Take Home Test

Ashwin Kumar Ashok Kumar

5 February 2016

You are in charge of forecasting industrial output produced in the state of Armorica at the Federal Reserve Bank. Monthly industrial output data for Armorica has been provided to you in the spreadsheet.

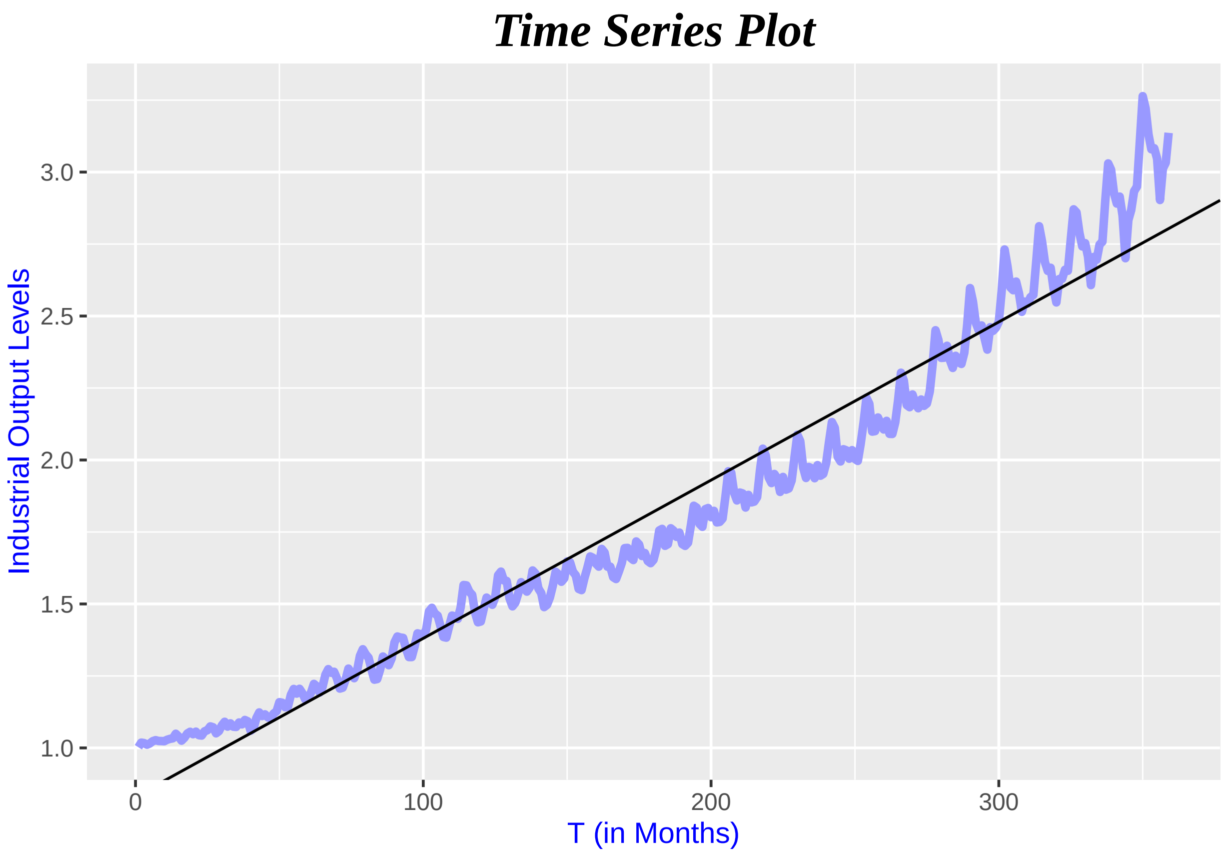
```
setwd("~/Acads/MFE 2016/2nd Quarter/Empirical Methods in Finance/Week 5")
library(lubridate)
library(xlsx)
```

```
## Loading required package: rJava

## Loading required package: xlsxjars
```

