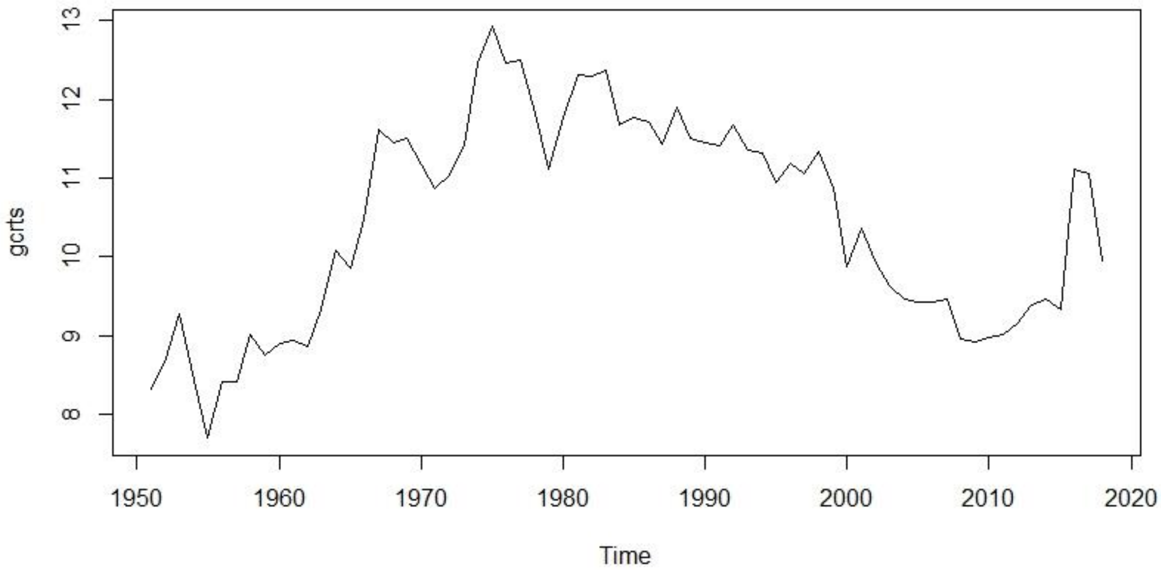
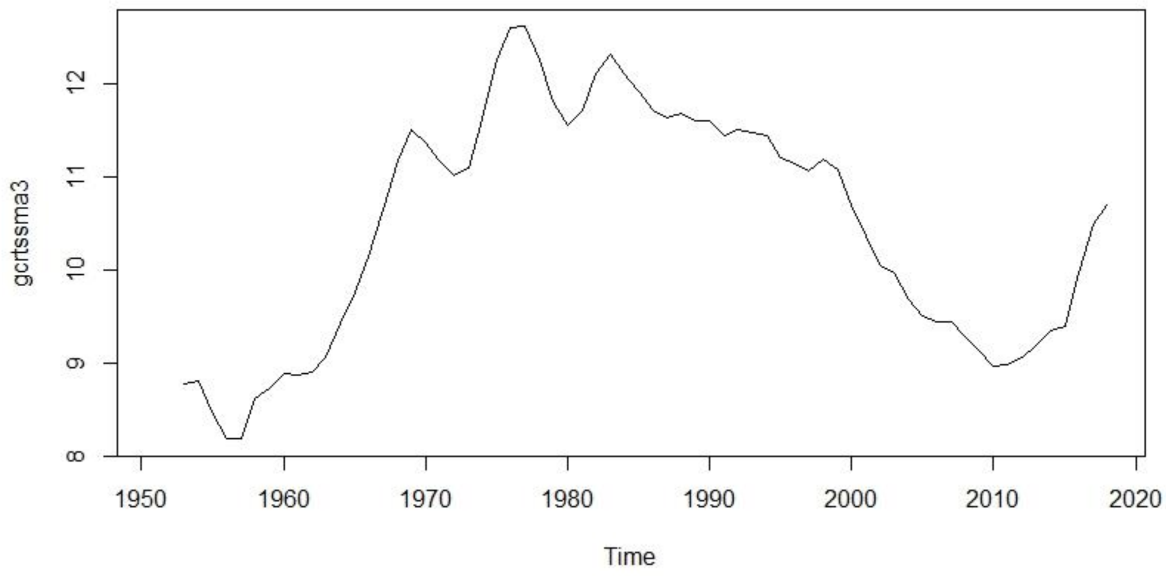
GDP to Currency Ratio Time-series data

