STATS HW2

Zd2221

3/5/2019

```
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 3.5.2
```

```
mydata <- read_excel("/Users/zhejindong/Downloads/milk2.xls")
```

```
## readxl works best with a newer version of the tibble package.
## You currently have tibble v1.4.2.
## Falling back to column name repair from tibble <= v1.4.2.
## Message displays once per session.
```

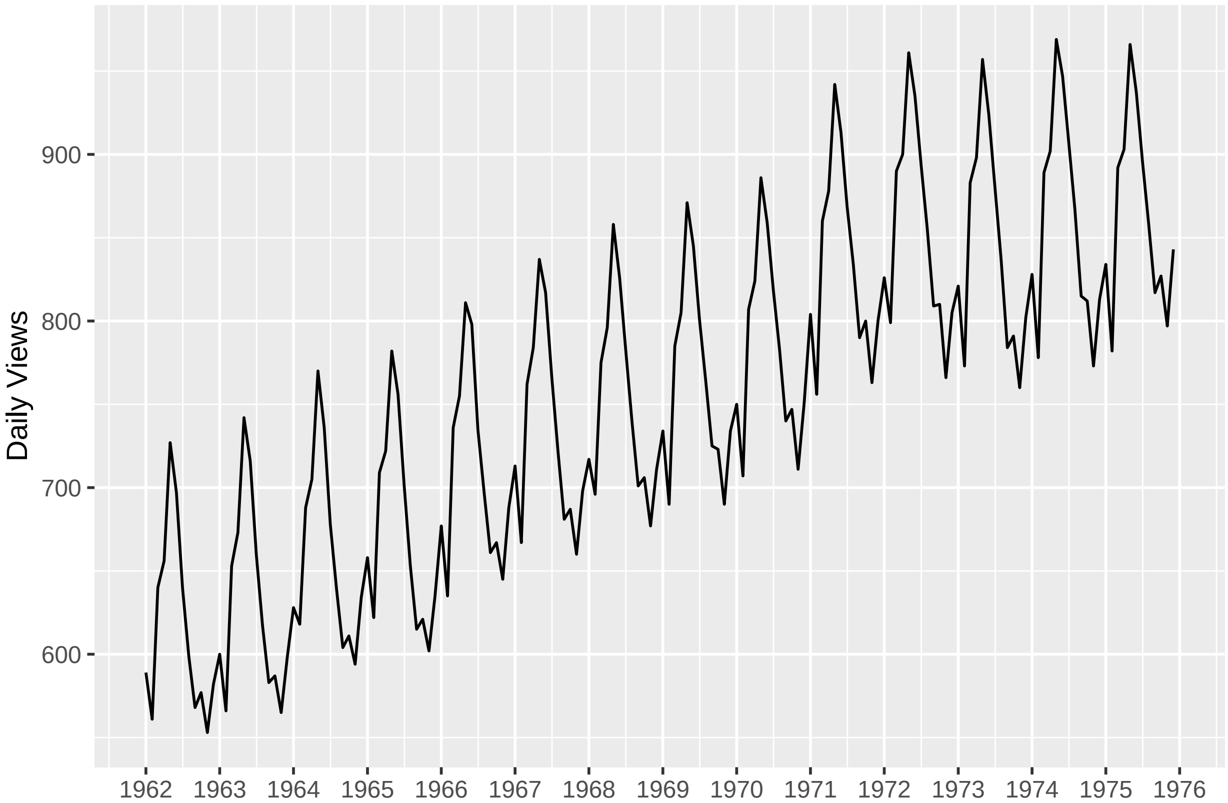
```
library("tidyverse")
```

```
## -- Attaching packages ---------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
ggplot(mydata, aes(as.Date(mydata$Month),Production)) +geom_line()+xlab("") + ylab("Daily Views")+scale_
```



We found two patterns for production data:

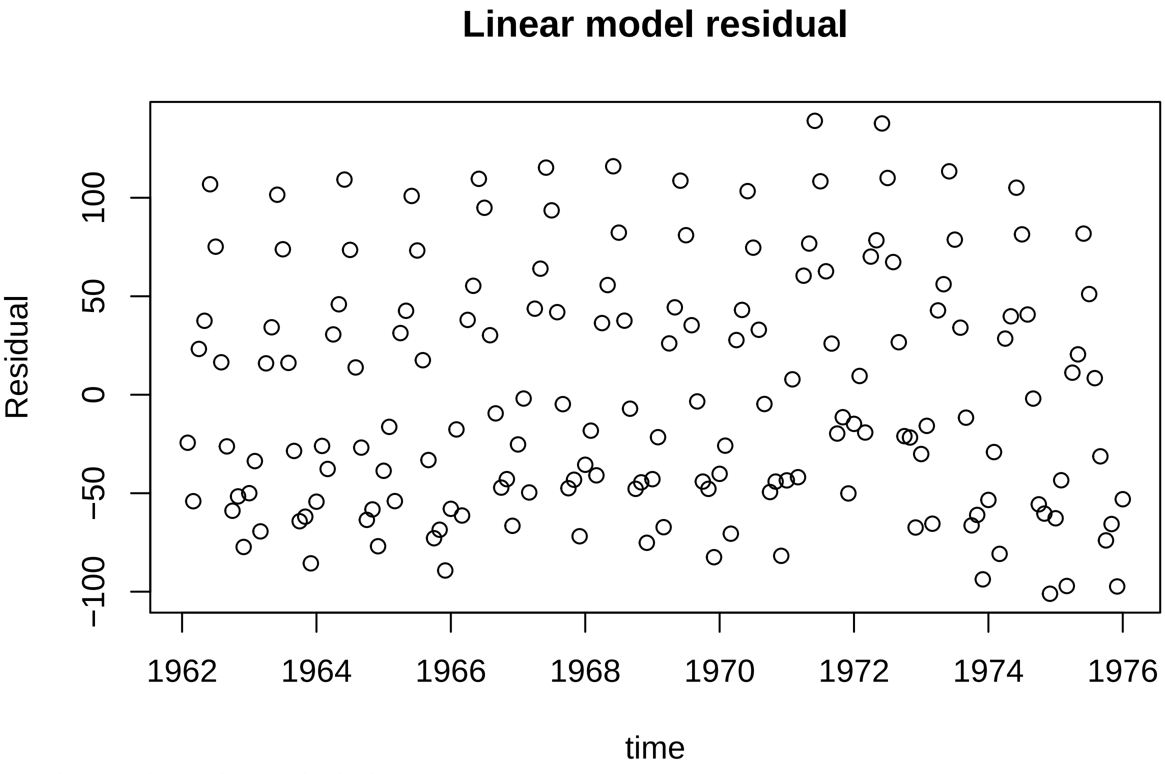
1 linear trend with years

2 Seasonality

To do time series analysis, we need to get rid of the linear trend by fitting a liner model of time.

1. Fit a linear model to the data and comment the estimated coefficients. Do you see some structure in the residuals?

```r
mydata$Time<-as.numeric(substr(mydata$Month,6,7))/12+as.numeric(substr(mydata$Month,1,4))

LModel<-lm(Production~time(Time),mydata)
plot(mydata$Time, LModel$residuals, main="Linear model residual",
   xlab="time", ylab="Residual")
```



###

conclusion about the residual plot:

There is obvious repetition seasonal pattern between the residuals. The residual has stationarity, and we can do time series analysis on that. In fact the residuals are detrended production data because the mean is arond 0 but the vairance still has a seasonal pattern, so we still need to model on the residuals.

