Modeling and forecasting inflation in Sri Lanka using ARIMA models

Zhaoshuai Wang / Srinesh Heshan Fernando

2021/6/4

INTRODUCTION

This study is based on a data set of annual rates of inflation in Sri Lanka (LK) ranging over the period 1960 – 2019. All the data was adapted from the World Bank data sources. We will use ARIMA model to forecast the future rates of inflation.

Load the data

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(forecast)
```

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

```
library(ggplot2)
library(tseries)
setwd("C:/Users/wangz/Desktop")
md <- read.csv("data.csv",header = TRUE)
SL<- md[md$Country.Name == 'Sri Lanka',]
CN<- md[md$Country.Name == 'China',]
SL <- SL[ , !names(SL) %in% c("Country.Name", "Country.Code","Indicator.Name","Indicator.Code")]
SL <-t(SL)
rownames(SL) <- c(1960:2020)
colnames(SL) <- c("inflation rates")
```

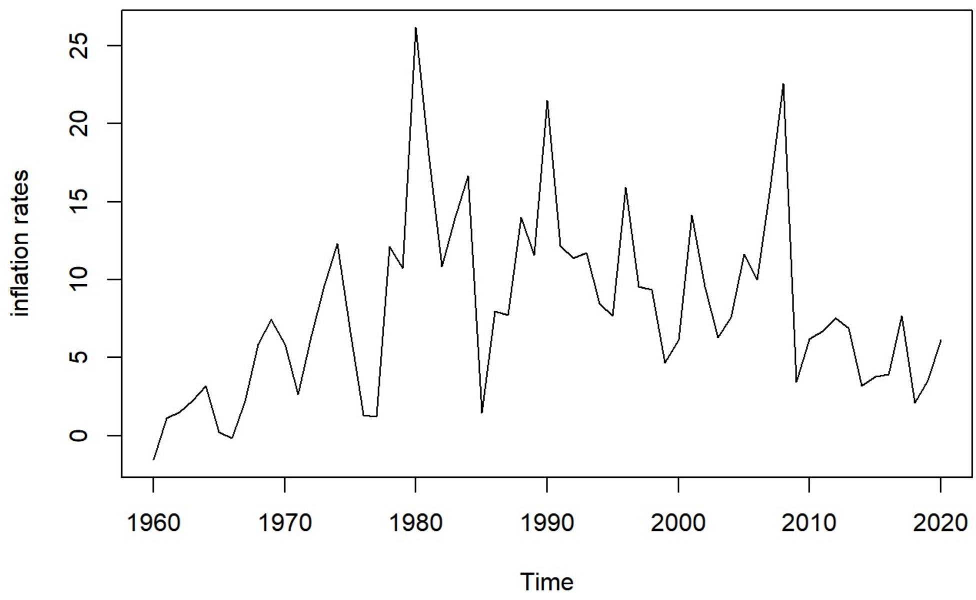
Converting data to time series format (TS)

```
Sri_Lanka <- ts(SL,start=1960,frequency=1)
str(Sri_Lanka)
```

```
##  Time-Series [1:61, 1] from 1960 to 2020: -1.54 1.13 1.5 2.27 3.2 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "inflation rates"
```

View trend chart

```
plot.ts(Sri_Lanka)
```



Check if Stationary

The augmented Dickey Fuller (ADF) test can be used to test whether a time series is stationary or not. The Ho is that the sequence is nonstationary.

ADF test:

```
adf.test(Sri_Lanka,alternative="stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Sri_Lanka
## Dickey-Fuller = -2.6328, Lag order = 3, p-value = 0.3189
## alternative hypothesis: stationary
```

As the p-value is higher than 0.05, so we can not reject the Ho, which means the sequence is nonstationary.

So we have to change it into a stationary sequence by differential processing. Here, the ndiffs() function is used to judge the difference degree:

Judge the difference degree

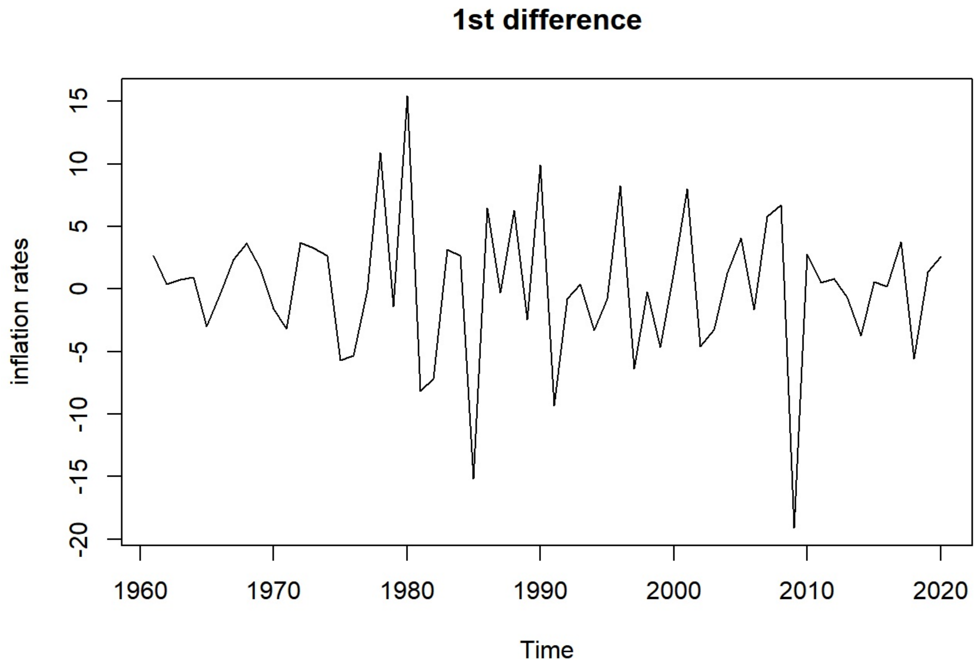
```
ndiffs(Sri_Lanka)
```