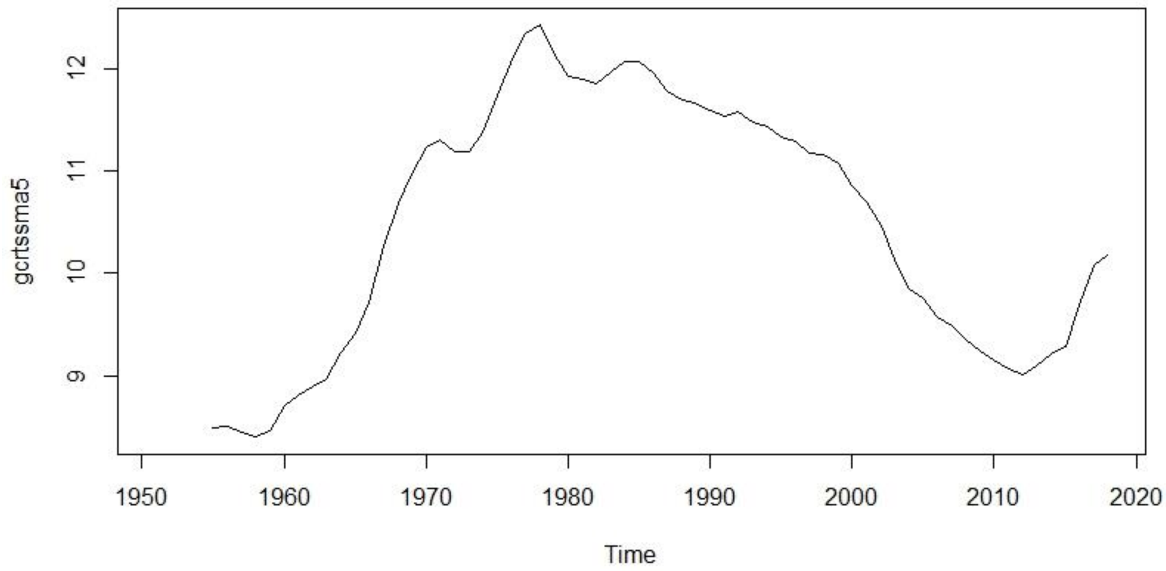
Took the yearly data from DBIE-RBI. Data was available from 1951-52 FY to 2018-19 FY. Below fig. is the plot after converting the data into time-series. As we can see, it is the non-seasonal data and could probably be described using an additive model, since the random fluctuations in the data are roughly constant in size over time and is non-stationary as the meansor the levels are not constant over time. As this is non-seasonal data, this only contains the trend component and the irregular component.



Decomposing the time-series data:

In order to decompose, SMA() has been used with n=3 initially and gradually increased to n=5 to get a smooth graph to predict the behaviour of the data. Below attached are the graphs with n=3 and n=5 respectively. And the irregular components are decreased.

