Macroeconometrics

PS1

Group Member: Jianqiu Bei, Xinru Huang, Jingyi Liu, Zhirui Wang

1. (a) $y_1 = 20 + 0.4 \times 100$

$y_2 = 20 + 0.4 \times (20 + 0.4 \times 100)$

$y_3 = 20 + 0.4 \times (20 + 0.4 \times (20 + 0.4 \times 100))$

$y_t = 20 \times \sum_{i=0}^{t-1} 0.4^i + 0.4^t \times 100$

(b) $y_t = \frac{-20}{1.4} + \frac{1}{1.4} \times y_{t+1}$

$y_{t+n-1} = -\frac{20}{1.4} + \frac{1}{1.4} \times 100$

$y_{t+n-2} = -\frac{20}{1.4} + \frac{1}{1.4}\left(-\frac{20}{1.4} + \frac{1}{1.4} \times 100\right)$

$y_{t+n-3} = -\frac{20}{1.4} + \frac{1}{1.4}\left(-\frac{20}{1.4} + \frac{1}{1.4}\left(-\frac{20}{1.4} + \frac{1}{1.4} \times 100\right)\right)$

$y_t = -\frac{20}{1.4} \sum_{i=0}^{n-1} \left(\frac{1}{1.4}\right)^i + \left(\frac{1}{1.4}\right)^n \times 100$

(c) $\Delta y_t = 20$

$y_t = 100 + 20 \times t$

(d) $\Delta y_t = \varepsilon_t$

$y_t = 100 + \sum_{i=1}^{t} \varepsilon_i, \quad t \geq 1$

(e) $y_t = \alpha L y_t + \varepsilon_t$

$(1 - \alpha L) y_t = \varepsilon_t$

$y_t = \frac{\varepsilon_t}{1 - \alpha L}, \quad |\alpha| < 1$

$y_t = \sum_{i=0}^{\infty} \alpha^i L^i \varepsilon_t = \sum_{i=0}^{\infty} \alpha^i \varepsilon_{t-i}$

(f) $y_t = y_{t-1} + \varepsilon_t$

$\Delta y_t = \varepsilon_t$

$y_t = y_0 + \sum_{i=1}^{t} \varepsilon_i, \quad t \geq 1$

Problem Set 1

Jianqiu Bei, Xinru Huang, Jingyi Liu, Zhirui Wang

June 14, 2016

Problem 2

(a)

assign assumed values

```
a <- 50
b <- 1
gama <- 1
beta <- 0.5
t <- c(1:100)
d <- rep(NA,100)
s <- rep(NA,100)
p <- rep(NA,100)
epsilon <- 1
```

start from the steady state

```
p[1] <- (a-b)/(beta+gama)
d[1] <- a-gama*p[1]
s[1] <- d[1]
```

give the system a shock

```
s[2] <- b+beta*p[1]+epsilon
d[2] <- s[2]
p[2] <- (a-d[2])/gama
for(i in 3:100){
  s[i] <- b+beta*p[i-1]
  d[i] <- s[i]
  p[i] <- (a-d[i])/gama
}
```

calculate IM and IRF

```
IM <- (p[2]-p[1])/epsilon
print(IM)
```

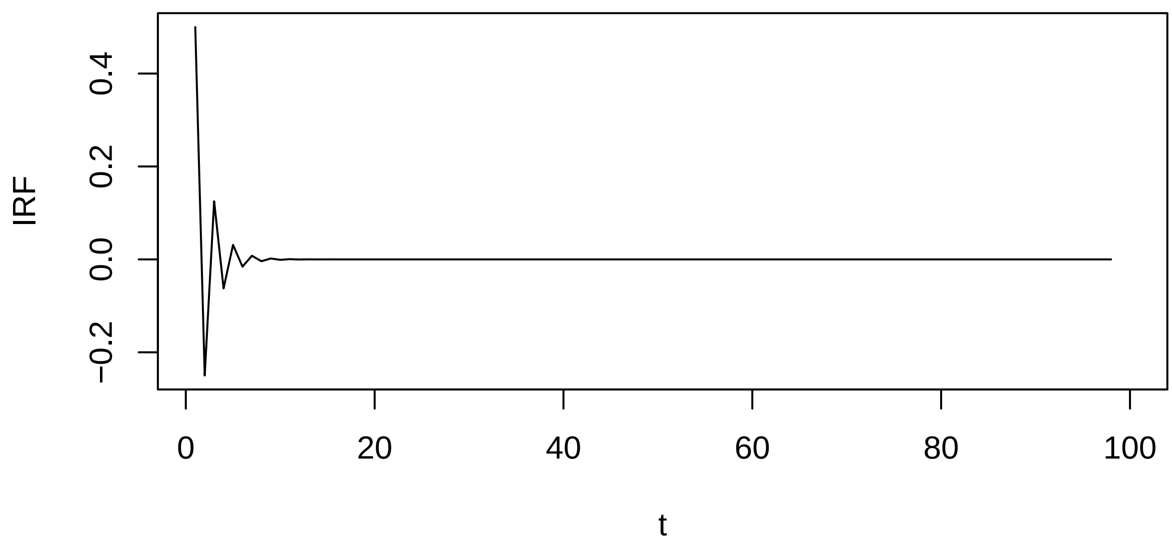
```
## [1] -1
```

```
IRF <- rep(NA,100)
for(i in 1:100){
  IRF[i] <- (p[i+2]-p[1])/epsilon
}
print(round(IRF[1:30],5))
```

```
##  [1]  0.50000 -0.25000  0.12500 -0.06250  0.03125 -0.01562  0.00781
##  [8] -0.00391  0.00195 -0.00098  0.00049 -0.00024  0.00012 -0.00006
## [15]  0.00003 -0.00002  0.00001  0.00000  0.00000  0.00000  0.00000
## [22]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [29]  0.00000  0.00000
```