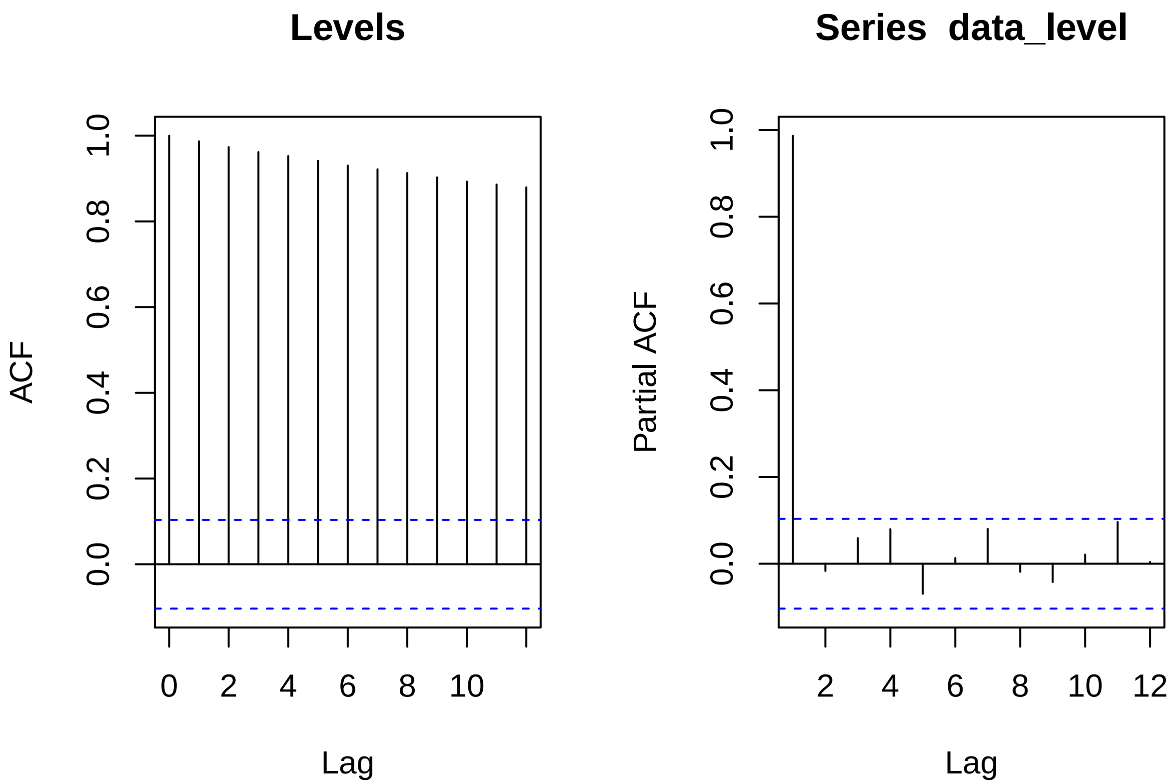
```
library(xts)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```
library(ggplot2)
data_level <- read.xlsx('open_book_data.xlsx','Sheet1')
t <- 1:length(data_level[,1])
colnames(data_level)<- "Levels"
reg <- lm(data_level[,1]~t)
c1 <- reg$coefficients[1]
m1 <- reg$coefficients[2]
logdata <- log(data_level)
reg2 <- lm(logdata[,1]~t)
c2 <- reg2$coefficients[1]
m2 <- reg2$coefficients[2]
y_diff <- diff(logdata[,1], lag=1)
diffdata <- data.frame(y_diff)
colnames(diffdata) <- "DiffLog"
t2 <- 1:length(diffdata[,1])
p1 <- ggplot(data_level,aes(x=t, y=data_level[,1])) + geom_line(color= '#9999FF', size=1.5) + ggtitle("
xlab("T (in Months)") + ylab("Industrial Output Levels") +
theme(plot.title =element_text(family="Times", face="bold.italic", size=18) ) + theme(axis.title = eleme
p1 + geom_abline(intercept = c1 , slope = m1)
```



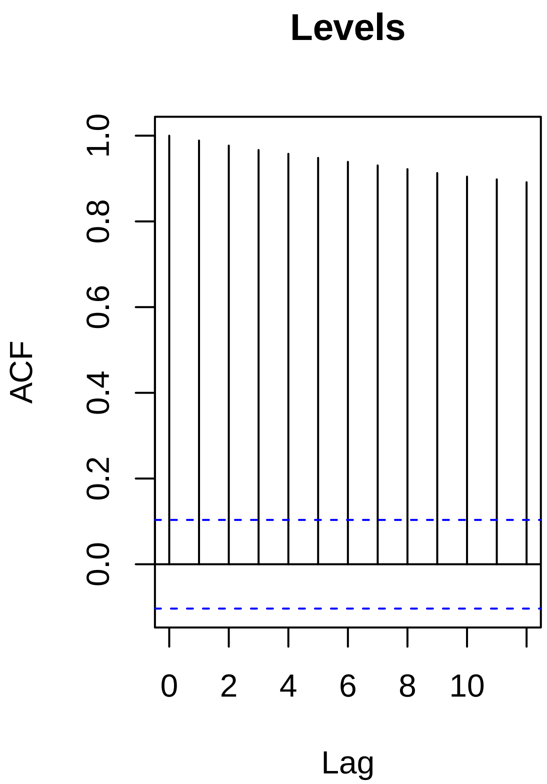

```
par(mfrow =c(1,2))
acf(data_level,lag.max = 12)
pacf(data_level,lag.max = 12)
```