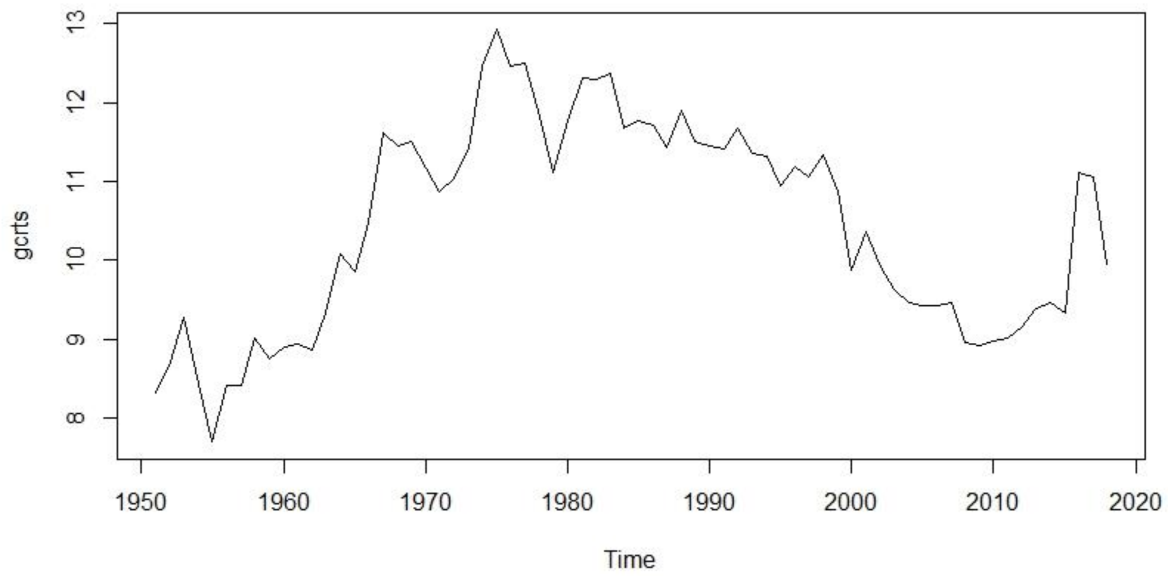




The data smoothened with a simple moving average of order 5 which gives a clearer picture of the trend component, and we can see that the GDP to Currency ratio seems to have increased from about 8 to about 13 during first ~30 years i.e., from 1951-52 FY to around 1979-80 FY and then decreased after that to about 9 near 2012-13 FY and then started increasing to around 10 until now i.e., 2018-19 FY. From the past 68 years it has increased and decreased.

Check for non-stationarity:

By plotting the time-series data, we got a graph with different levels or means through the time. By looking at the graph we can say that it is non-stationary data.



In order to get a clear picture, ADF test (unit root test), PP test, and KPSS test are done and following are the results:

```
> adf.test((gcrts),alternative="stationary",k=0)

        Augmented Dickey-Fuller Test

data:  (gcrts)
Dickey-Fuller = -1.8487, Lag order = 0, p-value = 0.6366
alternative hypothesis: stationary
```

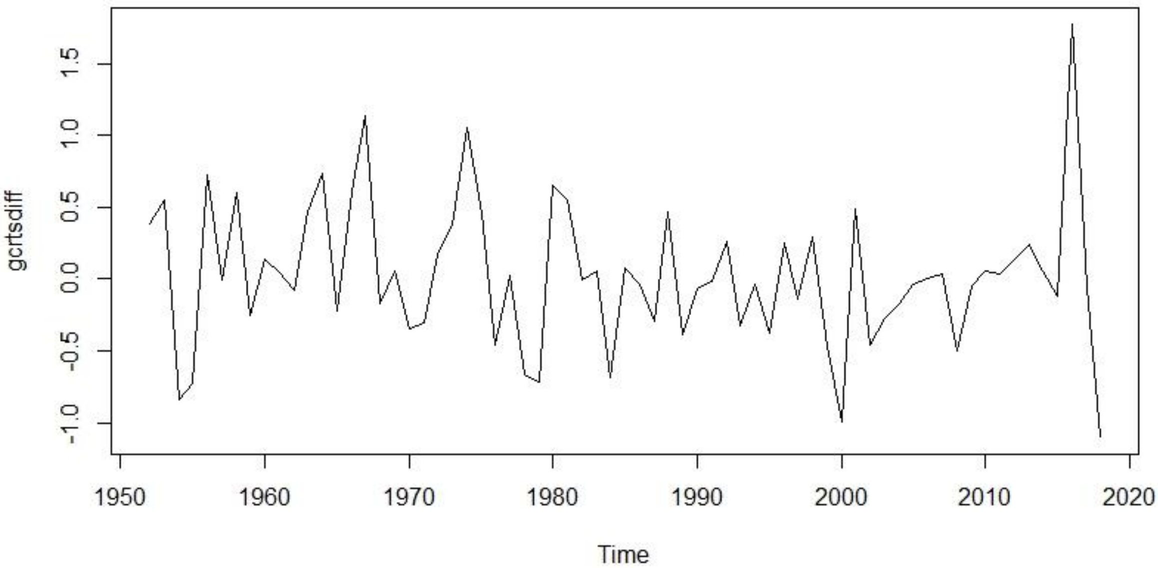
```
> pp.test(gcrts)
Phillips-Perron Unit Root Test
alternative: stationary

Type 1: no drift no trend
 lag   Z_rho p.value
   3 0.0707   0.702
-----
 Type 2: with drift no trend
 lag Z_rho p.value
   3 -5.68   0.415
-----
 Type 3: with drift and trend
 lag Z_rho p.value
   3 -5.19   0.786
---------------
Note: p-value = 0.01 means p.value <= 0.01
```