plot IRF

```r
plot(t,IRF,type = "l")
```



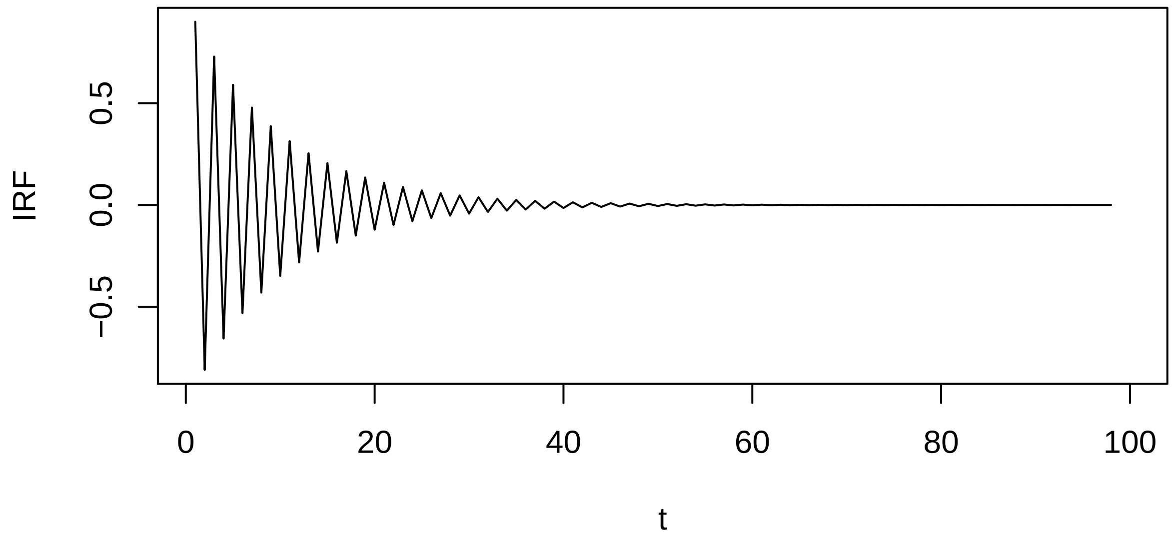
(b)

change the assumed value of beta=0.9

IM and IRF are

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

```
##  [1]  0.90000 -0.81000  0.72900 -0.65610  0.59049 -0.53144  0.47830
##  [8] -0.43047  0.38742 -0.34868  0.31381 -0.28243  0.25419 -0.22877
## [15]  0.20589 -0.18530  0.16677 -0.15009  0.13509 -0.12158  0.10942
## [22] -0.09848  0.08863 -0.07977  0.07179 -0.06461  0.05815 -0.05233
## [29]  0.04710 -0.04239  0.03815 -0.03434  0.03090 -0.02781  0.02503
## [36] -0.02253  0.02028 -0.01825  0.01642 -0.01478  0.01330 -0.01197
## [43]  0.01078 -0.00970  0.00873 -0.00786  0.00707 -0.00636  0.00573
## [50] -0.00515  0.00464 -0.00417  0.00376 -0.00338  0.00304 -0.00274
## [57]  0.00247 -0.00222  0.00200 -0.00180
```



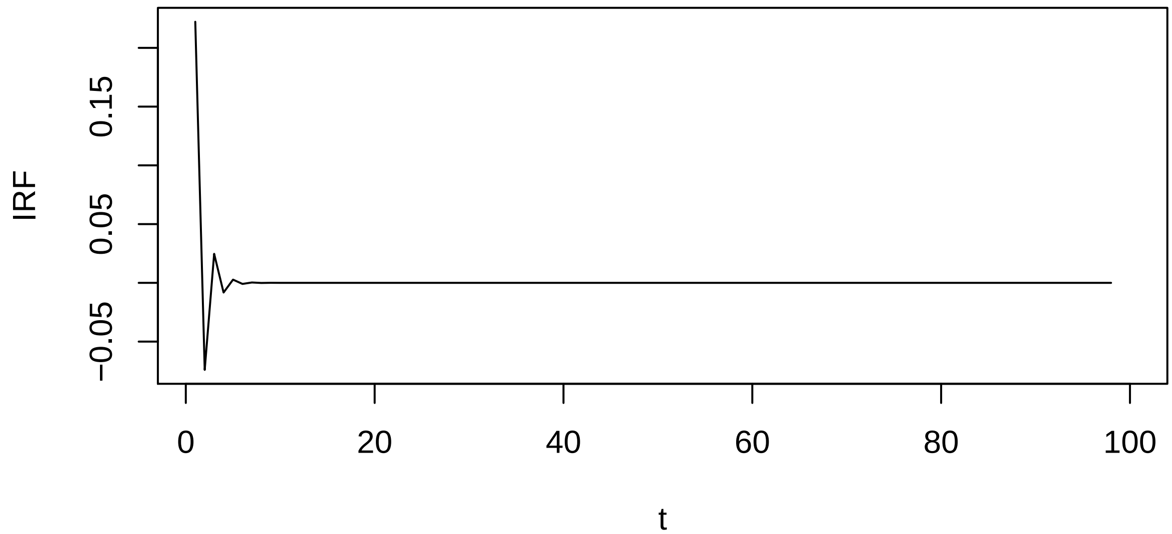
We can see that the value of IM is remained unchanged, but IRF converges slower. This because IM is determined only by gama, but IRF is determined by both gama and beta. And with a larger beta, the system needs a longer time to reach the steady state again.

remain beta=0.5,but change the assumed value of gama=0.9

IM and IRF are

```
## [1] -0.6666667
```

```
##  [1]  0.22222 -0.07407  0.02469 -0.00823  0.00274 -0.00091  0.00030
##  [8] -0.00010  0.00003 -0.00001  0.00000  0.00000  0.00000  0.00000
## [15]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [22]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [29]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [36]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [43]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [50]  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
## [57]  0.00000  0.00000  0.00000  0.00000
```

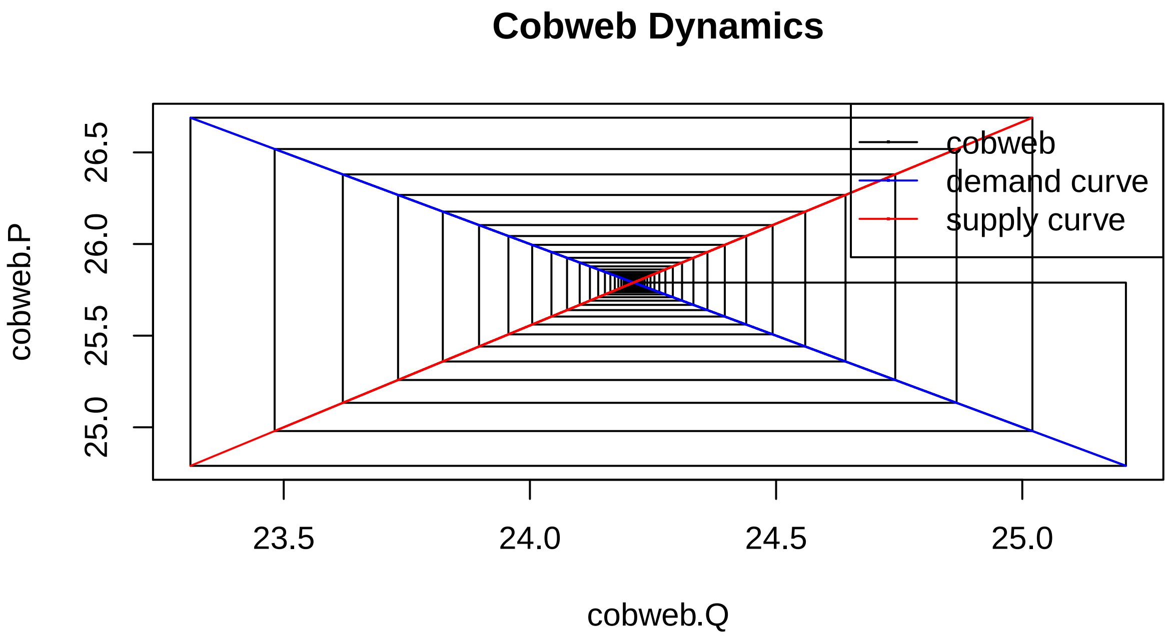


With a larger gama, IM is smaller,IRF converges to zero in a shorter period, and the system reaches the steady state in a shorter period.