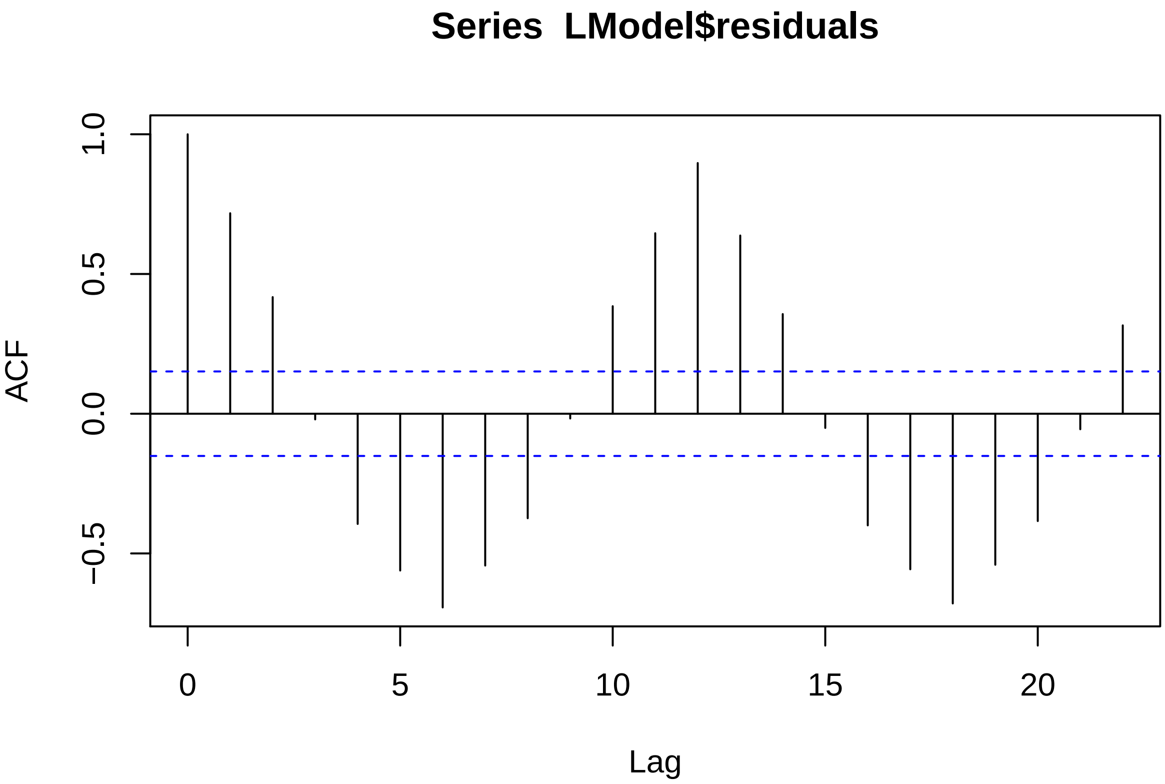
```
LModel$coefficients
```

```
## (Intercept)  time(Time)
##  611.682350    1.692615
```

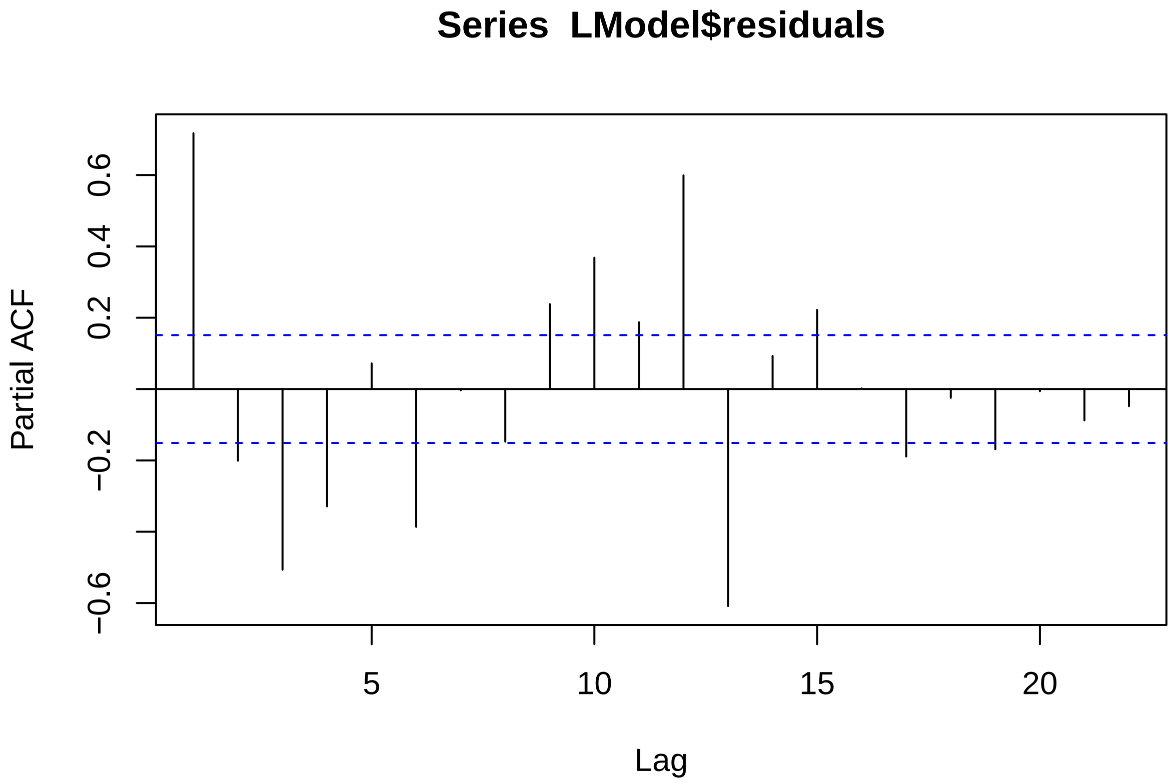
Time feature has a positive coefficient which indicates that with time passing, the production increases.

2. Plot the correlogram and partial correlogram of the residuals obtained in the previous point and comment them.

```
library(ggplot2)
ACF<-acf(LModel$residuals, type="correlation",plot = TRUE)
```



```
PACF<-pacf(LModel$residuals,plot = TRUE)
```



###

comment on acf and pacf of residuals:

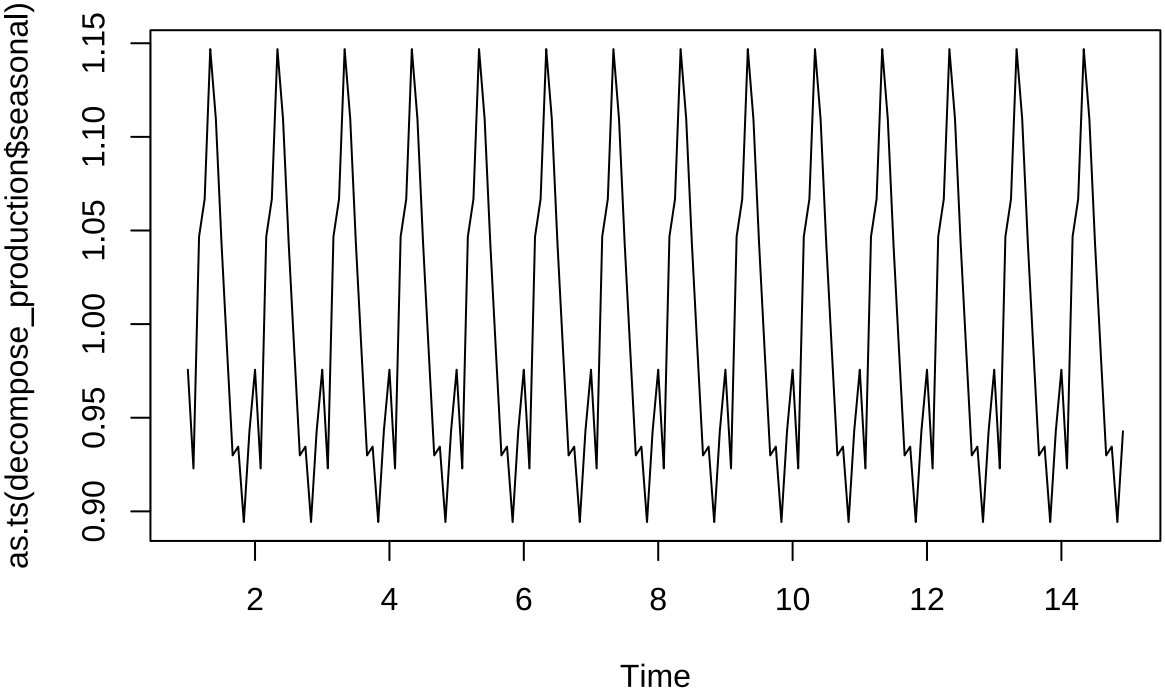
Apparent period cycles appear in both acf and pacf pictures which indicates simple linear model cannot capture the dynamic structure behind data. Residual are not independent of variable 'time' which unreveals the drawback of our model.

Besides, by observing the acf and pacf, we notice there is strongly yearly pattern for the production.Thus, we should do time series analysis on the data.