In both the results, as the p-values (0.6366 & 0.702) are larger, we don't reject the null hypothesis i.e., the given data is non-stationary.

```
> kpss.test(gcrts)

        KPSS Test for Level Stationarity

data:  gcrts
KPSS Level = 0.39002, Truncation lag parameter = 3, p-value = 0.08146
```

p-value in KPSS is less than 0.1, so we reject the null hypothesis which says that data is stationary and we can say that the data is non-stationary at 90% significance level.

In order to convert it to stationary data, diff() function is been used and got the difference time-series data which is stationary. Following are the graph and ADF test, PP test, and KPSS test results of the time-series data of <u>*first differences*</u>:




```
> adf.test((gcrtsdiff),alternative="stationary",k=0)

        Augmented Dickey-Fuller Test

data:  (gcrtsdiff)
Dickey-Fuller = -7.6796, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary
```

```
> pp.test(gcrtsdiff)
Phillips-Perron Unit Root Test
alternative: stationary

Type 1: no drift no trend
 lag Z_rho p.value
   3 -59.2    0.01
-----
 Type 2: with drift no trend
 lag Z_rho p.value
   3 -59.2    0.01
-----
 Type 3: with drift and trend
 lag Z_rho p.value
   3 -59.2    0.01
---------------
Note: p-value = 0.01 means p.value <= 0.01
```

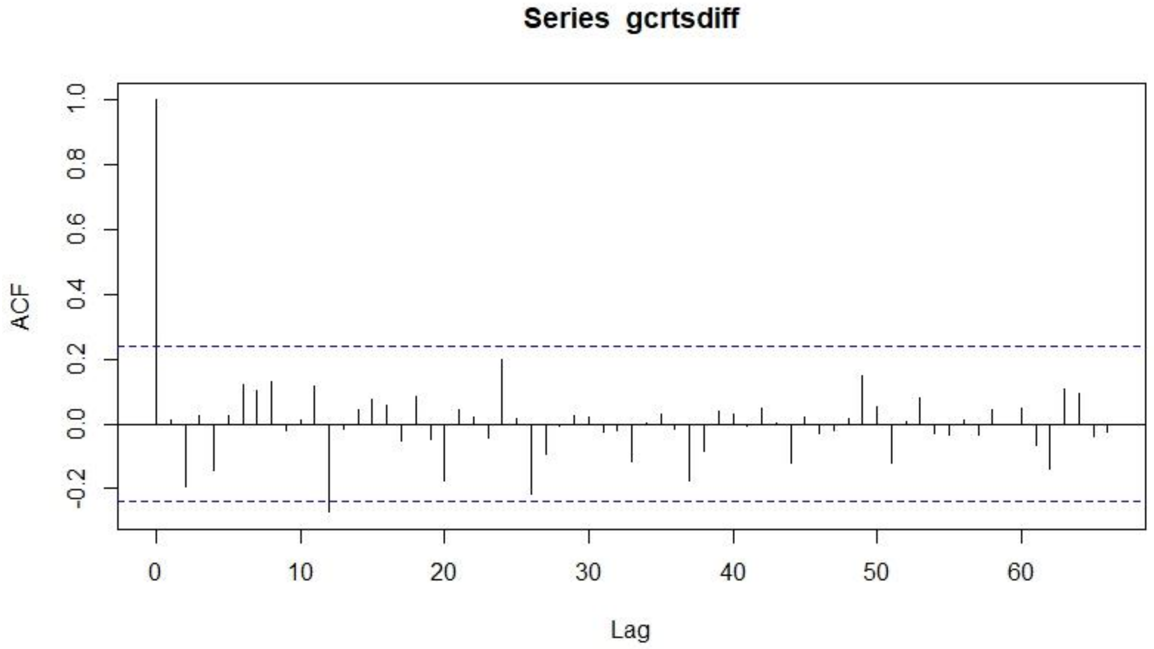
In both the results, as the p-values (0.01 & 0.01) of the difference time-series are changed and less, so we can reject the null hypothesis and conclude that the changed data became stationary.