(c)

draw the cobweb plot

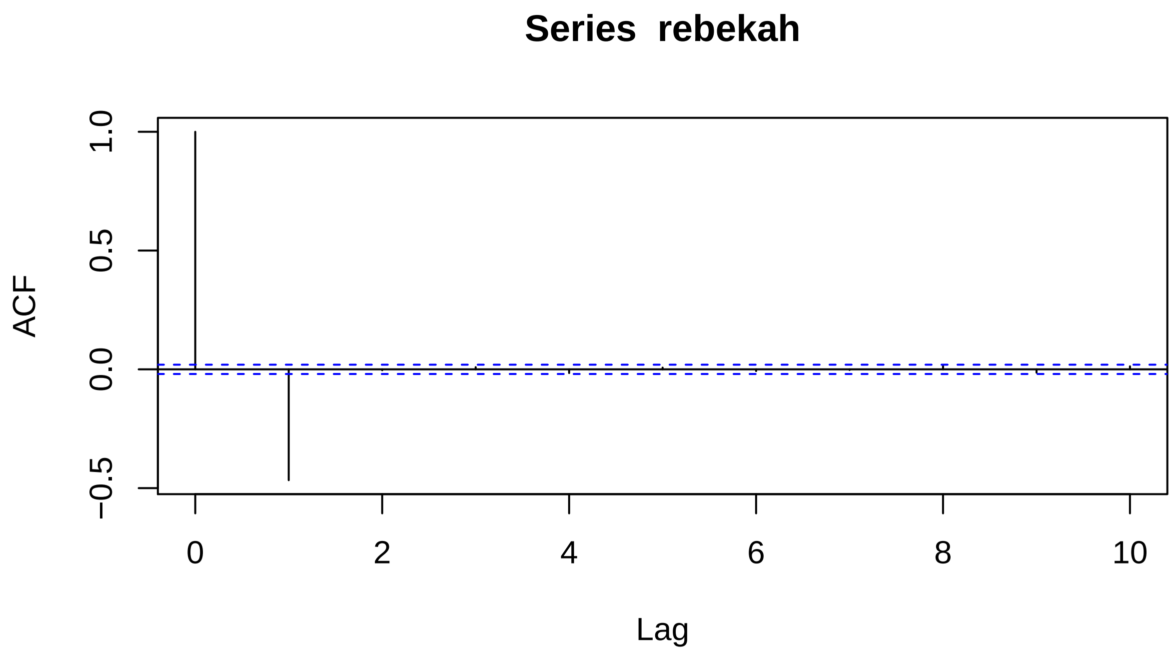
```
cobweb.Q <- rep(NA,200)
cobweb.P <- rep(NA,200)
for(i in 1:100){
  cobweb.Q[2*i-1] <- s[i]
  cobweb.Q[2*i] <- d[i+1]
  cobweb.P[2*i-1] <- p[i]
  cobweb.P[2*i] <- p[i]
}
plot(cobweb.Q,cobweb.P,type="l",main = "Cobweb Dynamics")
lines(d,p,col="blue")
lines(s[-c(1,2)],p[-c(1,100)],col="red")
legend("topright",c("cobweb","demand curve","supply curve"),
       lty = c(1,1,1),pch = c(46,46,46),col = c("black","blue","red"))
```



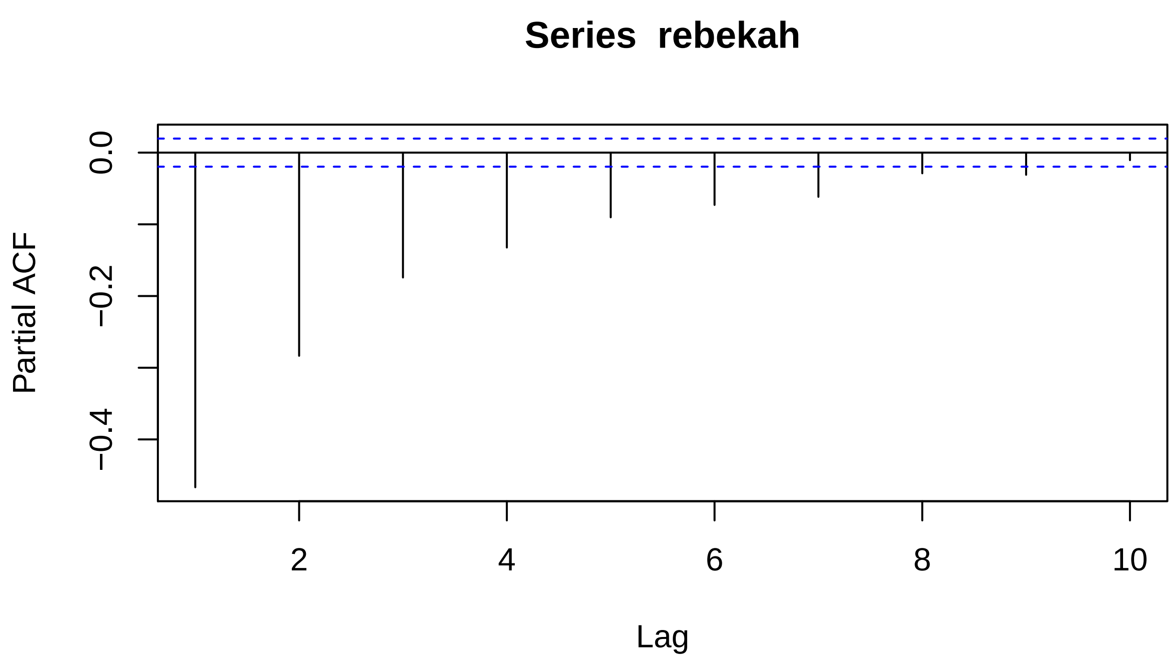
It is called cobweb model because the dynamic process of “s>d>s>d>……” convergence is like a cobweb on the supply-demand graph. In addition, even when the system do not converge, the graph will still show a gradually bigger and bigger cobweb.