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

The result of the function is the first-order difference, let's do 1st difference of the data and check ACF/PACF:

1st difference/ ACF and PACF

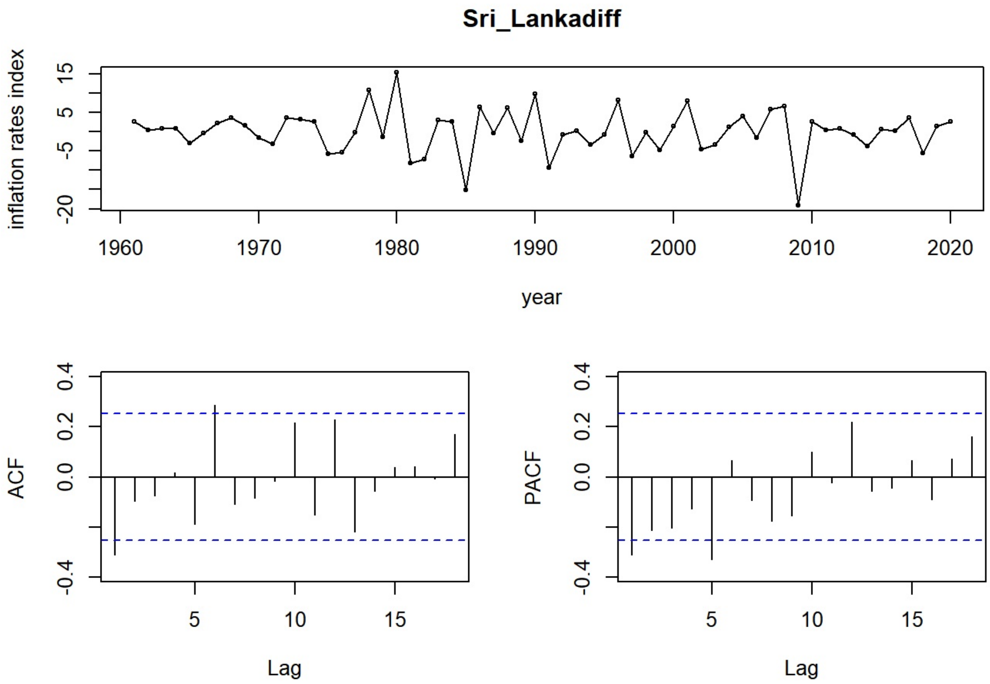
```
Sri_Lankadiff<-diff(Sri_Lanka)
plot.ts(Sri_Lankadiff,main="1st difference")
```



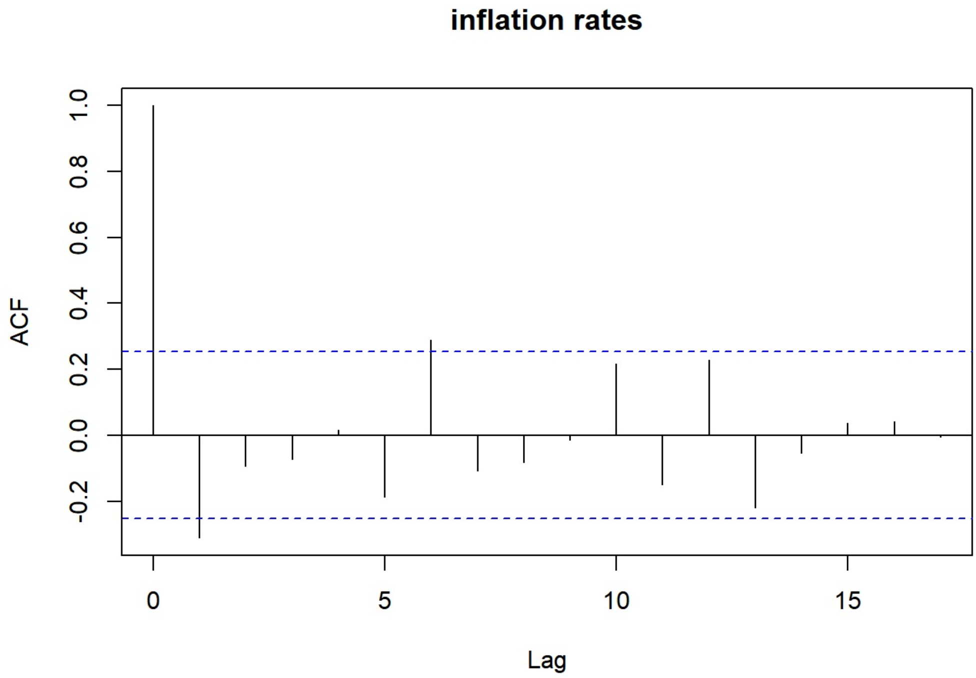
As we could see the sequence become stationary.

Then use 2 different way to check ACF and PACF:

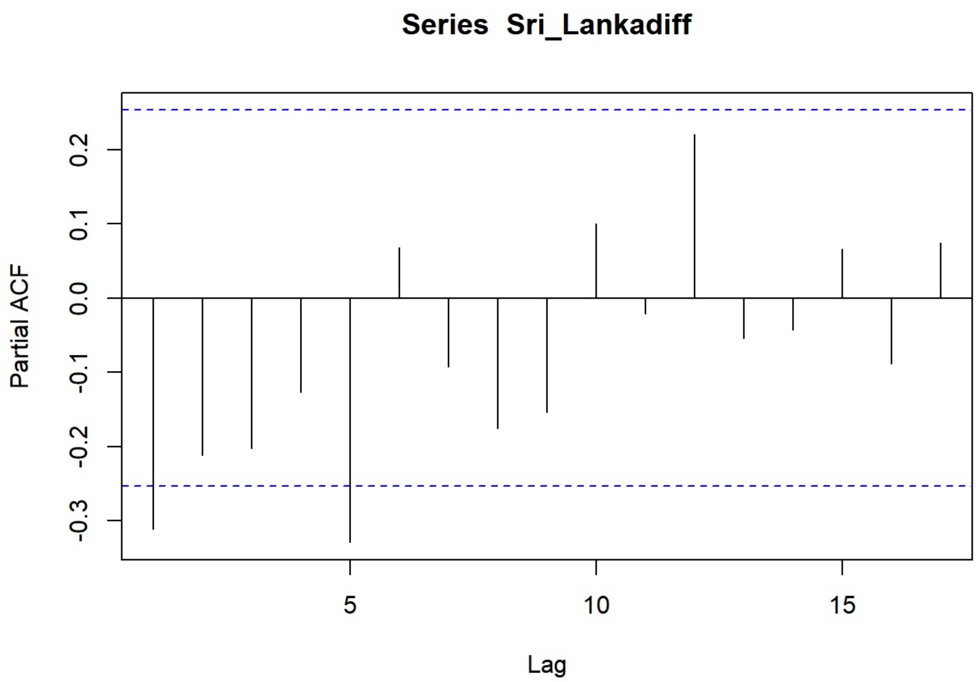
```
tsdisplay(Sri_Lankadiff,xlab="year",ylab="inflation rates index")
```