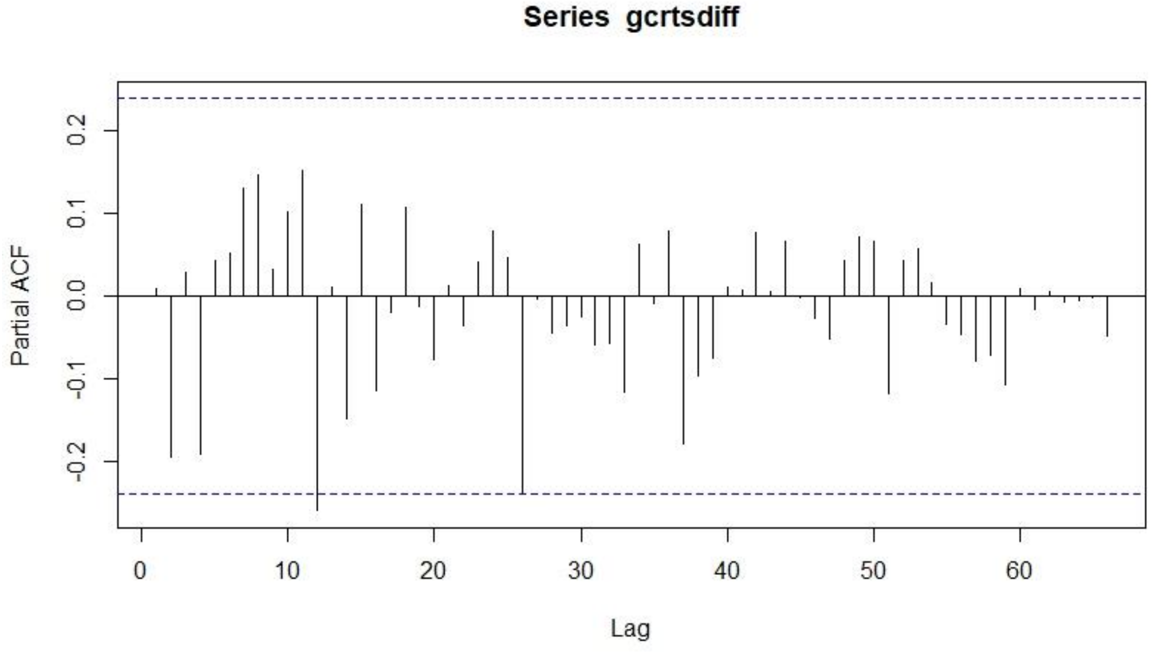
```
> kpss.test(gcrtsdiff)

        KPSS Test for Level Stationarity

data:  gcrtsdiff
KPSS Level = 0.25828, Truncation lag parameter = 3, p-value = 0.1

Warning message:
In kpss.test(gcrtsdiff) : p-value greater than printed p-value
```

Here, the p-value is greater than 0.1 which means we cannot reject the null hypothesis and conclude that the data is stationary.

Selection of ARIMA Model

Plotted correlogram and partial correlogram are as follows:





There are no significant values at the beginning and the significant values which came in the middle may be due to error terms and we cannot conclude anything from these plots.

