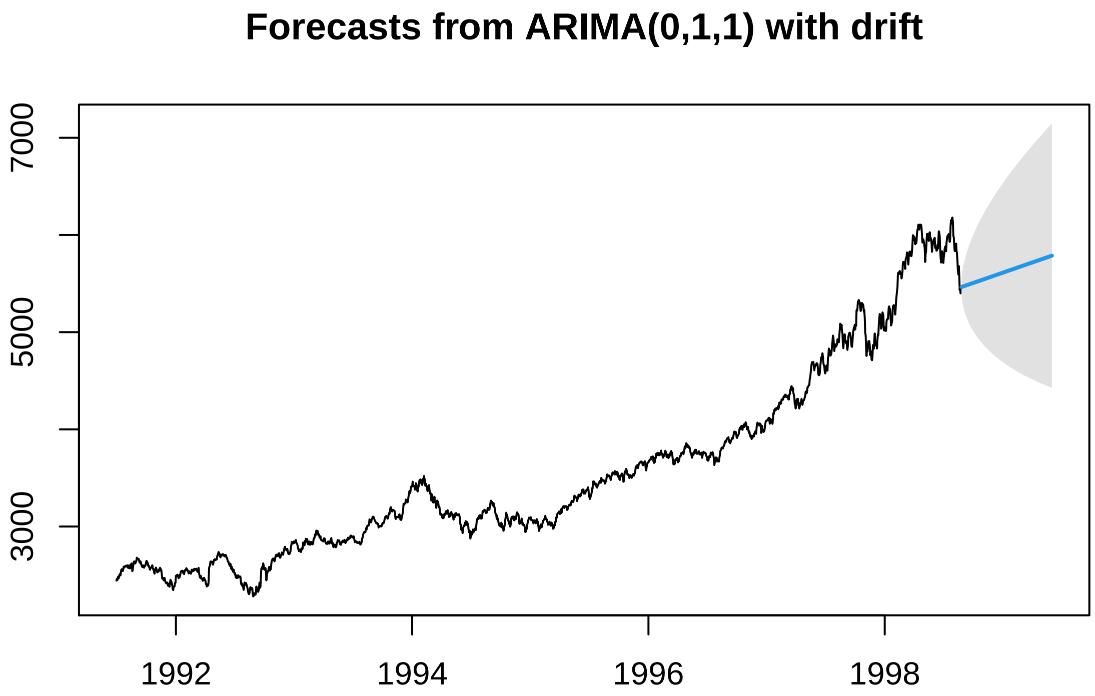
Running the Ljung-Box test for double checking the absence of serial correlation

```
Box.test(model$residuals,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  model$residuals
## X-squared = 0.0010769, df = 1, p-value = 0.9738
```

This model looks to be a good model, let’s predict the future outcomes

```
predicted_values = forecast(model,h=200, level=c(99.5))
plot(predicted_values)
```



The model seems to suggest that the model is increasing after 1998.