Problem 3

```
sometime <- read.csv("D:/Dropbox/16summer/Macroeconometrics/Problem Set1/sometimeseriesdata.csv")
suppressMessages(attach(sometime))
acf(rebekah,10)
```

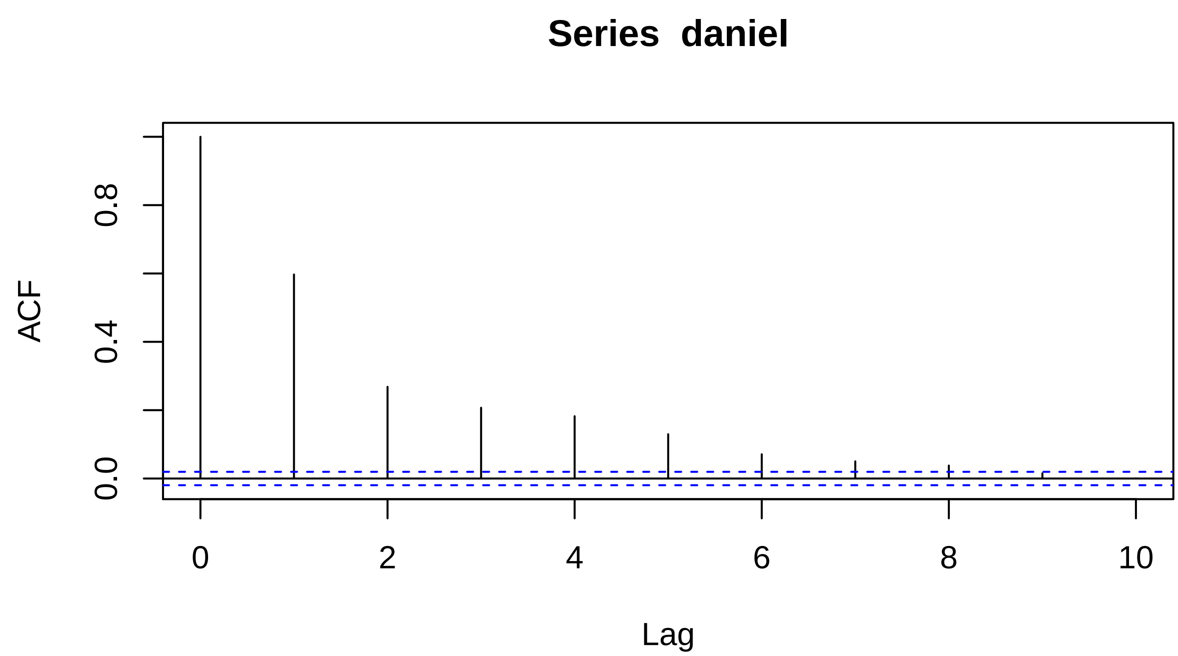



```
pacf(rebekah,10)
```

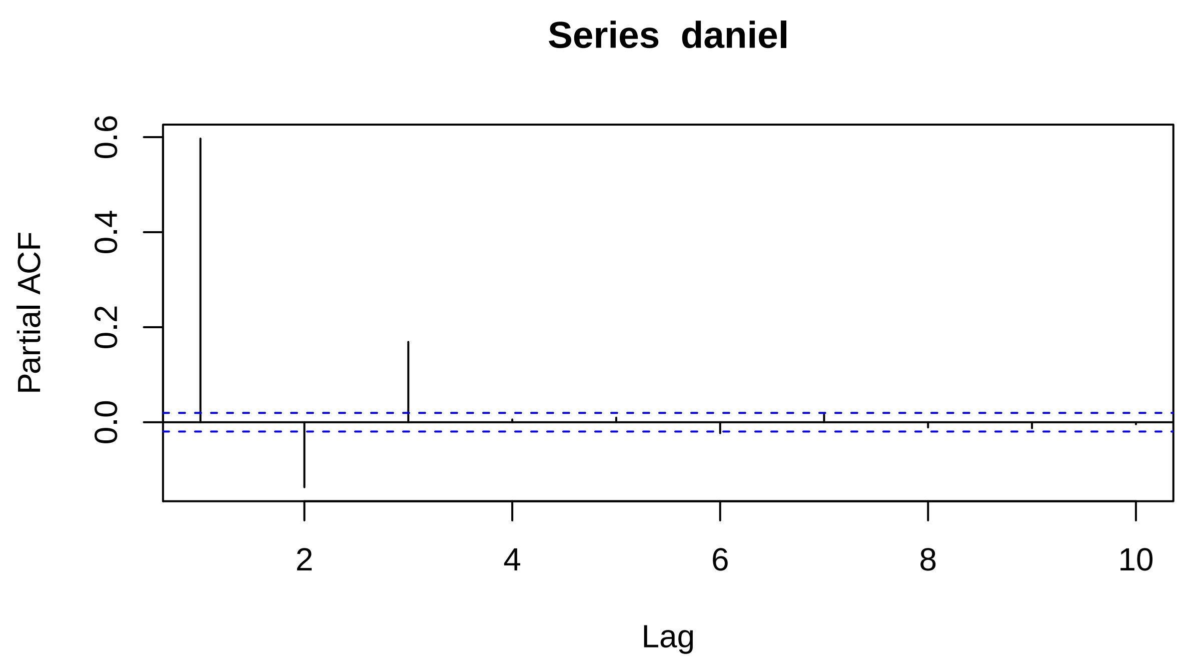


MA(1),beta<0

```
acf(daniel,10)
```

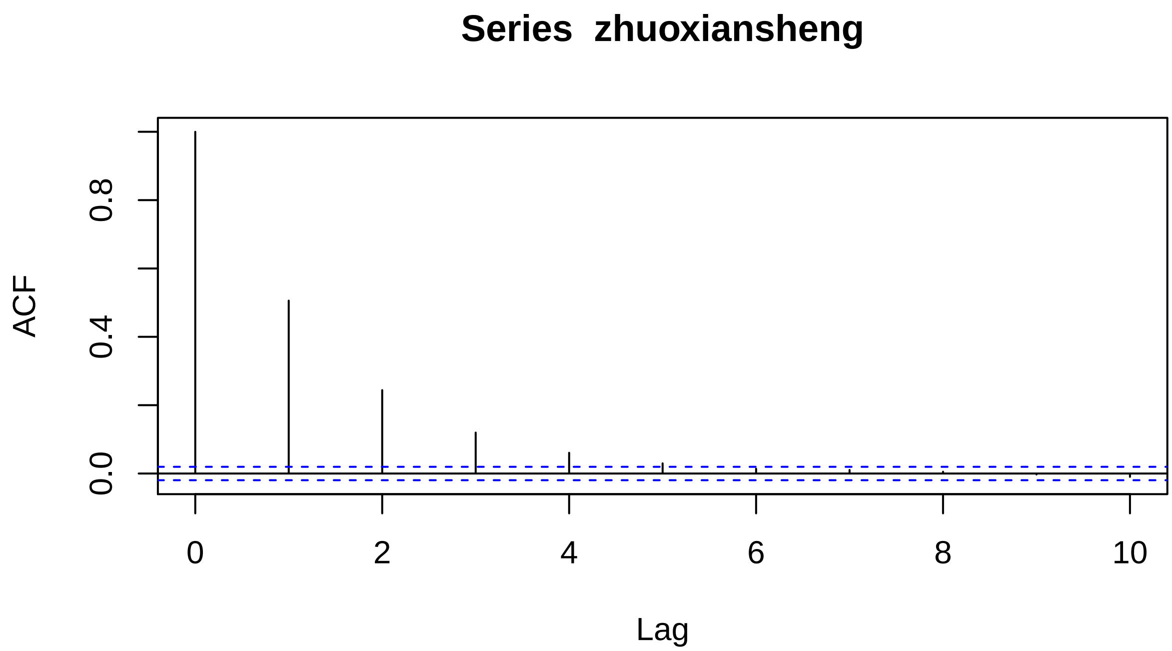



```
pacf(daniel,10)
```