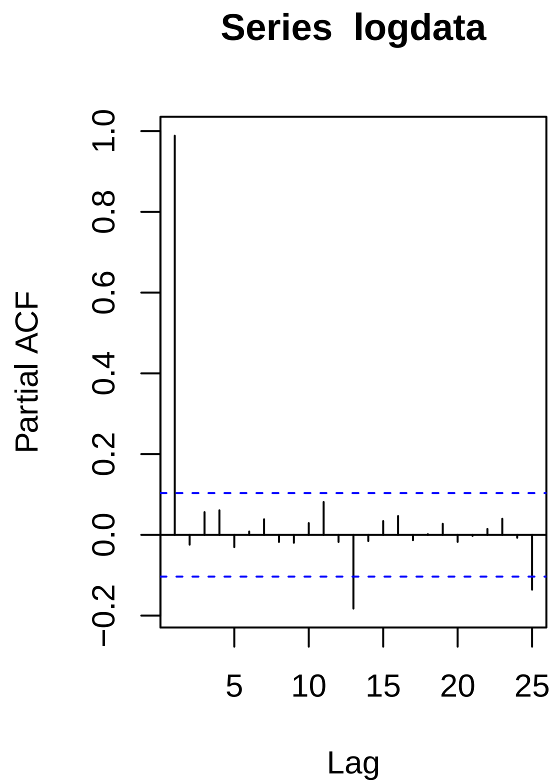


The time series plot of industrial data has R^2 = 0.95 a trend in data. The acf is signficant and high. We can notice that the acf is decreasing very slowly (the series is not weak stationary). This can be shown from the graph of the time series which when fitted with straight line has high R^2. Also, we do not have any conclusion from pacf. Thus levels data cannot be used to model the data. -> Log Levels

```
p2 <- ggplot(logdata,aes(x=t, y=logdata[,1])) + geom_line(color= '#9999FF', size=1.5) + ggtitle("Log Ti
    xlab("T (in Months)") + ylab("Industrial Output Levels") +
    theme(plot.title =element_text(family="Times", face="bold.italic", size=18) ) + theme(axis.title = e
p2+ geom_abline(intercept = c2 , slope = m2)
```

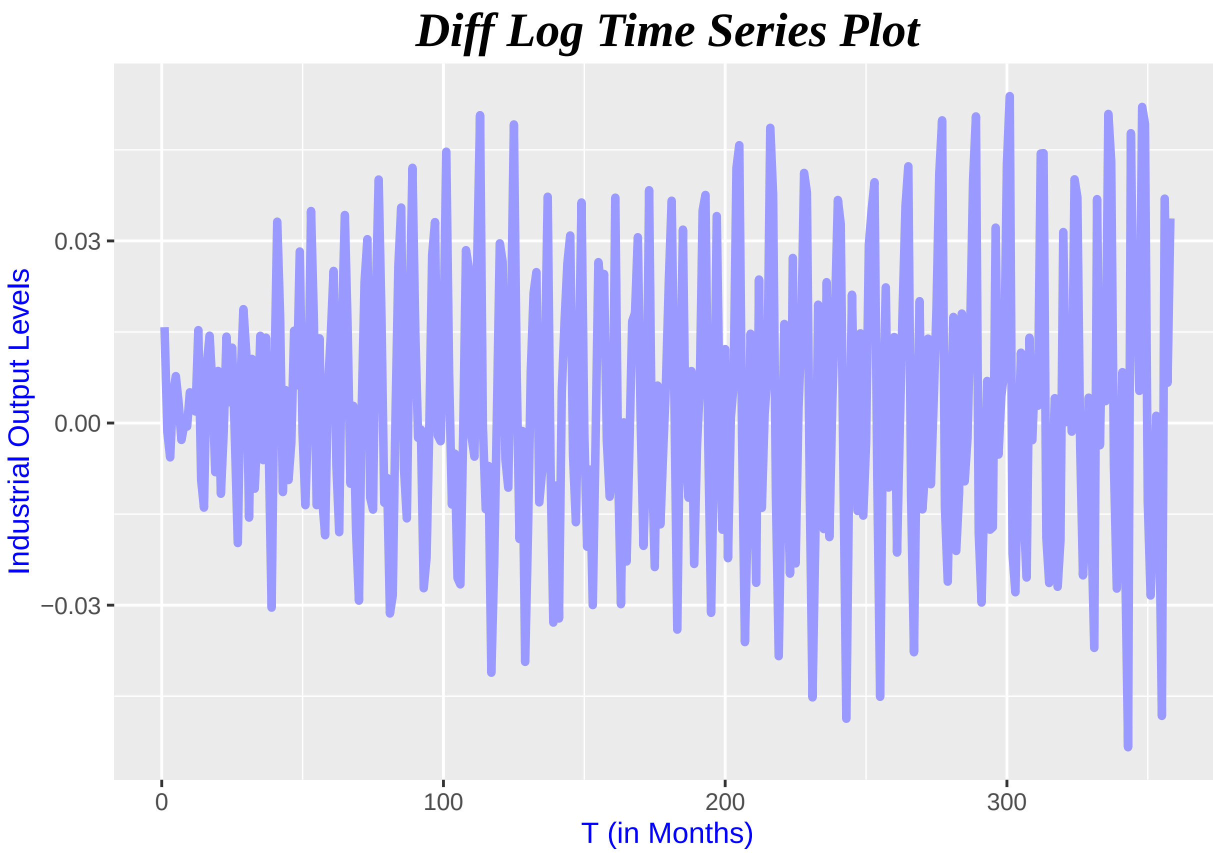

```
par(mfrow = c(1,2))
acf(logdata,lag.max = 12)
pacf(logdata)
```