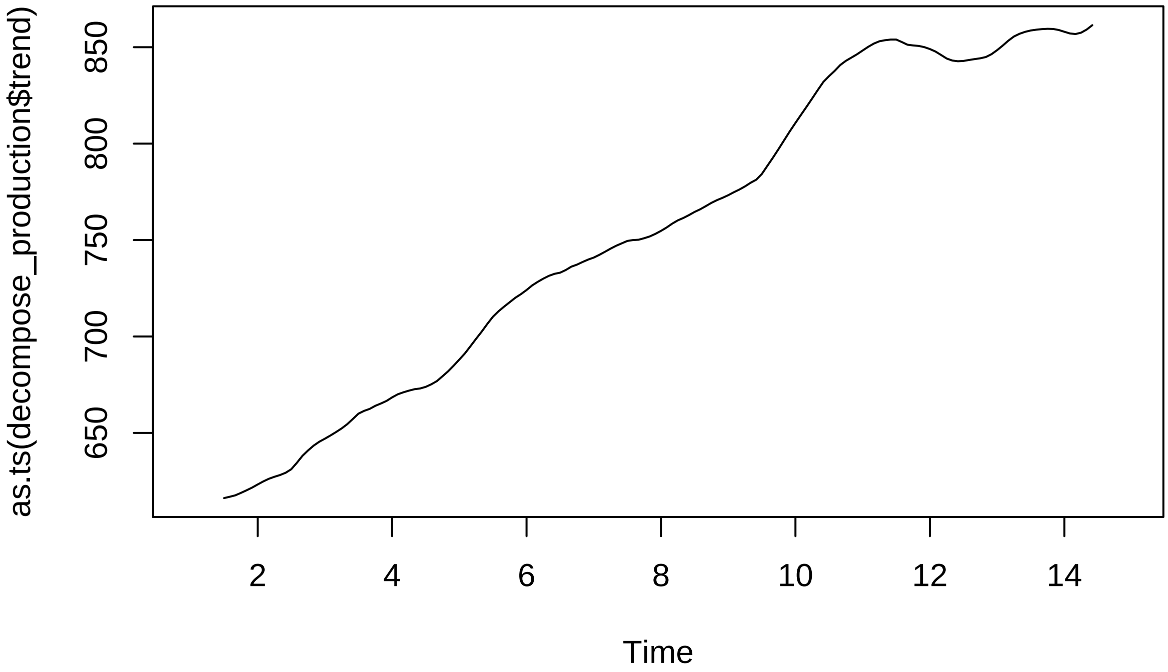
3. Try fitting an AR(1) to the data. Does it provides a better model? what about an AR(2) model?

detrend the data before time series anaylis

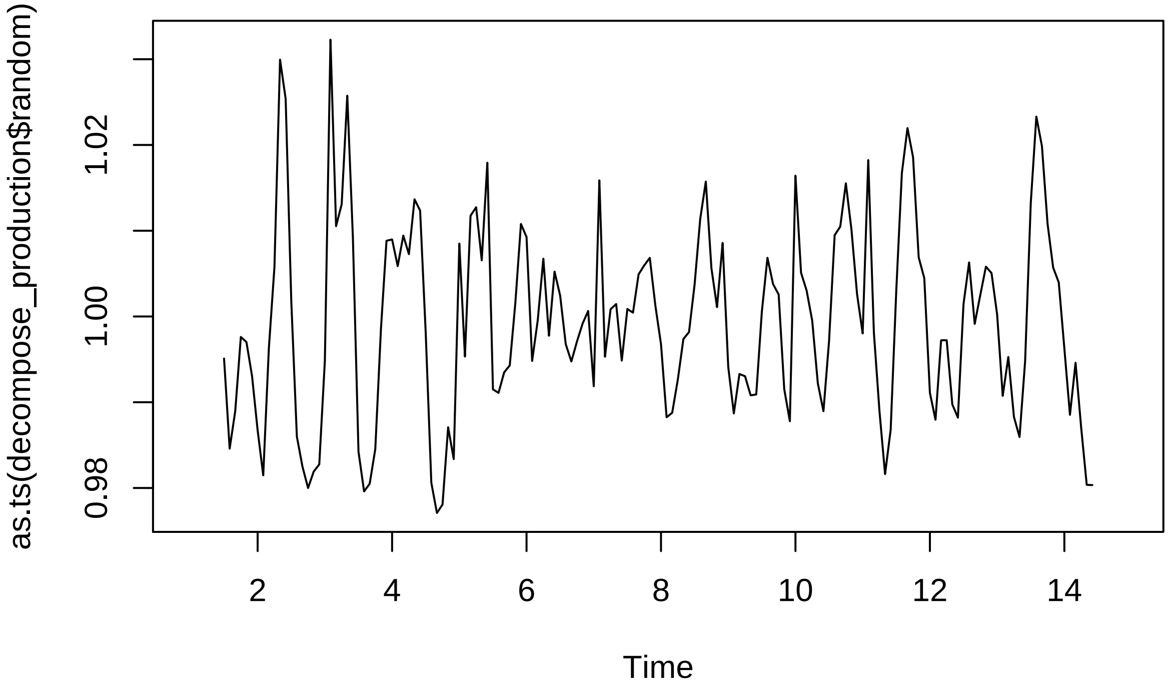
```
ts_production = ts(mydata$Production, frequency = 12)
decompose_production = decompose(ts_production, "multiplicative")
plot(as.ts(decompose_production$seasonal))
```



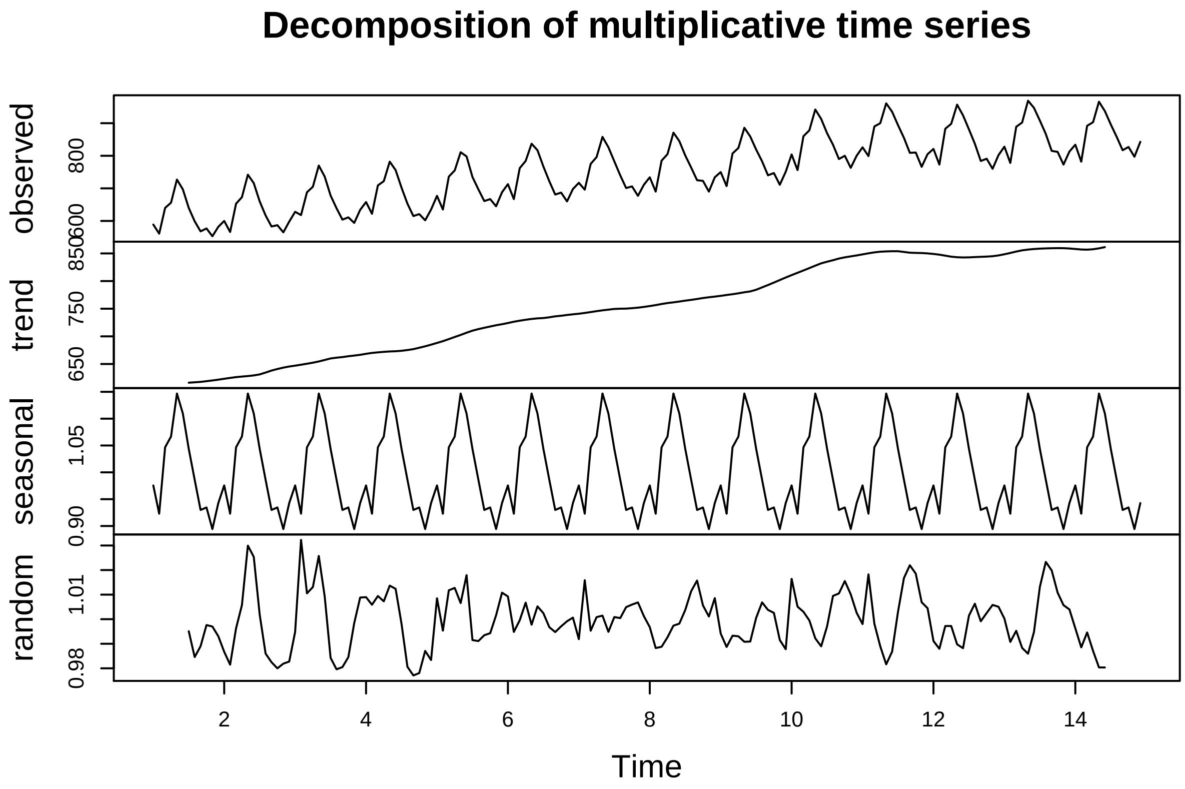
```
plot(as.ts(decompose_production$trend))
```