Auto.arima() function gave the results as follows and we got a model with ARIMA(0,1,0), which means we are unable to fit a model for the data and this may due to the data is fully random white noise process. Which means it is error term or constant term.

```
> auto.arima(gcrts)
Series: gcrts
ARIMA(0,1,0)

sigma^2 estimated as 0.2484:  log likelihood=-48.42
AIC=98.83   AICc=98.89   BIC=101.04
```

```
> auto.arima(gcrtsdiff)
Series: gcrtsdiff
ARIMA(0,0,0) with zero mean

sigma^2 estimated as 0.2484:  log likelihood=-48.42
AIC=98.83   AICc=98.89   BIC=101.04
```

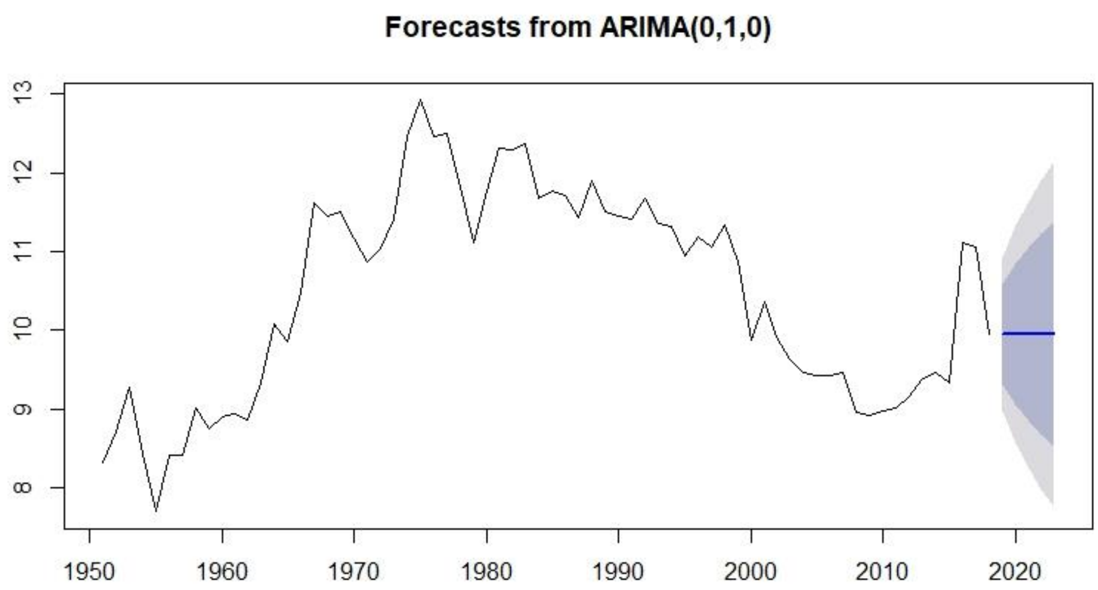
Predictive model for making forecasts

As we are unable to fit any model, we take ARIMA (0,1,0) and proceed to check whether we are getting a valid forecast or not. Our model has 0 parameters.

```
> gcrtsarima=arima(gcrts,order = c(0,1,0))
> gcrtsarima

Call:
arima(x = gcrts, order = c(0, 1, 0))


sigma^2 estimated as 0.2484:  log likelihood = -48.42,  aic = 98.83
```

We got no ma1 and standard error i.e., no model is fitted. But when forecast is run on the fitted model, we got the following result which gives us the GDP to Currency of next 5 financial years i.e., 2019-20 FY to 2023-2024 FY. We now need to test whether this is valid or not.