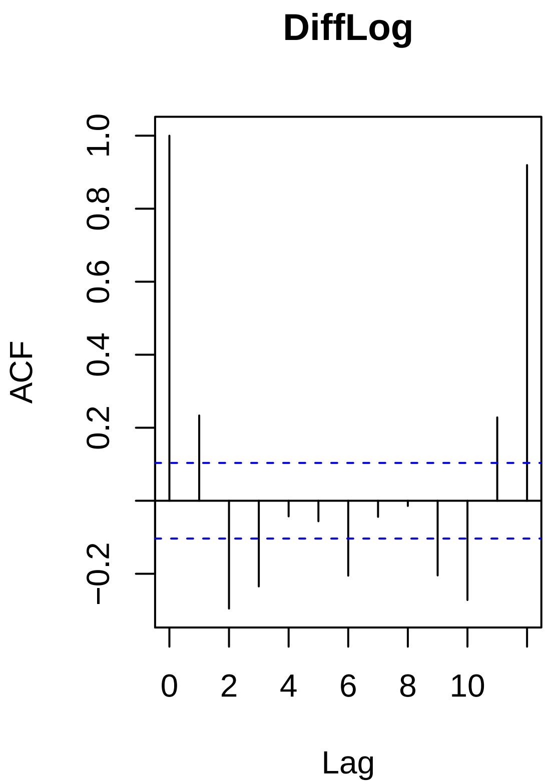


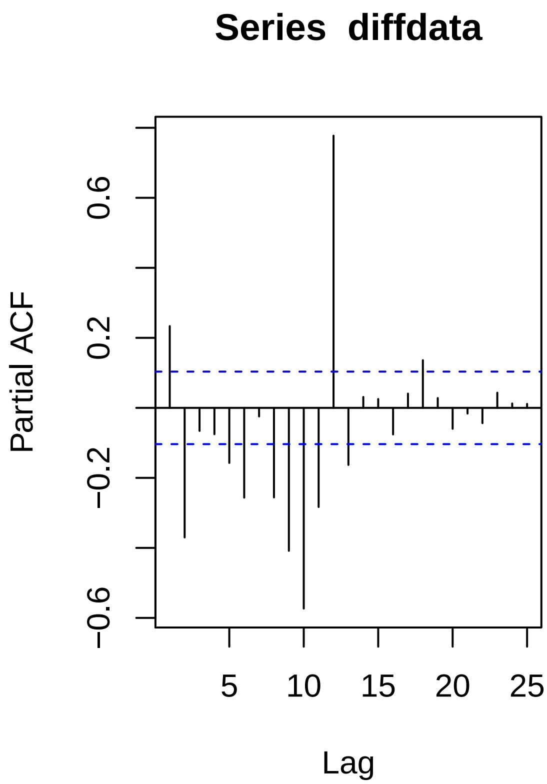
The above series fits the data better (R^2 = 0.98) than the Level. But still we find the trend (similar acf). The series is not weak stationary. -> Diff Log

```
ggplot(diffdata,aes(x=t2, y=diffdata[,1])) + geom_line(color= '#9999FF', size=1.5) + ggtitle("Diff Log '
    xlab("T (in Months)") + ylab("Industrial Output Levels") +
    theme(plot.title =element_text(family="Times", face="bold.italic", size=18) ) + theme(axis.title = e
```