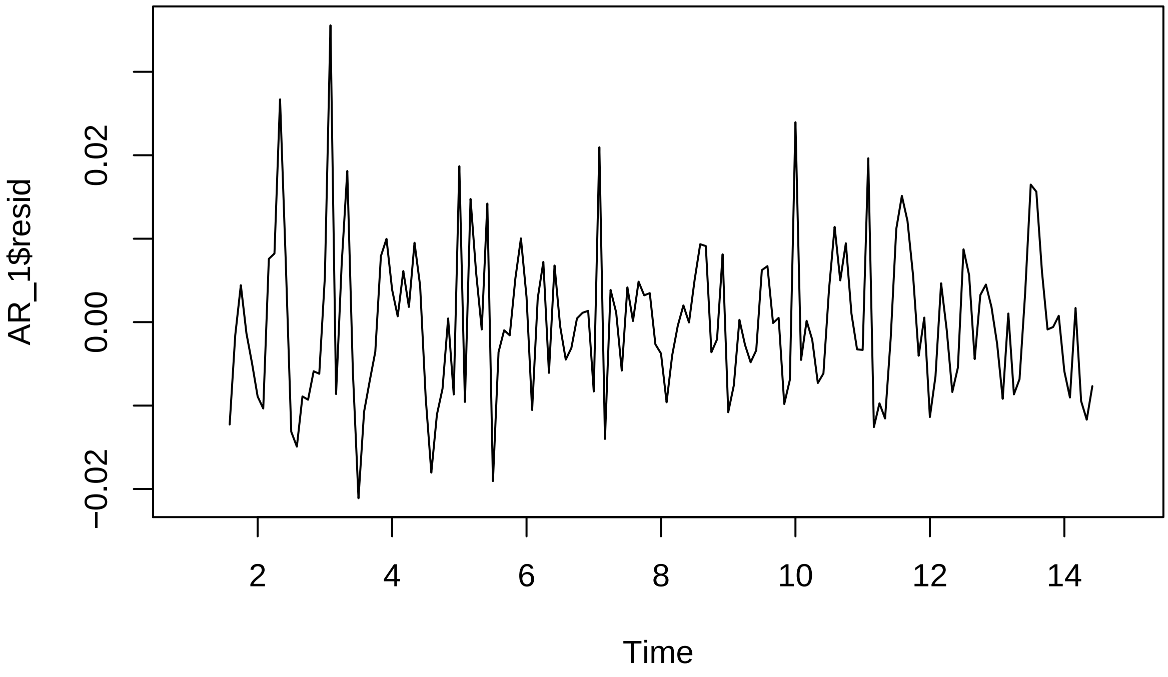
```
plot(as.ts(decompose_production$random))
```



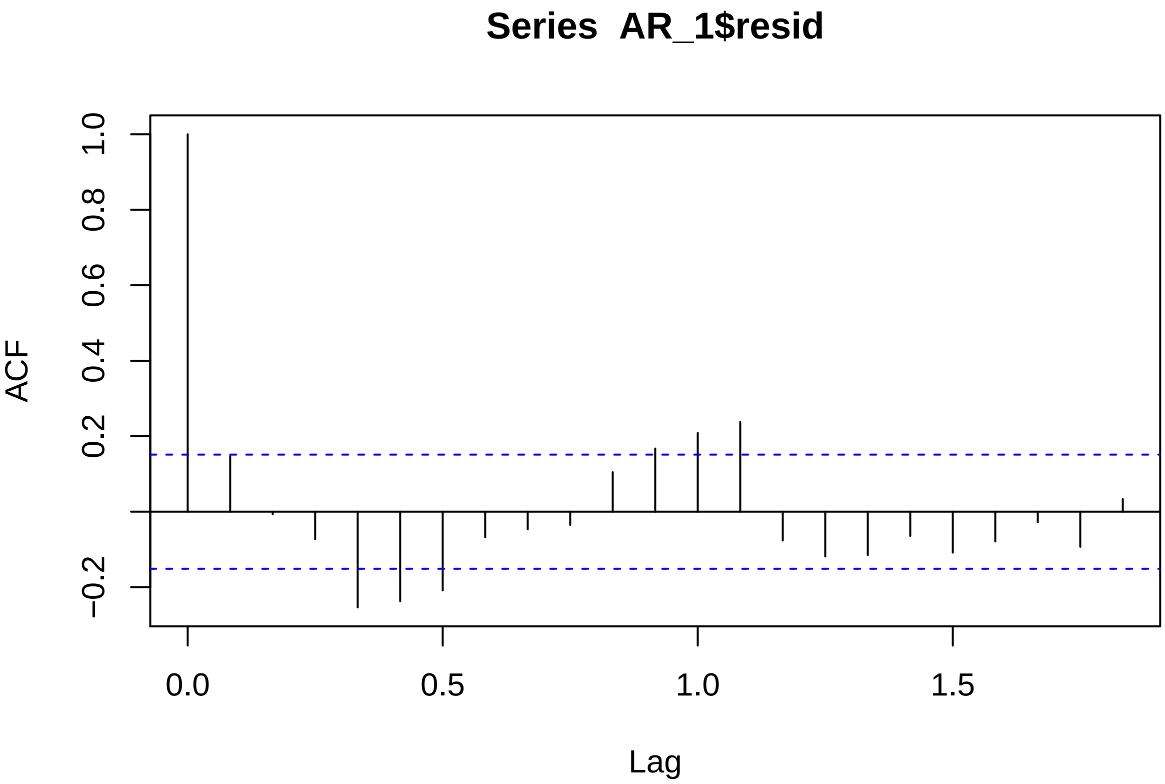
```
plot(decompose_production)
```



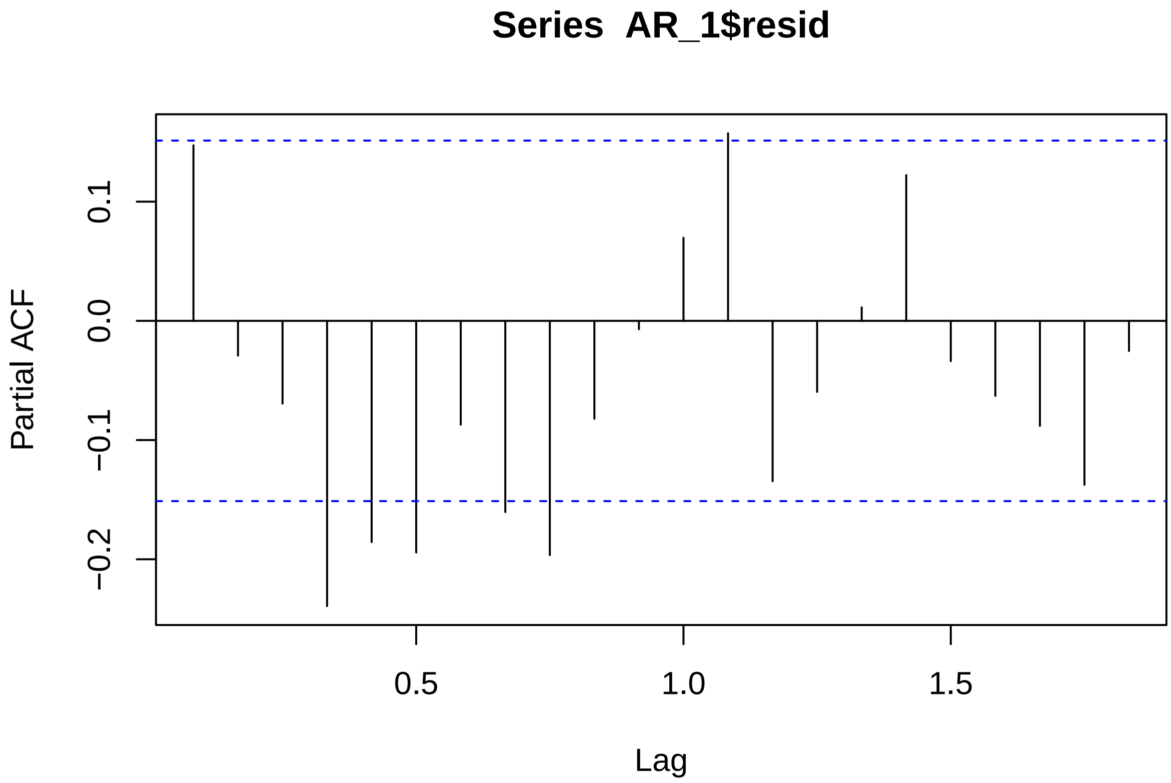
```
AR_1<-ar(decompose_production$random, aic = TRUE, order.max = 1, method=c("yule-walker"),na.action=na.pa

plot(AR_1$resid)
```



```
ACF<-acf(AR_1$resid, type="correlation",plot = TRUE,na.action=na.pass)
```