```
acf(Sri_Lankadiff)
```




```
pacf(Sri_Lankadiff)
```



As we could see the result of ACF and PACF, between the two blue dashed lines can be regarded as 0. There is no outlier for both ACF and PACF. So we could select ARIMA(1,1,1), the difference order is 1, the autoregressive order is 1, and the smooth moving order is 1.

Model fitting Then we do Model fitting:

```
pre<-arima(Sri_Lanka,order=c(1,1,1))
pre
```

```
##
## Call:
## arima(x = Sri_Lanka, order = c(1, 1, 1))
##
## Coefficients:
##          ar1      ma1
##       0.2977  -0.8213
## s.e.  0.1639   0.0941
##
## sigma^2 estimated as 24.5:  log likelihood = -181.42,  aic = 368.84
```

```
auto.arima(Sri_Lanka,seasonal=F)
```

```
## Series: Sri_Lanka
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.2977  -0.8213
## s.e.  0.1639   0.0941
##
## sigma^2 estimated as 25.34:  log likelihood=-181.42
## AIC=368.84   AICc=369.27   BIC=375.12
```

From first result, we could get AIC=368.84, and from second on, we get AIC=368.84 AICc=369.27 BIC=375.12. (AIC value is the standard to judge whether the model fits well or not, the smaller the better)

Also ,we could use Auto.ARIMA () in forecast can directly select the best fitting model, it is same with our choice.

```
arima1<-auto.arima(Sri_Lanka,trace=T)
```

```
##
##  ARIMA(2,1,2) with drift         : 374.8459
##  ARIMA(0,1,0) with drift         : 382.4603
##  ARIMA(1,1,0) with drift         : 378.6245
##  ARIMA(0,1,1) with drift         : 372.3281
##  ARIMA(0,1,0)                    : 380.3496
##  ARIMA(1,1,1) with drift         : 371.3853
##  ARIMA(2,1,1) with drift         : 373.6836
##  ARIMA(1,1,2) with drift         : 373.7145
##  ARIMA(0,1,2) with drift         : 371.5328
##  ARIMA(2,1,0) with drift         : 378.1811
##  ARIMA(1,1,1)                    : 369.2703
##  ARIMA(0,1,1)                    : 370.223
##  ARIMA(1,1,0)                    : 376.4478
##  ARIMA(2,1,1)                    : 371.4792
##  ARIMA(1,1,2)                    : 371.5118
##  ARIMA(0,1,2)                    : 369.4031
##  ARIMA(2,1,0)                    : 375.9293
##  ARIMA(2,1,2)                    : 372.5652
##
##  Best model: ARIMA(1,1,1)
```

Forecast according to the selected model

Forecast the value in the next 6 years and set up 95% confidence interval:

```
inflation<-forecast(pre,h=6,level=c(99.5))
inflation
```

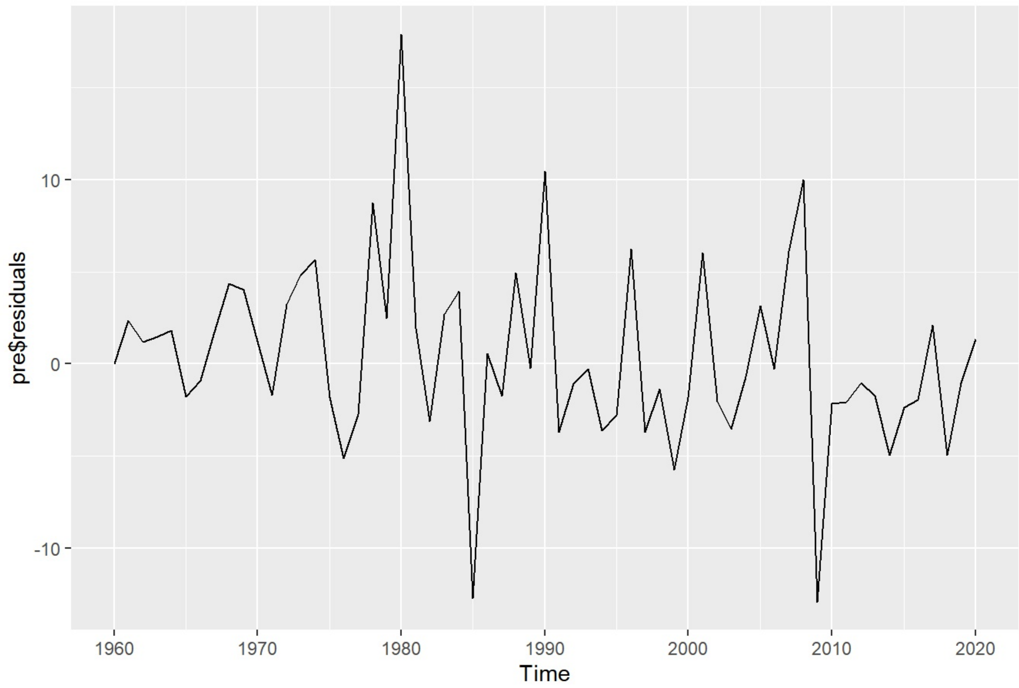
```
##      Point Forecast    Lo 99.5  Hi 99.5
## 2021       5.810083  -8.083320 19.70349
## 2022       5.707731  -9.681229 21.09669
## 2023       5.677265 -10.342735 21.69727
## 2024       5.668197 -10.798029 22.13442
## 2025       5.665498 -11.192988 22.52398
## 2026       5.664694 -11.565243 22.89463
```

Model evaluation

1. Judge whether the prediction error is a normal distribution with zero mean and constant variance

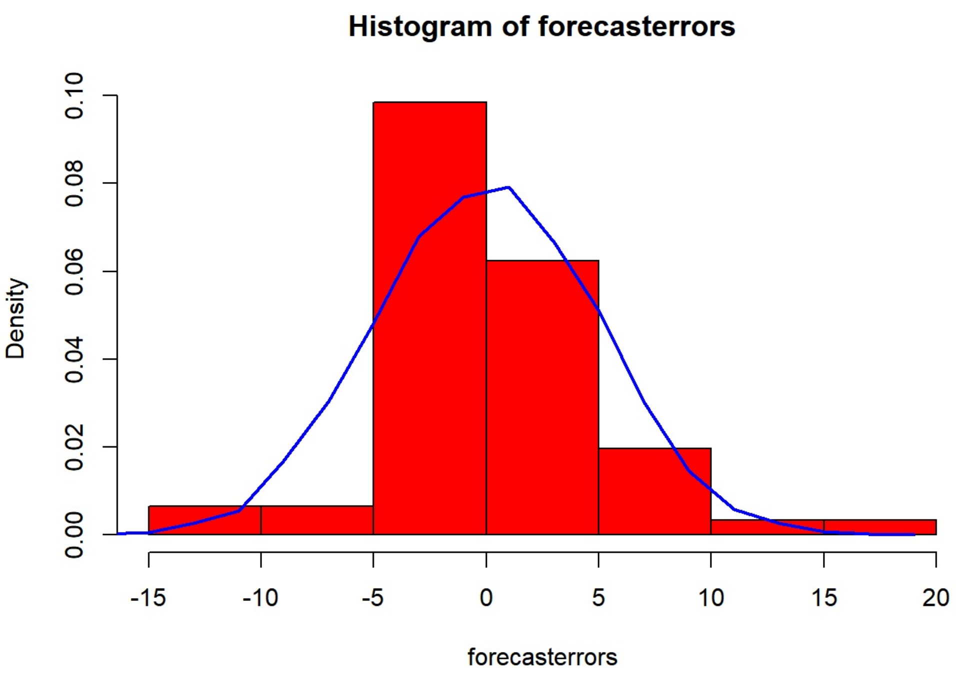
Plot it first:

```
autoplot(pre$residuals)
```



Then we make a function to Convert to normal distribution:

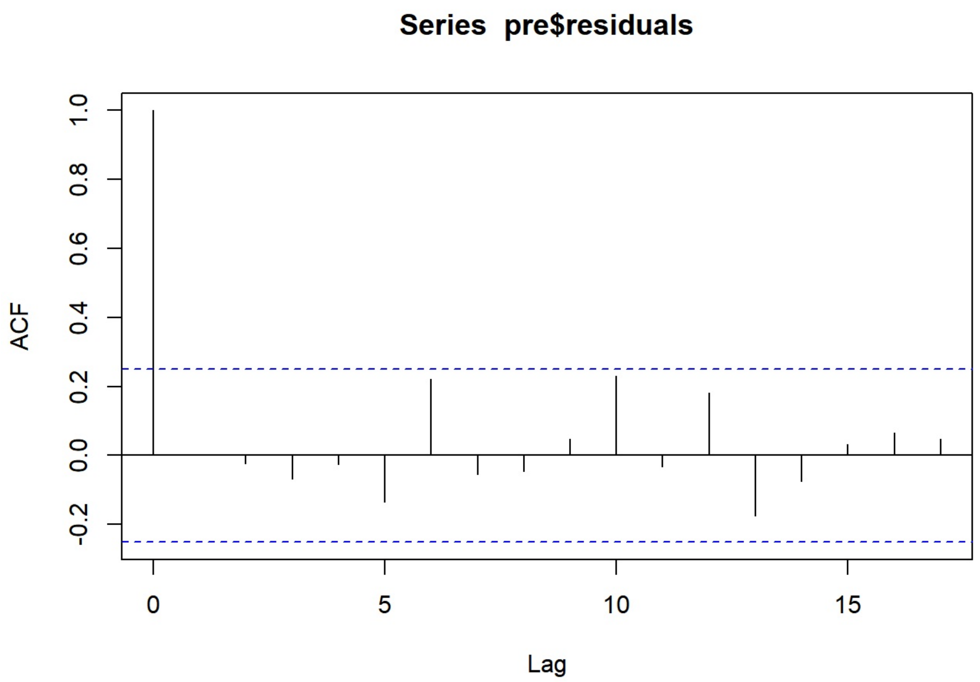
```
plotForecastErrors <- function(forecasterrors)
{
    # make a red histogram of the forecast errors:
    mysd <- sd(forecasterrors)
    hist(forecasterrors, col="red", freq=FALSE)
    # freq=FALSE ensures the area under the histogram = 1
    # generate normally distributed data with mean 0 and standard deviation mysd
    mynorm <- rnorm(10000, mean=0, sd=mysd)
    myhist <- hist(mynorm, plot=FALSE)
    # plot the normal curve as a blue line on top of the histogram of forecast errors:
    points(myhist$mids, myhist$density, type="l", col="blue", lwd=2)
}
plotForecastErrors(pre$residuals)
```



Get the above normal distribution map, we could say, the normality test: pass.

2. Then we carry out the independence test of residual