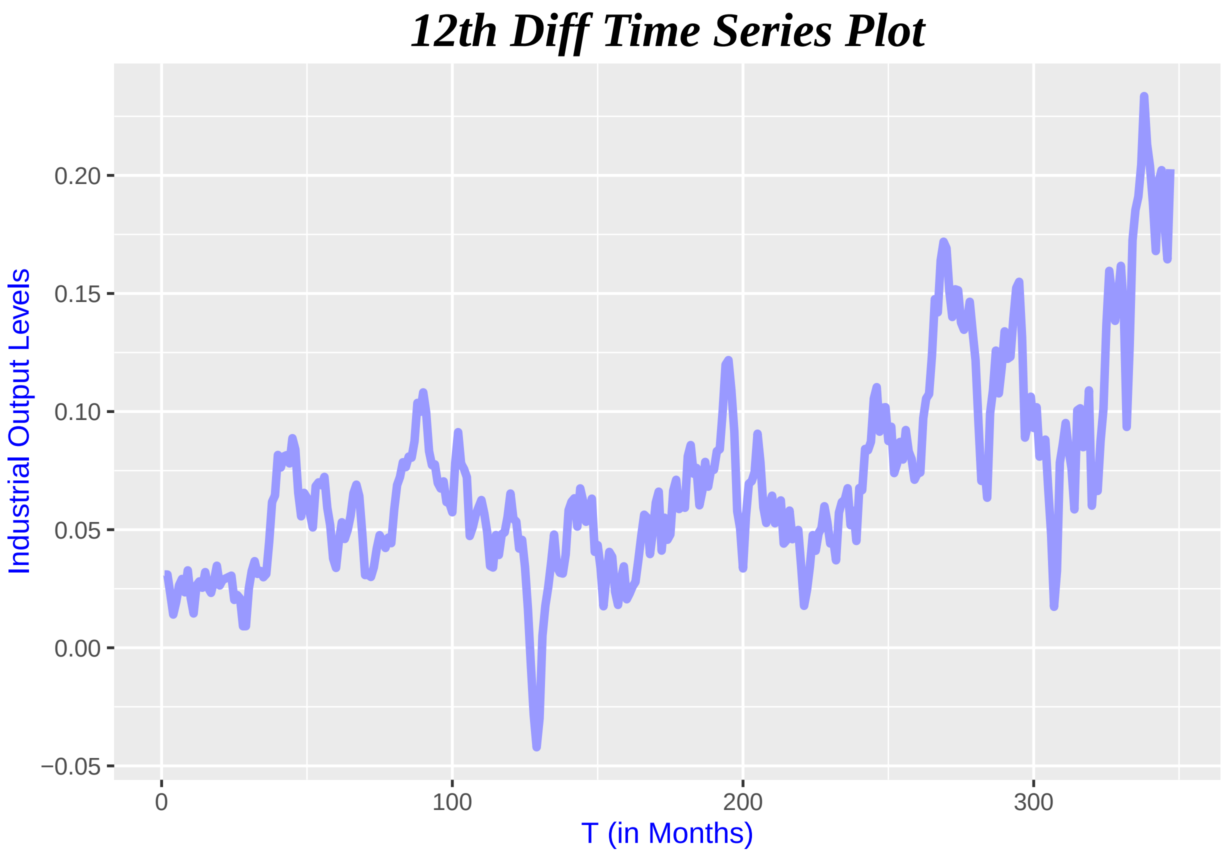

```
par(mfrow = c(1,2))
acf(diffdata, lag.max = 12)
pacf(diffdata)
```