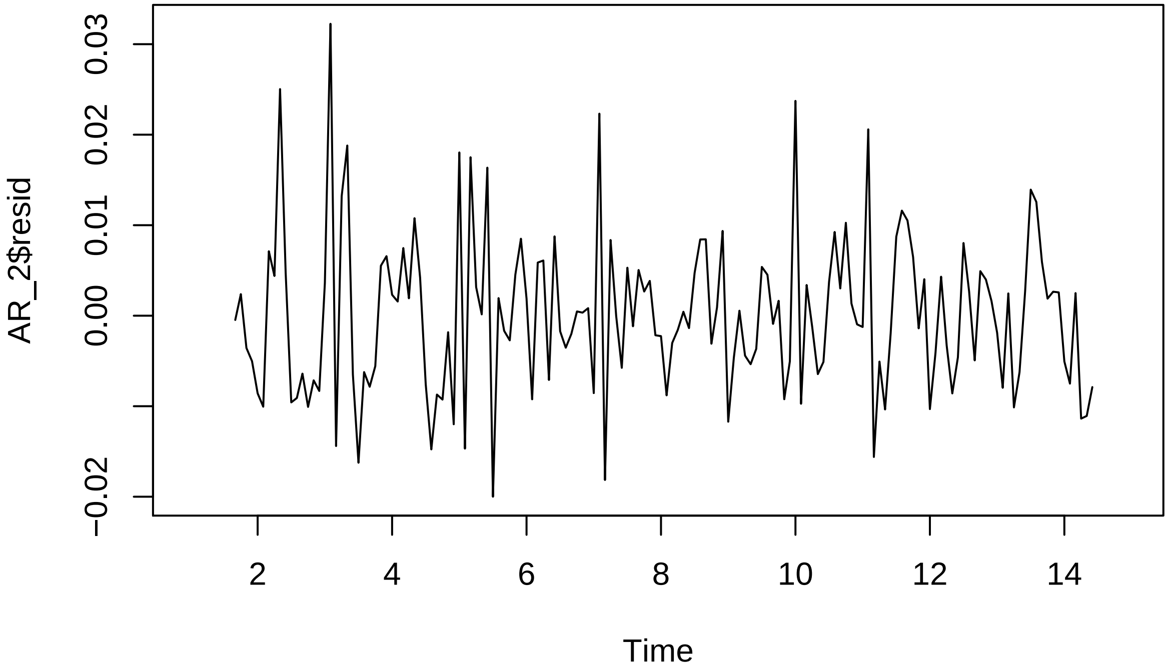


```
PACF<-pacf(AR_1$resid,plot = TRUE,na.action=na.pass)
```

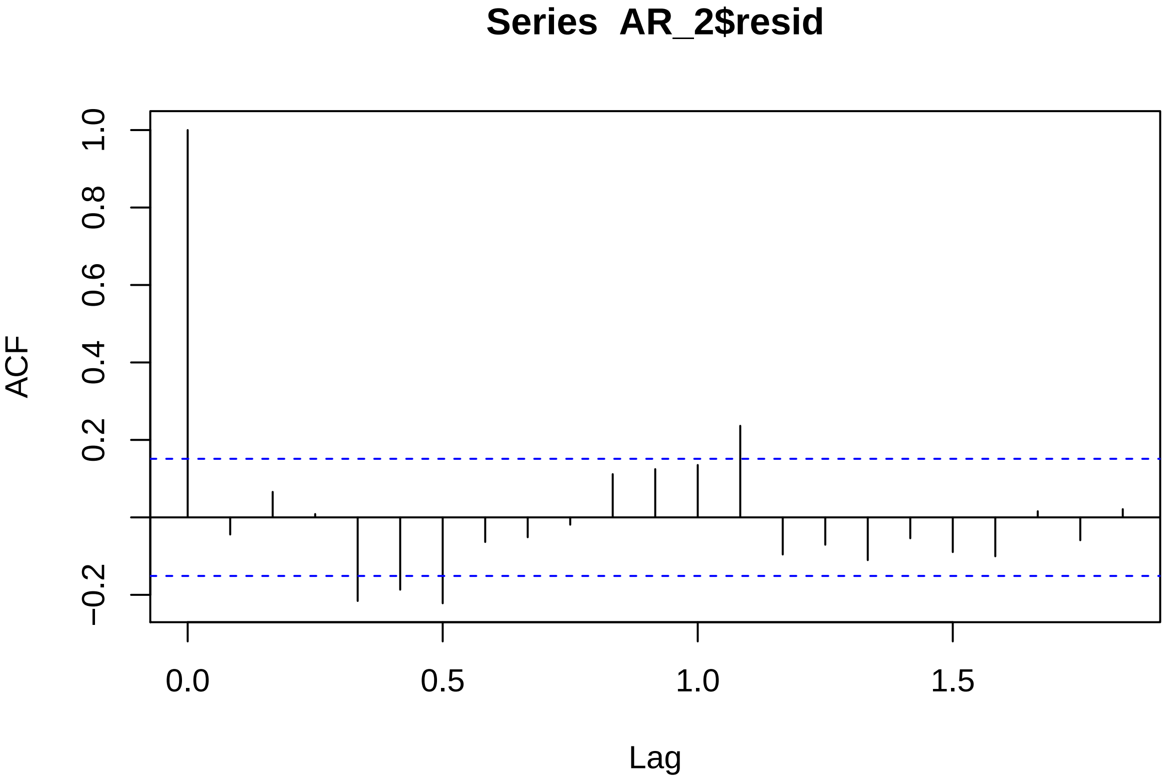


Through modeling the production data on detrended data, we can see that most of the residuals distributed in the 95% confidence interval which means that the we highly improved the independence among residuals. Thus, using AR(1) model can leverage the performance.

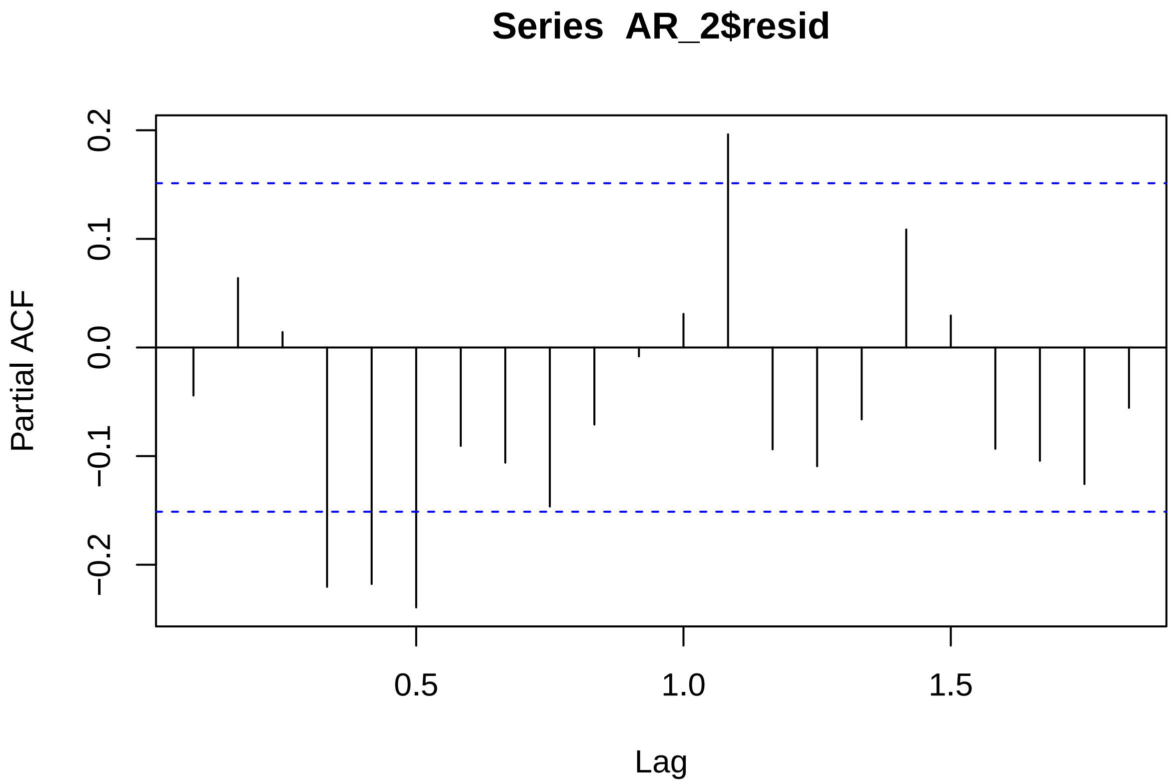
```
AR_2<-ar(decompose_production$random, aic = TRUE, order.max = 2, method=c("yule-walker"),na.action=na.pa

plot(AR_2$resid)
```



```
ACF<-acf(AR_2$resid, type="correlation",plot = TRUE,na.action=na.pass)
```