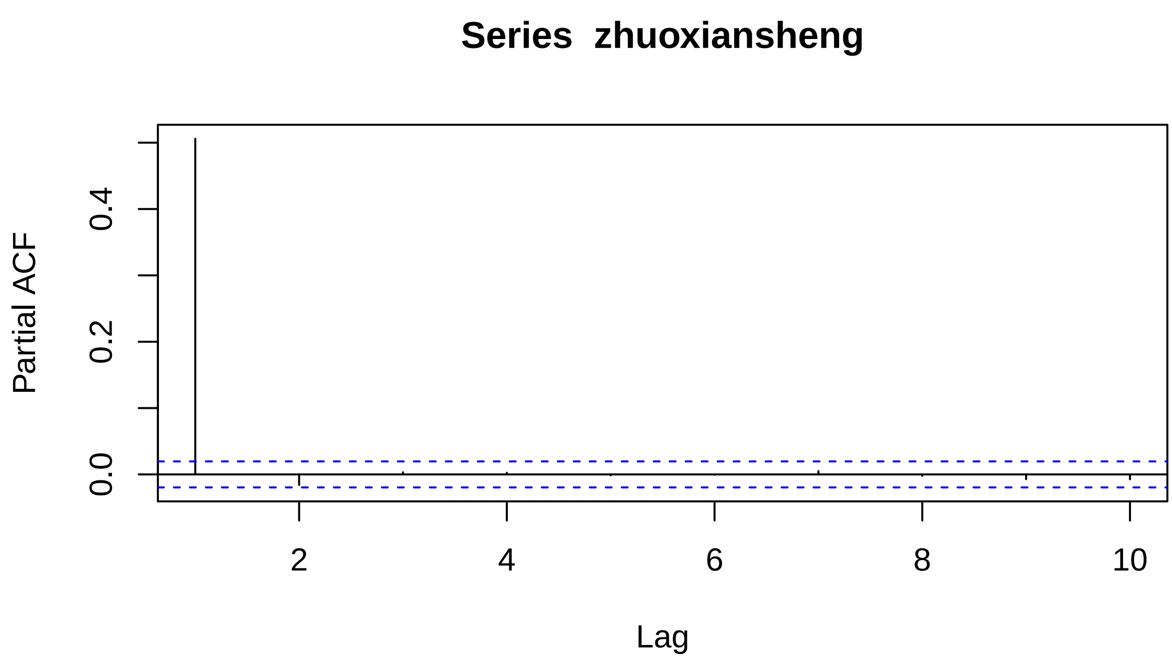


ARMA(1,1),a1>0

```
acf(zhuoxiansheng,10)
```

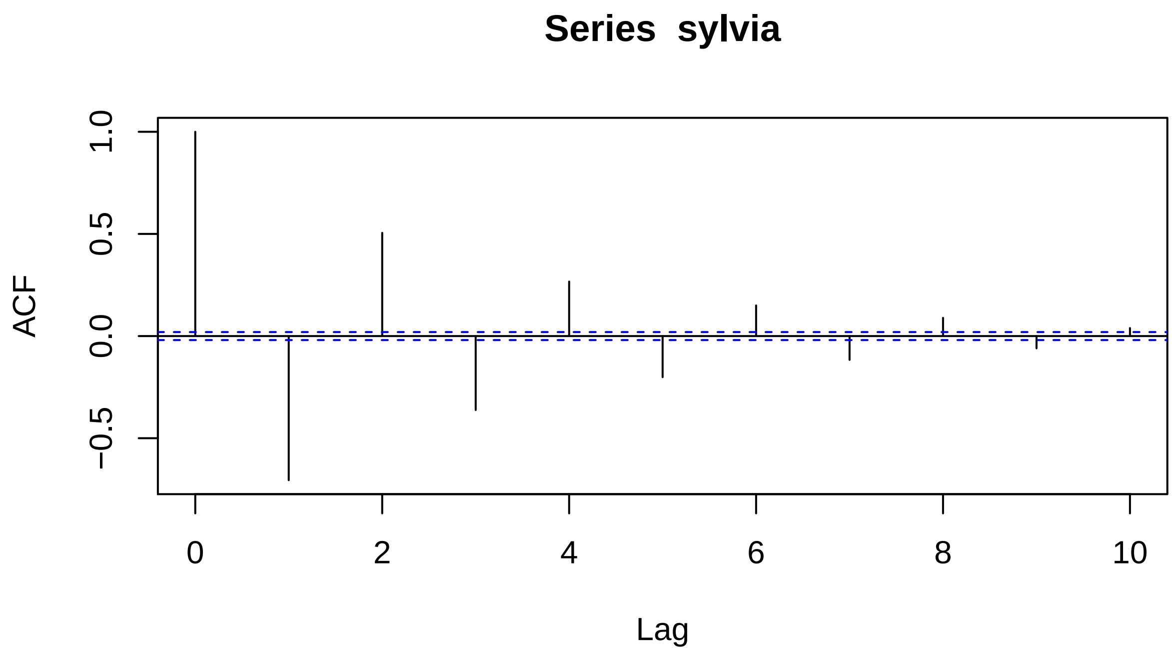



```
pacf(zhuoxiansheng,10)
```

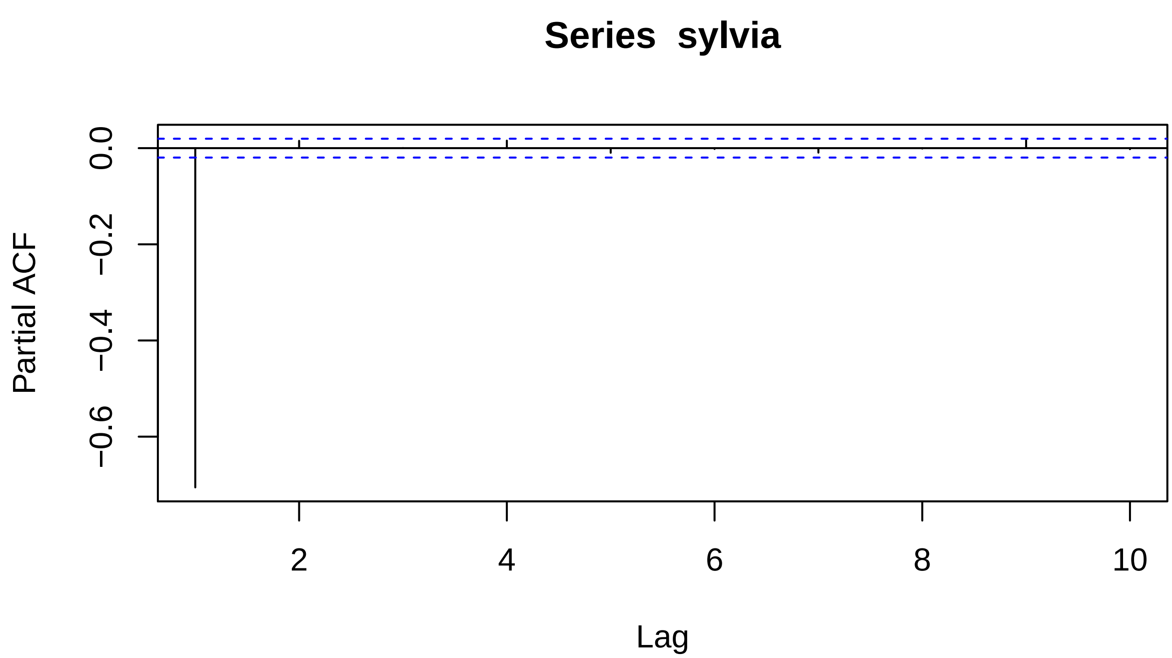


AR(1),a1>0

```
acf(sylvia,10)
```

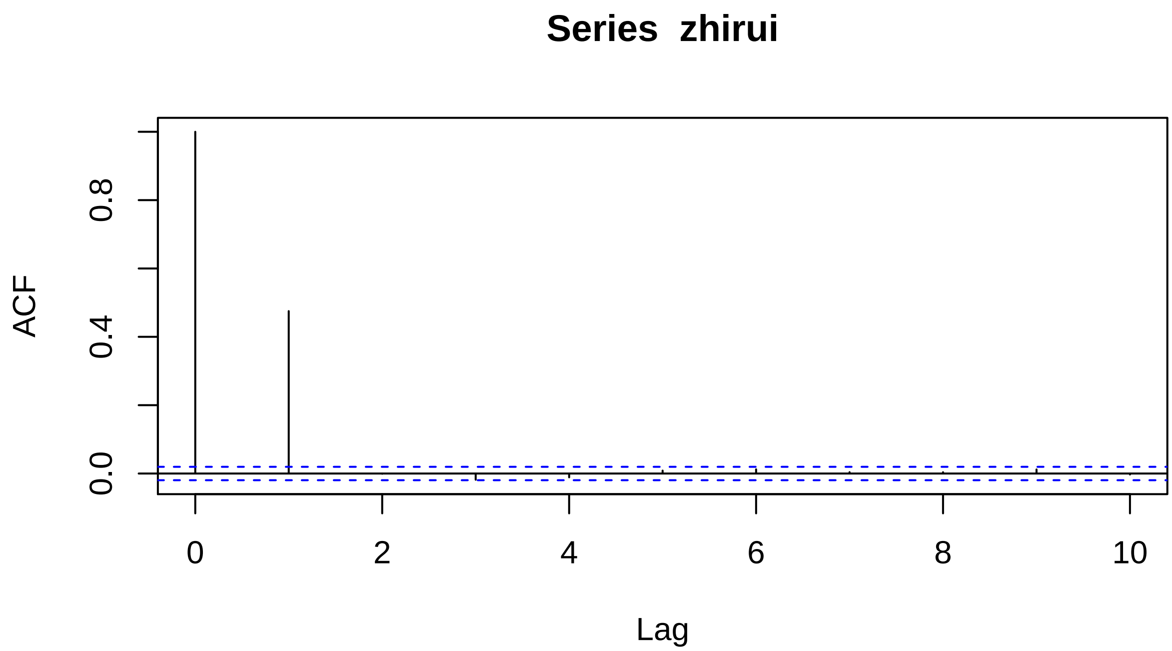



```
pacf(sylvia,10)
```



AR(1),a1<0

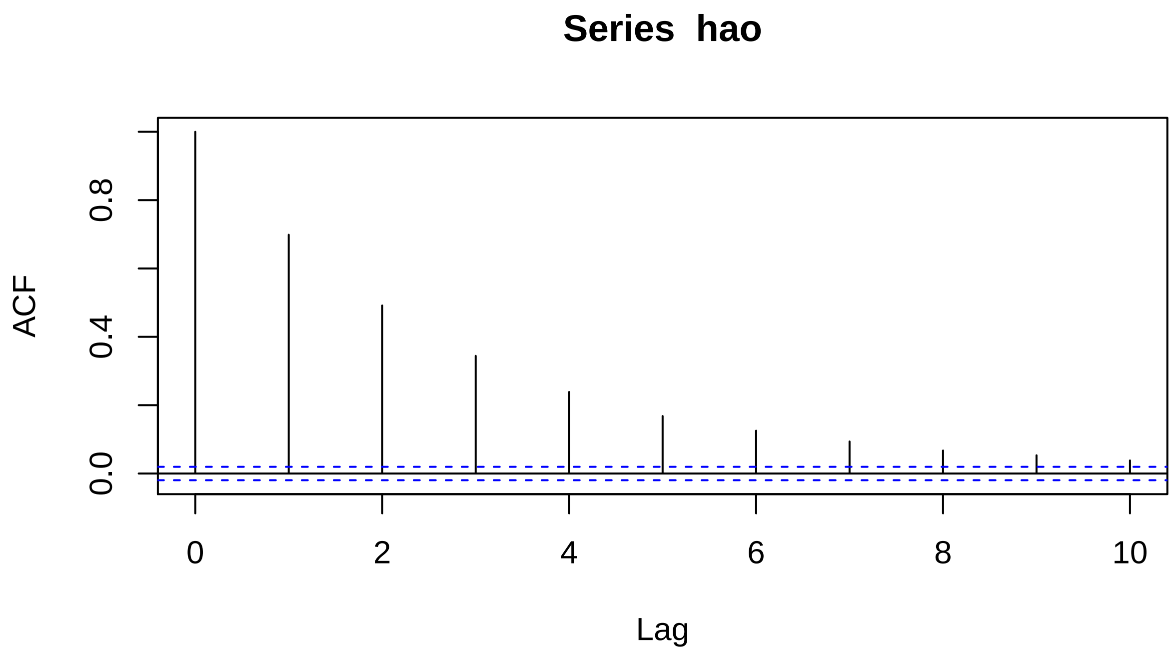
```
acf(zhirui,10)
```