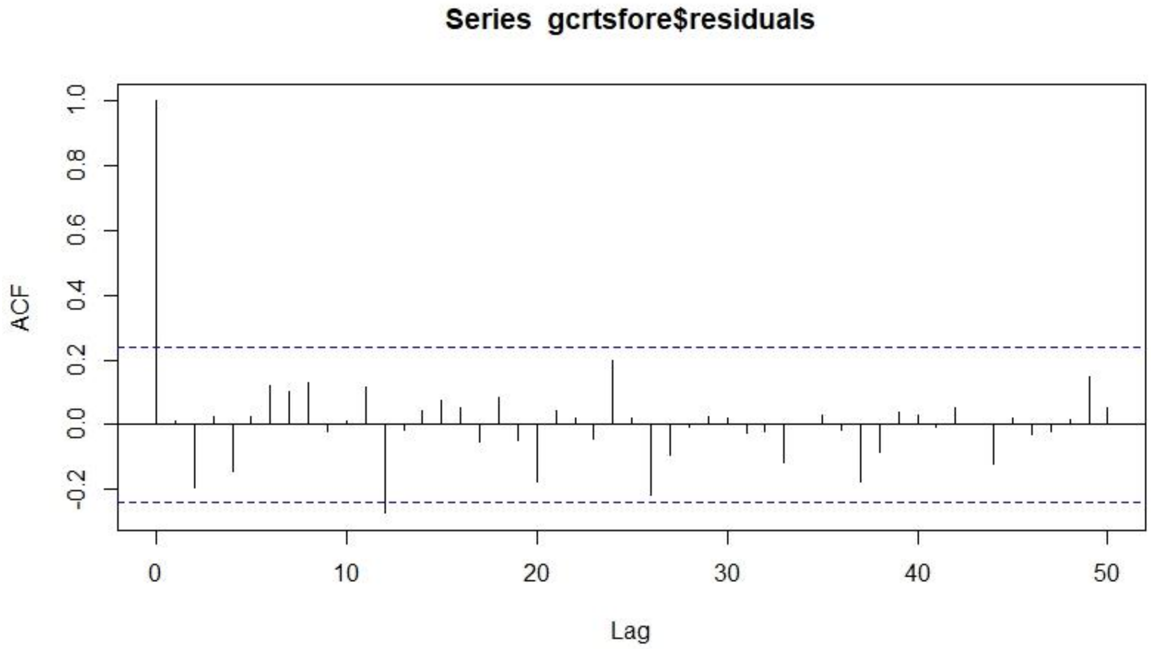
```
> gcrtsfore=forecast(gcrtsarima,h=5)
> gcrtsfore
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2019           9.95 9.311255 10.58875 8.973124 10.92688
2020           9.95 9.046678 10.85332 8.568488 11.33151
2021           9.95 8.843661 11.05634 8.258001 11.64200
2022           9.95 8.672510 11.22749 7.996248 11.90375
2023           9.95 8.521723 11.37828 7.765638 12.13436
```



Here, the dark grey region shows the forecasted range of GDP to Currency ratio from 2019-20 FY to 2023-24 FY at an 80% confidence interval and the lite grey shows us the range at 95% confidence interval.

Checking the predictive model

Following are the plot of the residuals and result of Ljung-Box test:

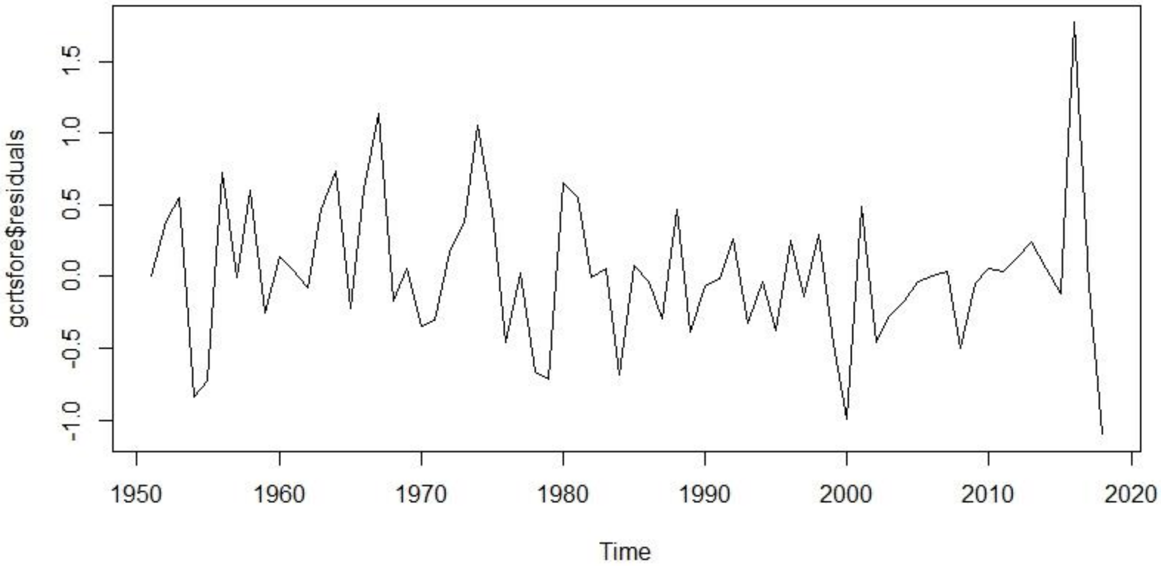


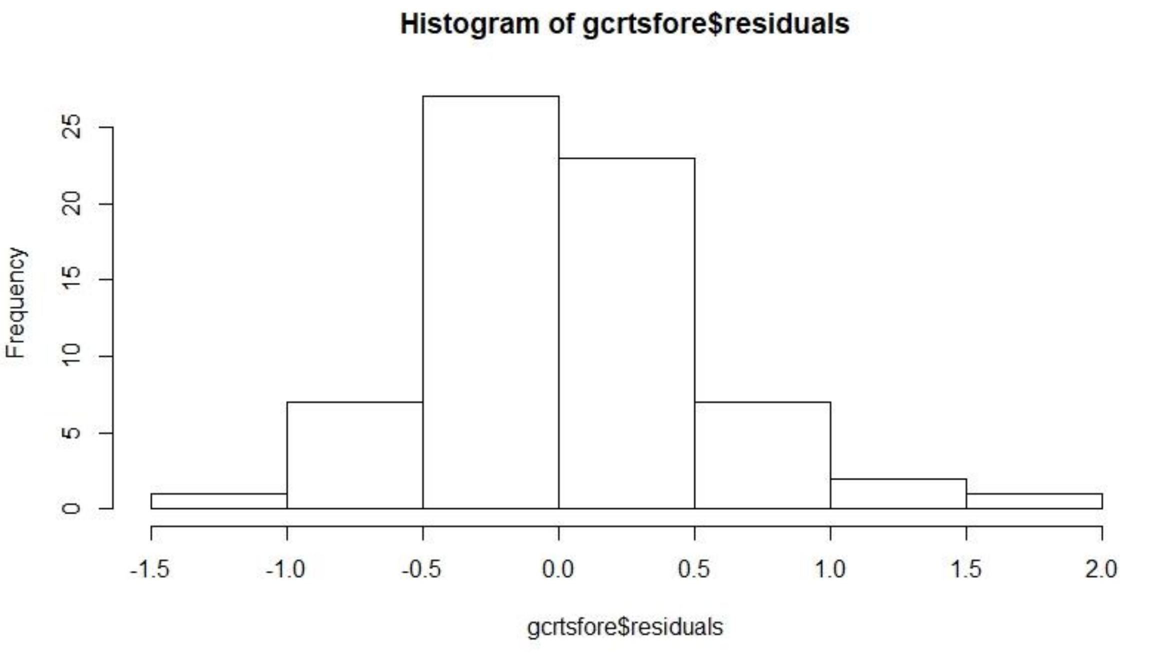

```
> Box.test(gcrtsfore$residuals,lag=50,type="Ljung-Box")

        Box-Ljung test

data:  gcrtsfore$residuals
X-squared = 50.371, df = 50, p-value = 0.4587
```

The correlogram shows only one significant value between 10 & 15 which we considered as an outlier or random value which came by chance; we ignore that. Other than that, none of the sample autocorrelations for lags 1-50 exceed the significance bounds, and the p-value for the Ljung-Box test is 0.4587, we can conclude that there is very little evidence for non-zero autocorrelations in the forecast errors at lags 1-50.

In order to get to the conclusions, we consider the residual plot(mean value too) and the histogram of forecast errors and are attached below.

```
> mean(gcrtsfore$residuals)
[1] 0.02394603
```





As we can see, the time plot of the in-sample forecast errors shows that the variance of the forecast errors seems to be roughly constant over time (there is slightly lower variance from 1985 to 2000 and then increased a little).

The histogram of the time series shows that the forecast errors are roughly normally distributed and the mean seems to be close to zero(0.024). Therefore, it is plausible that the forecast errors are normally distributed with mean zero and constant variance.

Since successive forecast errors do not seem to be correlated, and the forecast errors seem to be normally distributed with mean zero and constant variance, the ARIMA(0,1,0) (no model) does seem to provide an adequate predictive model for the GDP to Currency ratio.