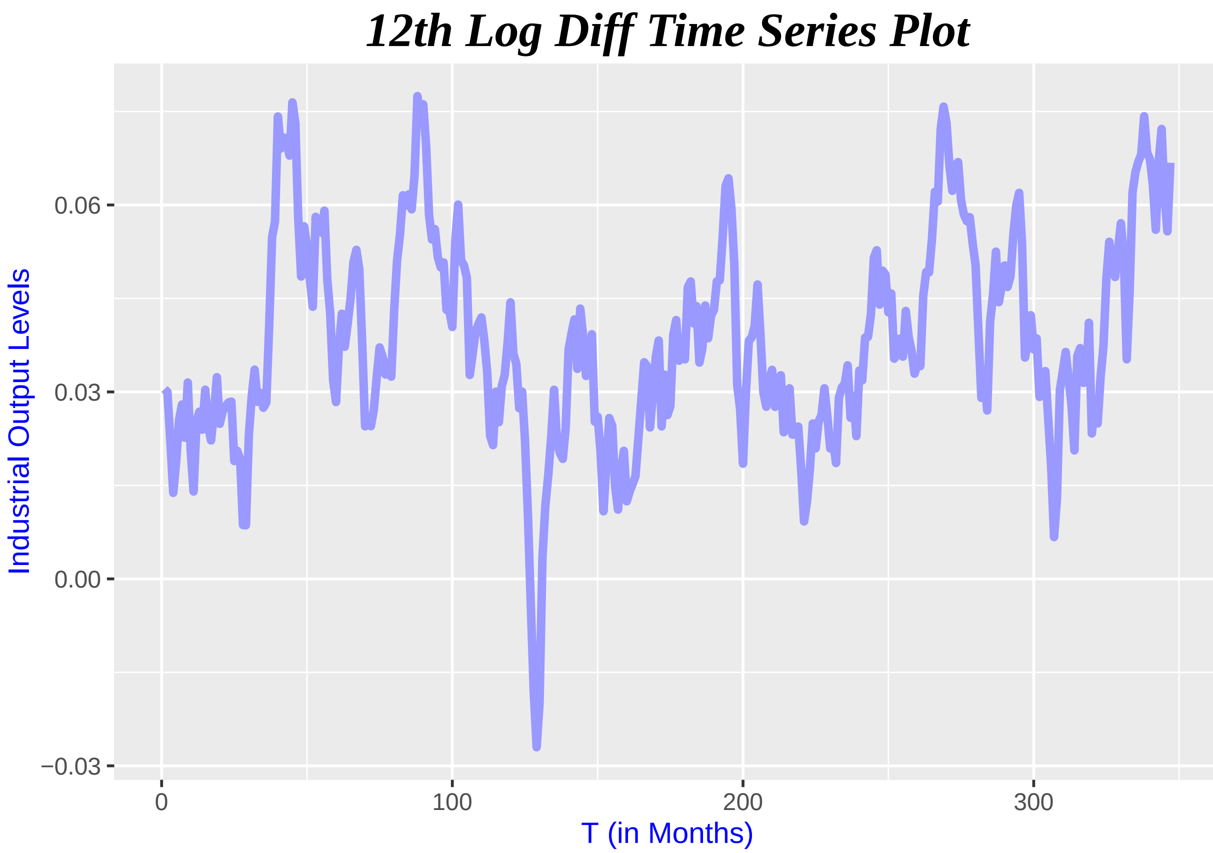


The diff log plot seems to have scattered. We can see that the trend is removed. But the series is is signifantly autocorrelated. -> Remove trend by taking 12th lag (removing seasonality).

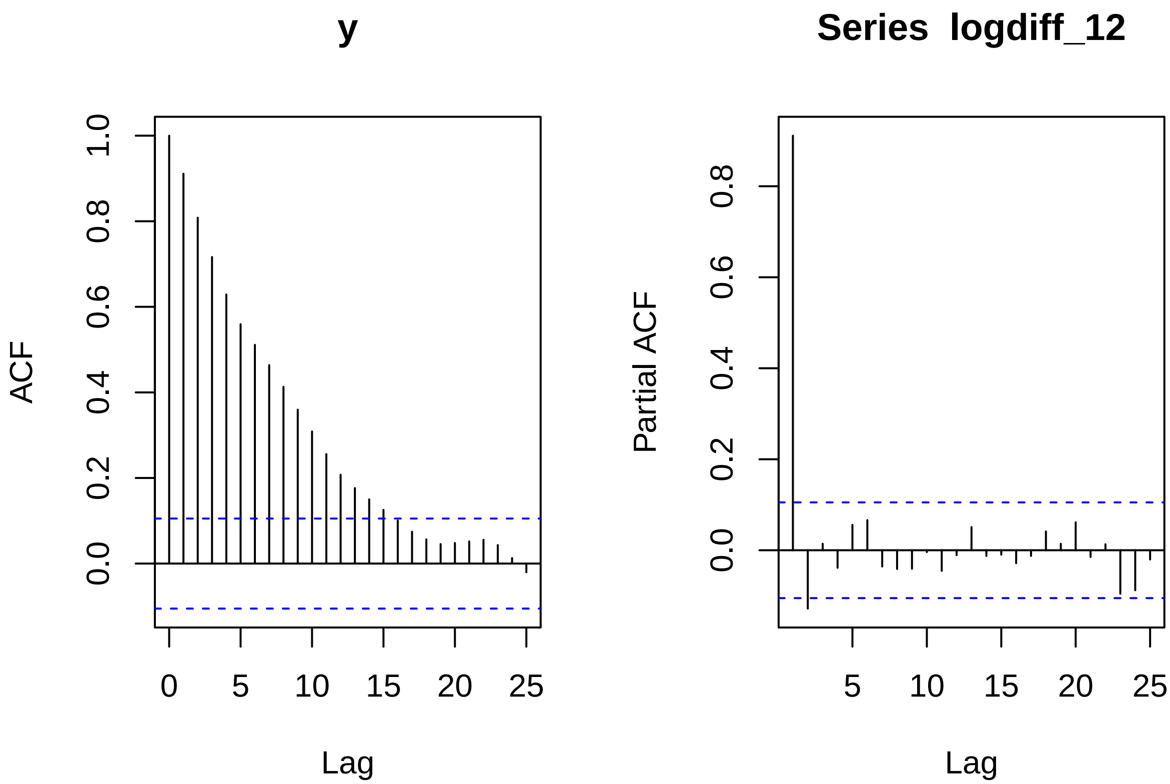
```
y1 <- diff(data_level[,1],lag=12)
t3 <- 1:length(y1)
diff_12 <- data.frame(y1)

ggplot(diff_12,aes(x=t3, y=diff_12[,1])) + geom_line(color= '#9999FF', size=1.5) + ggtitle("12th Diff T
    xlab("T (in Months)") + ylab("Industrial Output Levels") +
    theme(plot.title =element_text(family="Times", face="bold.italic", size=18) ) + theme(axis.title = e
```




```
y<- diff(logdata[,1],lag=12)
logdiff_12 <- data.frame(y)
ggplot(logdiff_12,aes(x=t3, y=logdiff_12[,1])) + geom_line(color= '#9999FF', size=1.5) + ggtitle("12th I
    xlab("T (in Months)") + ylab("Industrial Output Levels") +
    theme(plot.title =element_text(family="Times", face="bold.italic", size=18) ) + theme(axis.title = e
```