```
acf(pre$residuals)
```



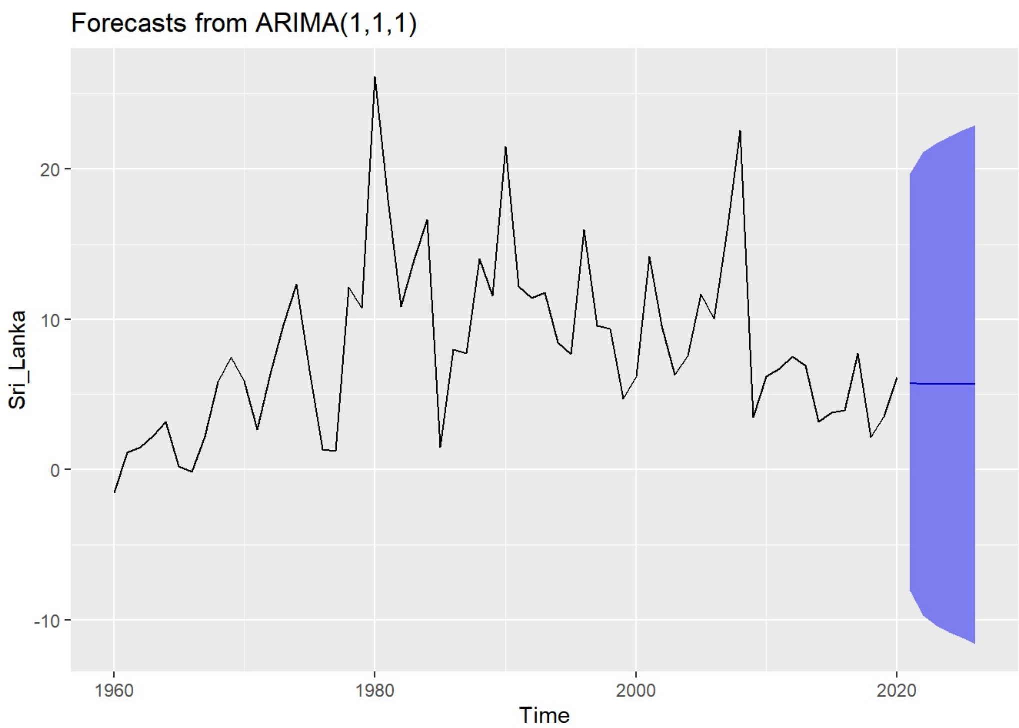

```
Box.test(pre$residuals, lag=1, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  pre$residuals
## X-squared = 8.595e-06, df = 1, p-value = 0.9977
```

From the result, we could see p-value = 0.9977>0.05, so we have to reject the hypothesis Ho, which means that the residual is considered to be independent.

Both tests are passed, so we could say our model seems good.

```
autoplot(inflation)
```



Conclusion

In this paper, we did the time series analysis and prediction.From ACF and PACF,we could get the final ARIMA(1,1,1) model, and our prediction basically conforms to the trend law of data.