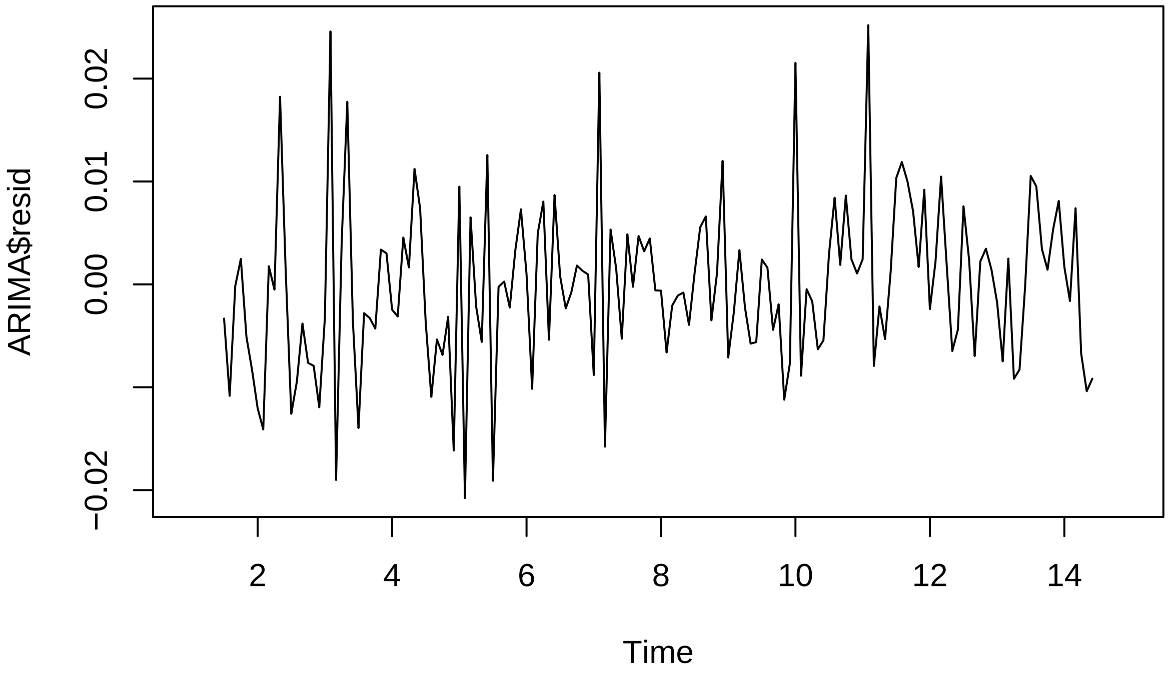
```
PACF<-pacf(AR_2$resid,plot = TRUE,na.action=na.pass)
```



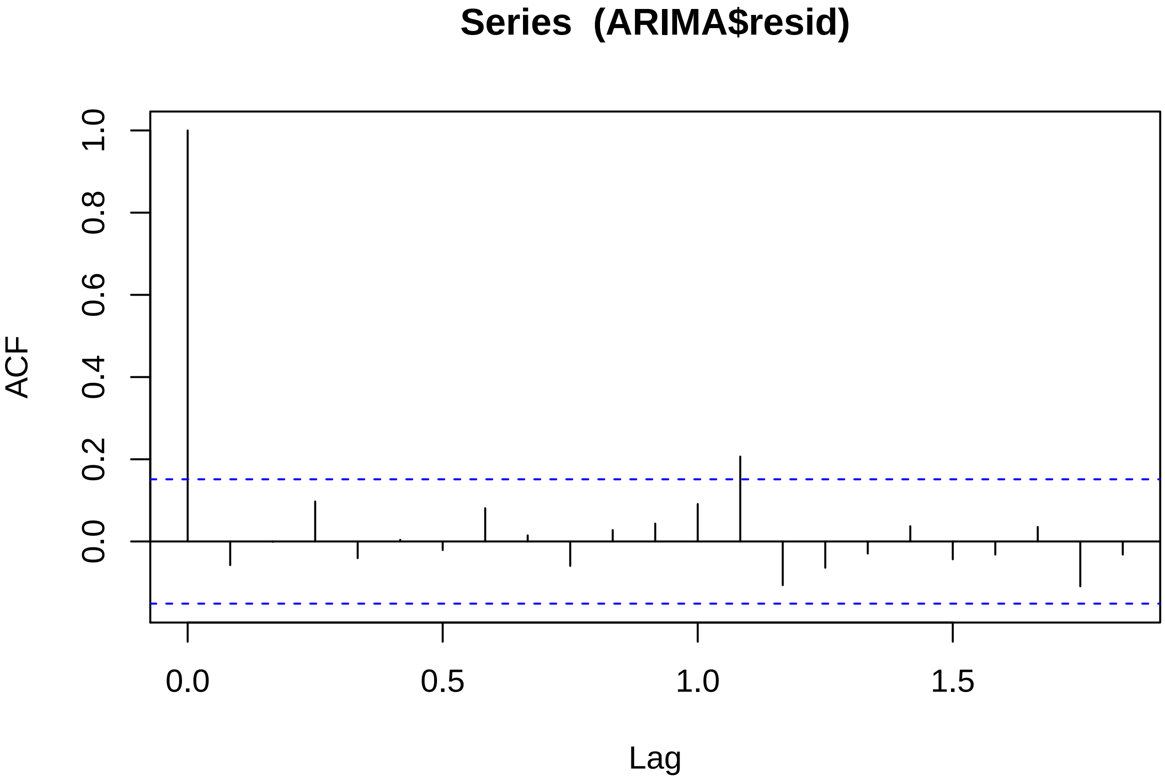
By increasing the order of AR model by 1, we can see from the ACF and PACF pictures that the dependence decreases further but now that much.

4. Try different ARMA models and present two models that best fit the datain your analysis.

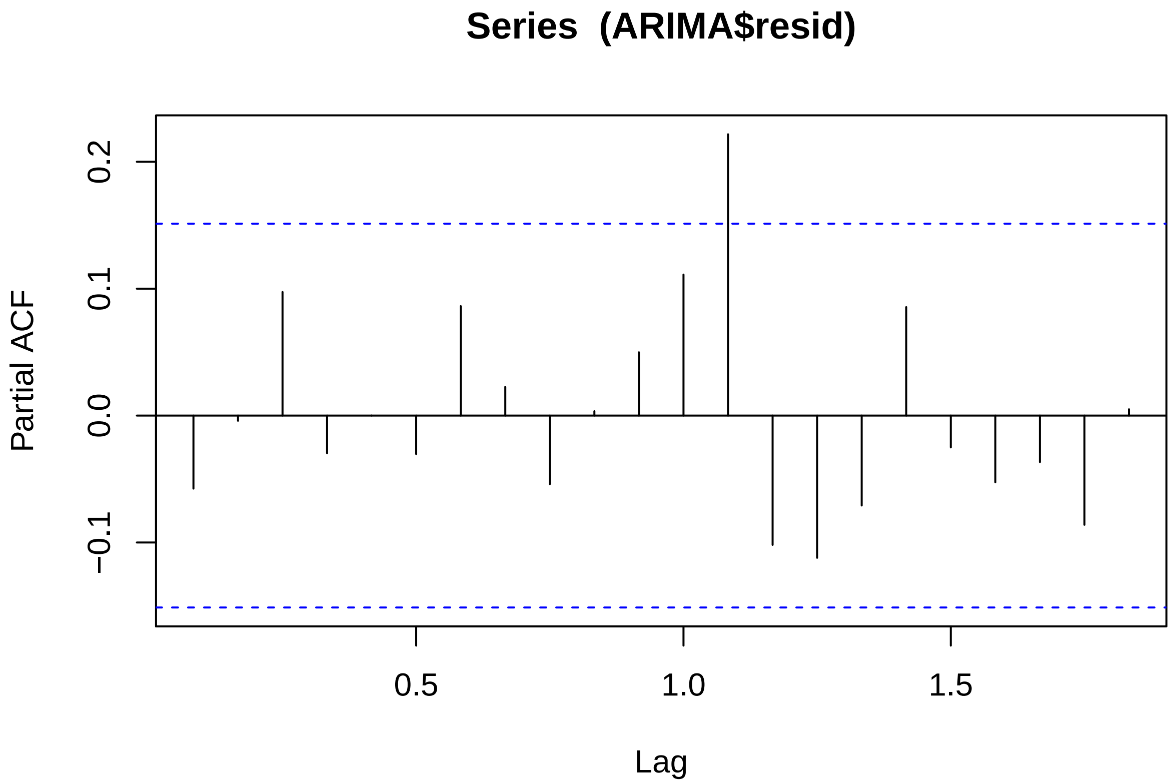
```
# fit a arma model
ARIMA<-arima(decompose_production$random, order = c(2,0,1))
plot(ARIMA$resid)
```