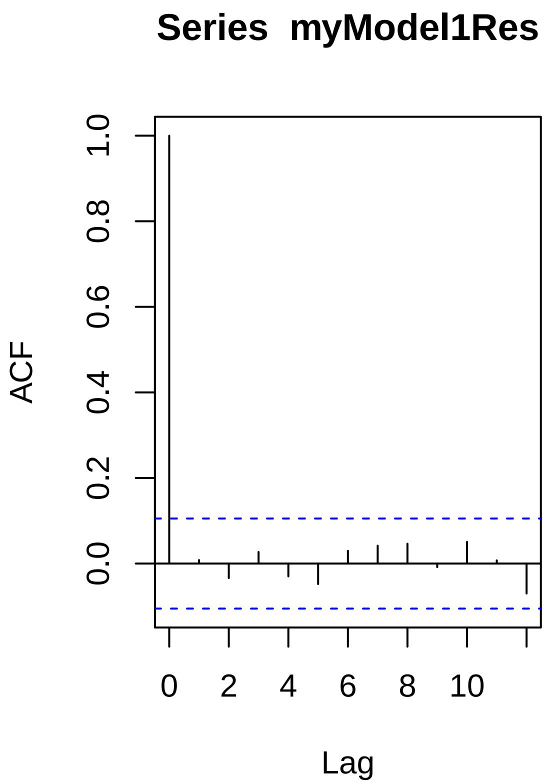


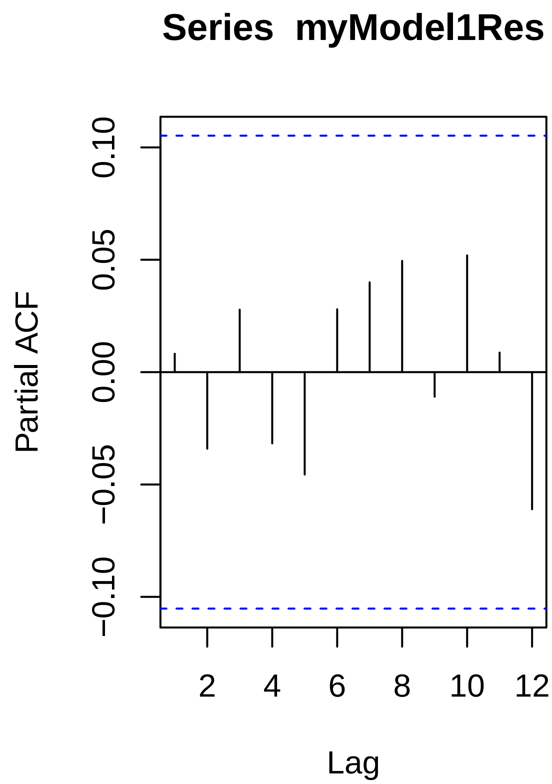

```r
par(mfrow = c(1,2))
acf(logdiff_12)
pacf(logdiff_12)
```



Note that the 12th lag is taken for log data. Otherwise the series (for lag 12 at levels) is not stationary. The resulting time series has a signifantly dropping acf. Also the pacf shows that there is no acf after 2nd lag(pacf =0 after second). Fitting AR(2) model

```
myModel1<-arima(logdiff_12, order=c(2,0,0))
myModel1Coef<-coef(myModel1)
myModel1Res<-residuals(myModel1)
par(mfrow = c(1,2))
acf(myModel1Res,lag.max = 12)
pacf(myModel1Res,lag.max = 12)
```