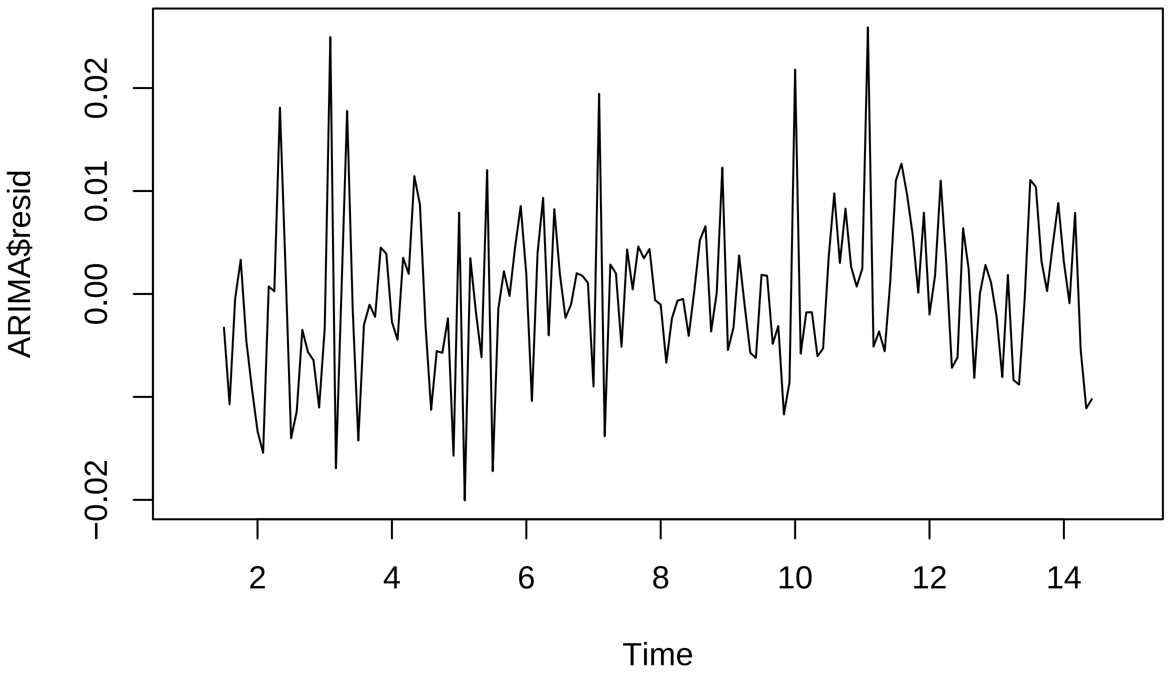
```
ACF<-acf((ARIMA$resid), type="correlation",plot = TRUE,na.action=na.pass)
```



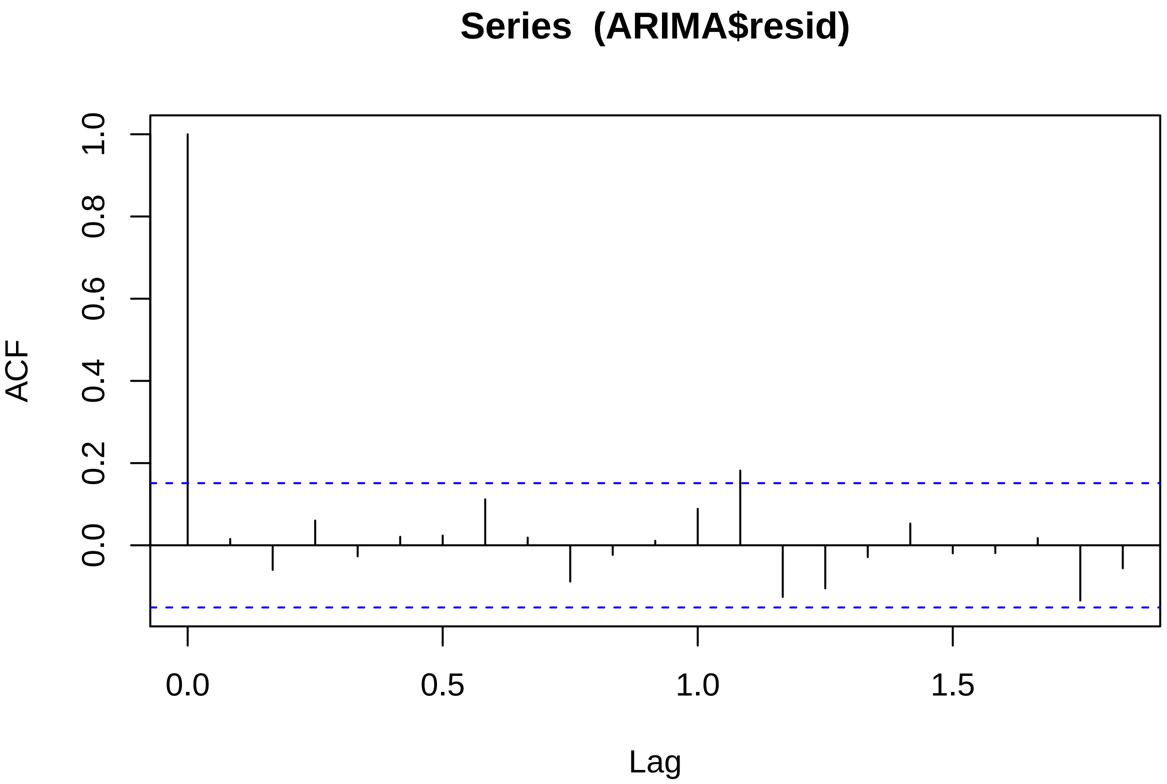
```
PACF<-pacf((ARIMA$resid),plot = TRUE,na.action=na.pass)
```



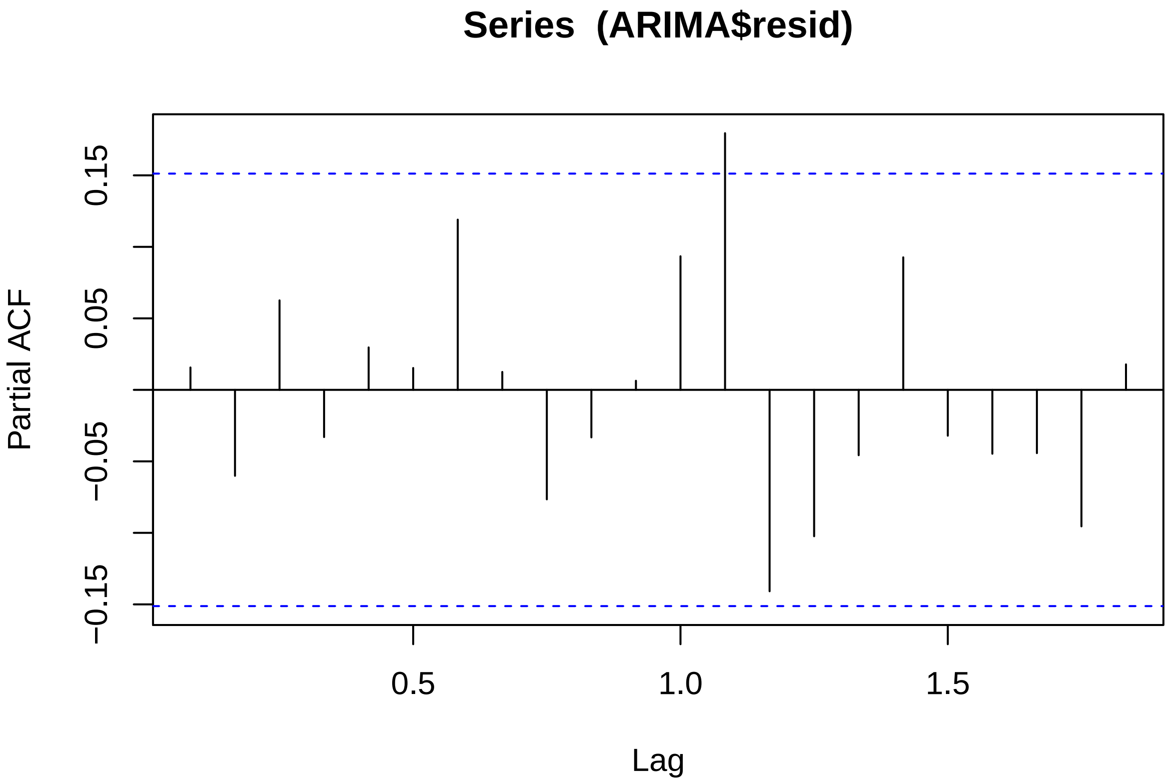
```
ARIMA<-arima(decompose_production$random, order = c(2,0,2))
plot(ARIMA$resid)
```