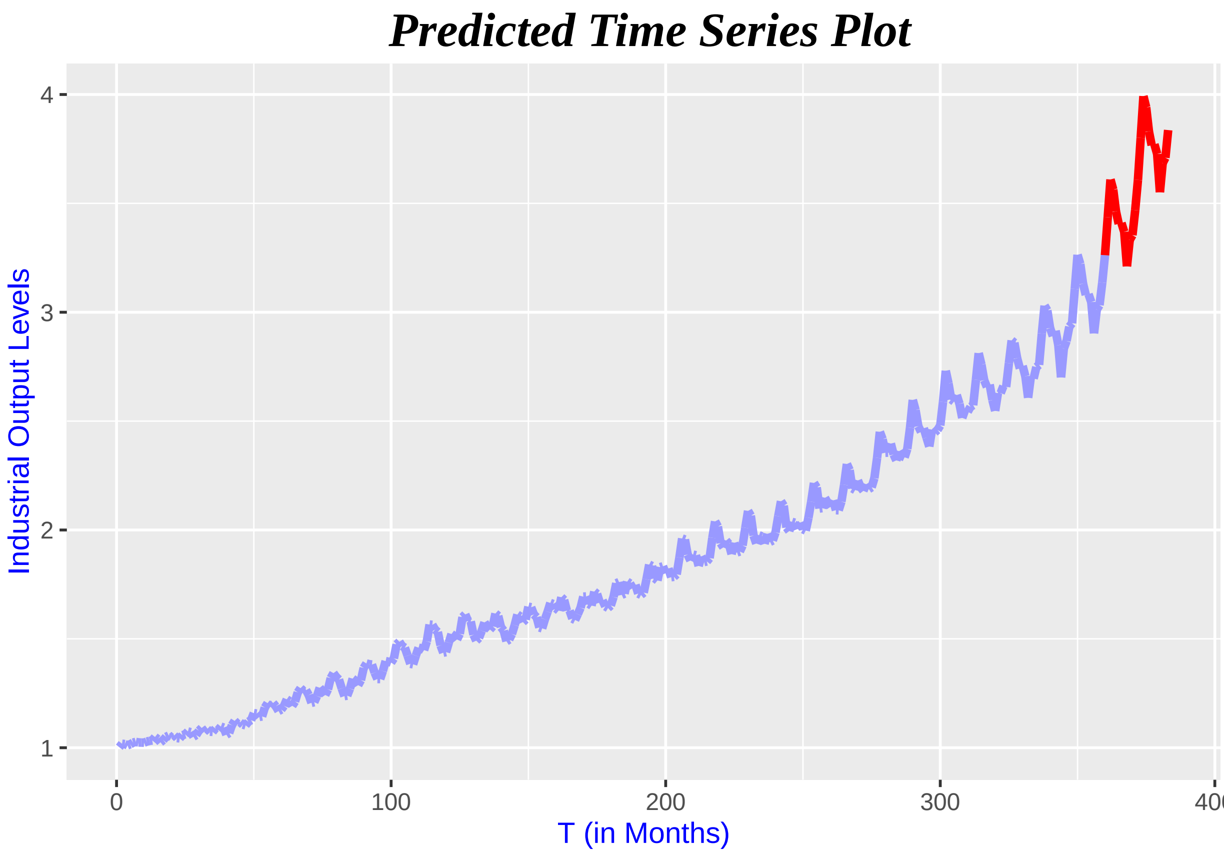




Our Model 1: AR(2,0,0) is a good fit for the data as the residuals are not correlated. Also, our model uses minimum number of parameters and hence is the best.

Forecast:

```
predition_t <- 24
y_predict <- rep(0,predition_t+2)
y_predict[1] <- logdiff_12[length(t3)-1,1]
y_predict[2] <- logdiff_12[length(t3),1]
for(i in 1:24){
    f <- y_predict[i+1]*as.numeric(myModel1Coef[1]) + y_predict[i]*as.numeric(myModel1Coef[2]) + as.nume
    y_predict[i+2] <- f
}
t0 <- length(logdata[,1])
logdata_prediction <- logdata[,1]
for(i in 1: predition_t){
    logdata_prediction[t0+i] <- logdata_prediction[t0+i-12] + y_predict[t+2]
}

data_predict <- data.frame(exp(logdata_prediction))
t4 <- 1:length(data_predict[,1])
ggplot(data_predict,aes(x=t4, y=data_predict[,1])) + geom_line(color= ifelse(t4<=359, '#9999FF','red'),
    xlab("T (in Months)") + ylab("Industrial Output Levels") +
    theme(plot.title =element_text(family="Times", face="bold.italic", size=18) ) + theme(axis.title = e
```



The time series plot in red is the prediction. We used the prediction for log diff (see the for loop). From there we get the difference predicted. Add the pred to the log data [t-12+1] and take exponential to get the graph.