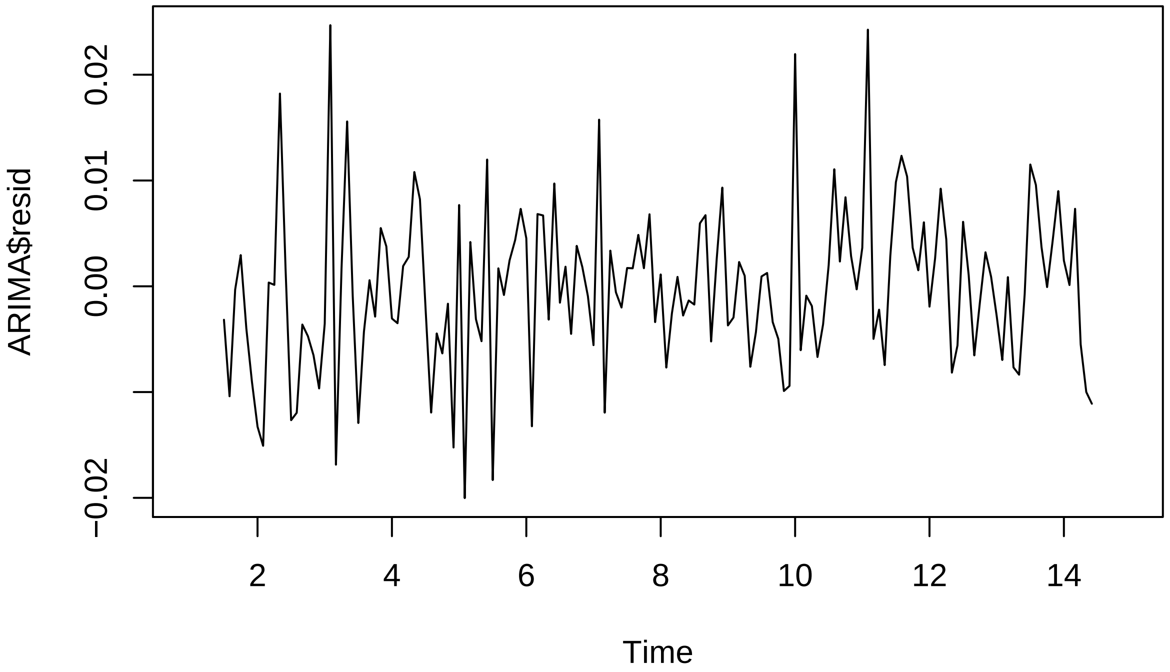
```
ACF<-acf((ARIMA$resid), type="correlation",plot = TRUE,na.action=na.pass)
```



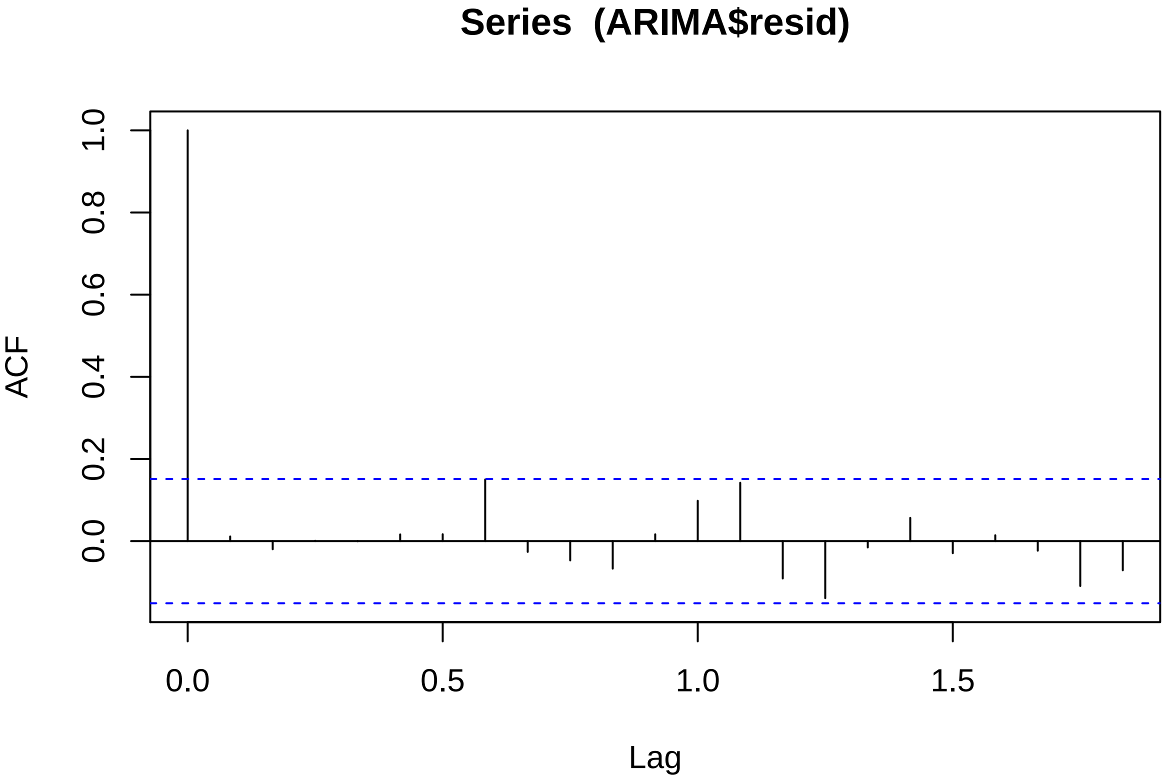
```
PACF<-pacf((ARIMA$resid),plot = TRUE,na.action=na.pass)
```



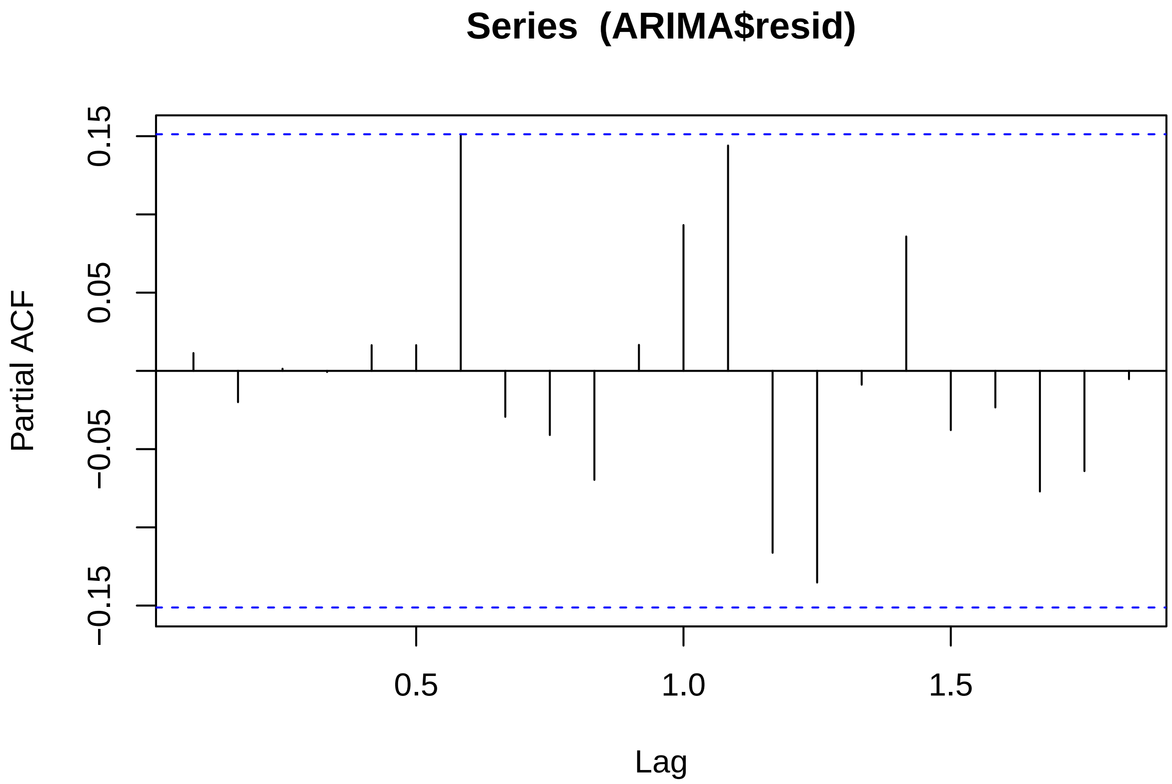
```
ARIMA<-arima(decompose_production$random, order = c(4,0,4))
plot(ARIMA$resid)
```



```
ACF<-acf((ARIMA$resid), type="correlation",plot = TRUE,na.action=na.pass)
```



```
PACF<-pacf((ARIMA$resid),plot = TRUE,na.action=na.pass)
```



By tuning the parameter of order in ARIMA model, we foud when order = c(4,0,4)), almost all residuals distribute in the 95% confidence interval. That is good sign that our residual behaves quite like white noise so the ARIMA model with order = c(4,0,4) can capture most variance of data.