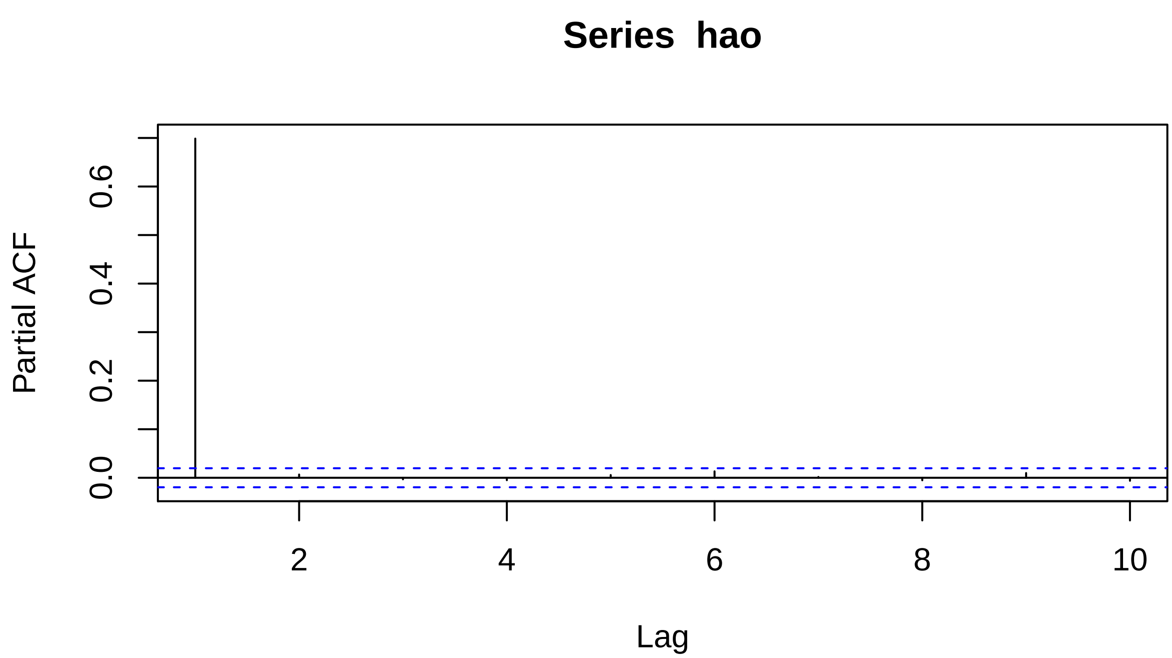

```r
pacf(zhirui,10)
```

MA(1),beta>0

```r
acf(hao,10)
```

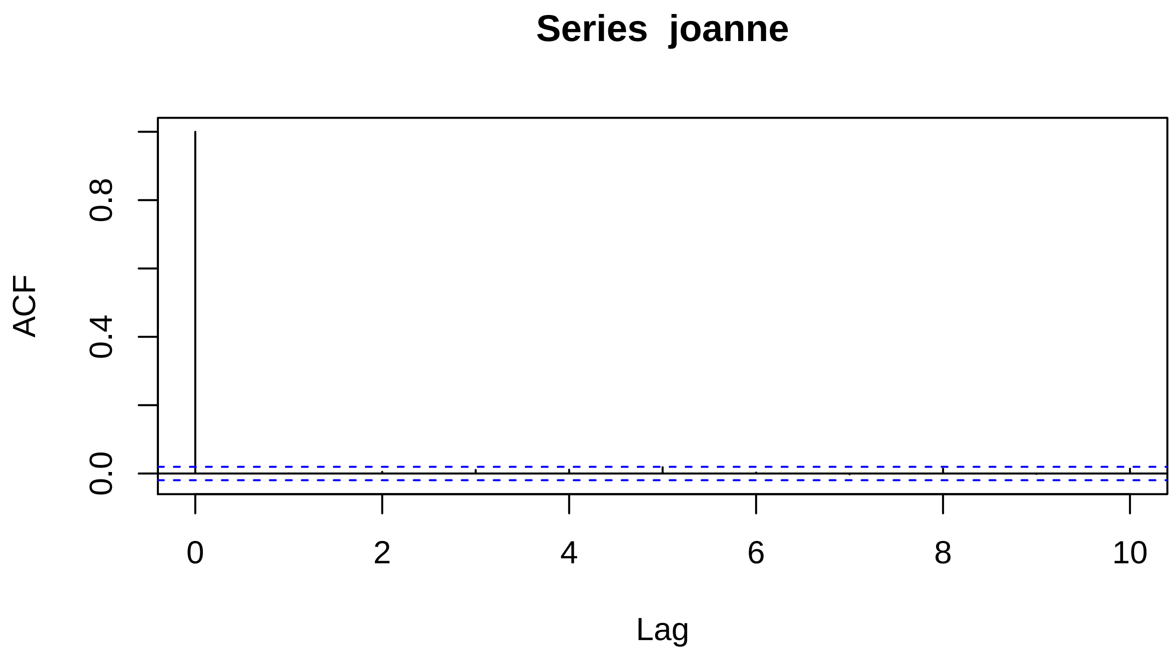



```r
pacf(hao,10)
```

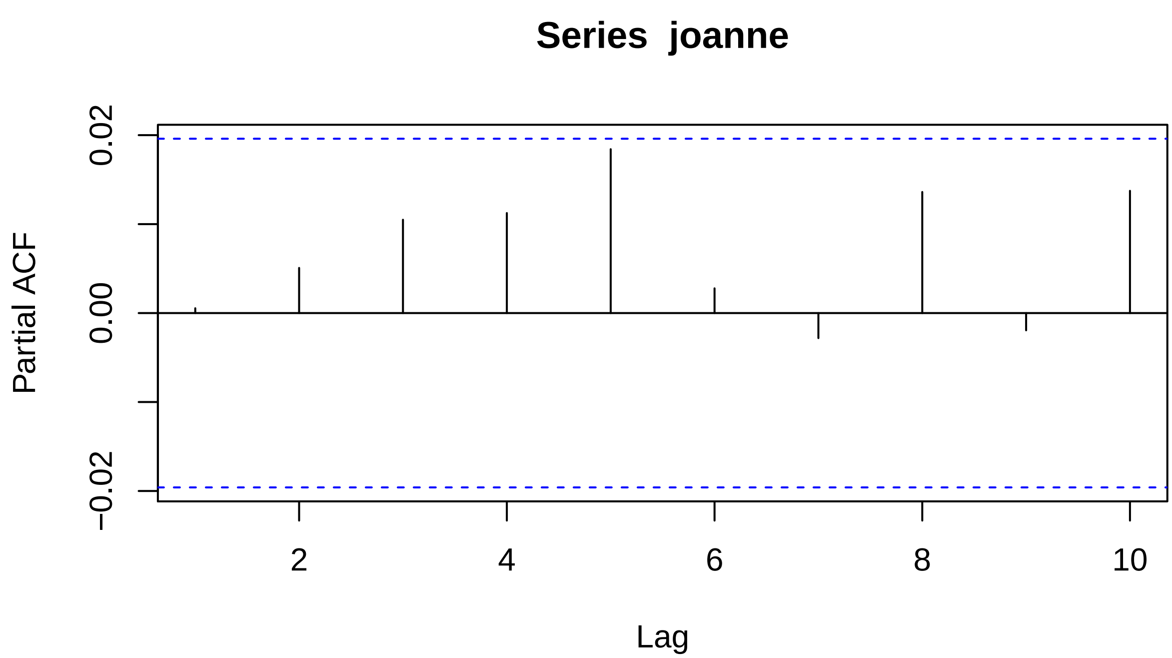


AR(1),a1>0

```r
acf(joanne,10)
```

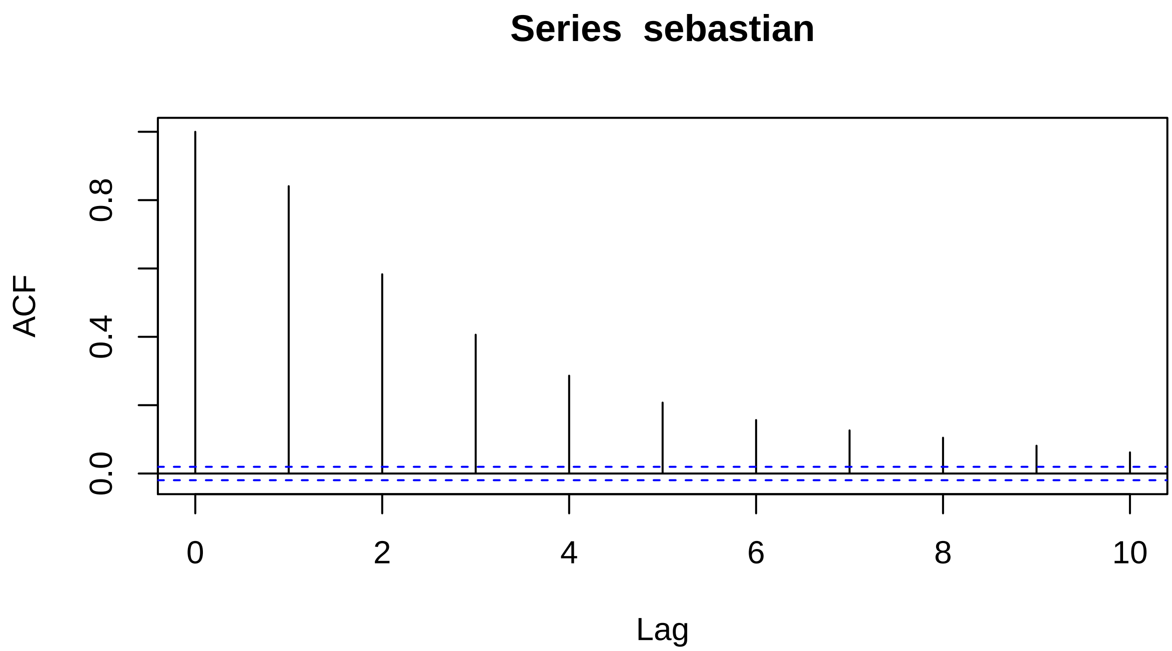



```
pacf(joanne,10)
```

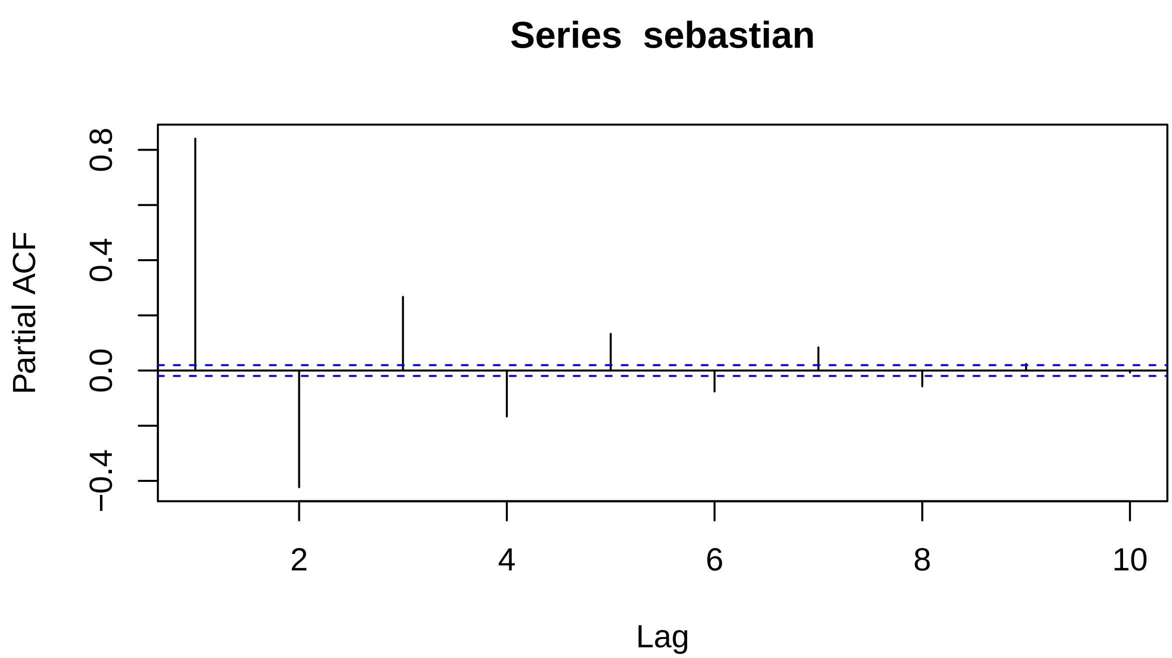


white noise

```
acf(sebastian,10)
```




```
pacf(sebastian,10)
```



ARMA(1,1),a1>0

Problem 4

This is the CBO longrun projection for medicare expenditure data made in June 2015. I am very interested in what time series model did CBO use to project medicare expenditure, so I choose this data.

```
suppressMessages(library(forecast))
library(tseries)
medicare <- read.csv("D:/Dropbox/16summer/Macroeconometrics/Problem Set1/medicare.csv")
```

test whether the data is stationary

```
adf.test(medicare$Net.Medicare)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  medicare$Net.Medicare
## Dickey-Fuller = -1.5371, Lag order = 4, p-value = 0.7644
## alternative hypothesis: stationary
```

Cannot reject null hypothesis. It is nonstationary.

Take first difference and test again

```
medi.d <- diff(medicare[,2])
adf.test(medi.d)
```

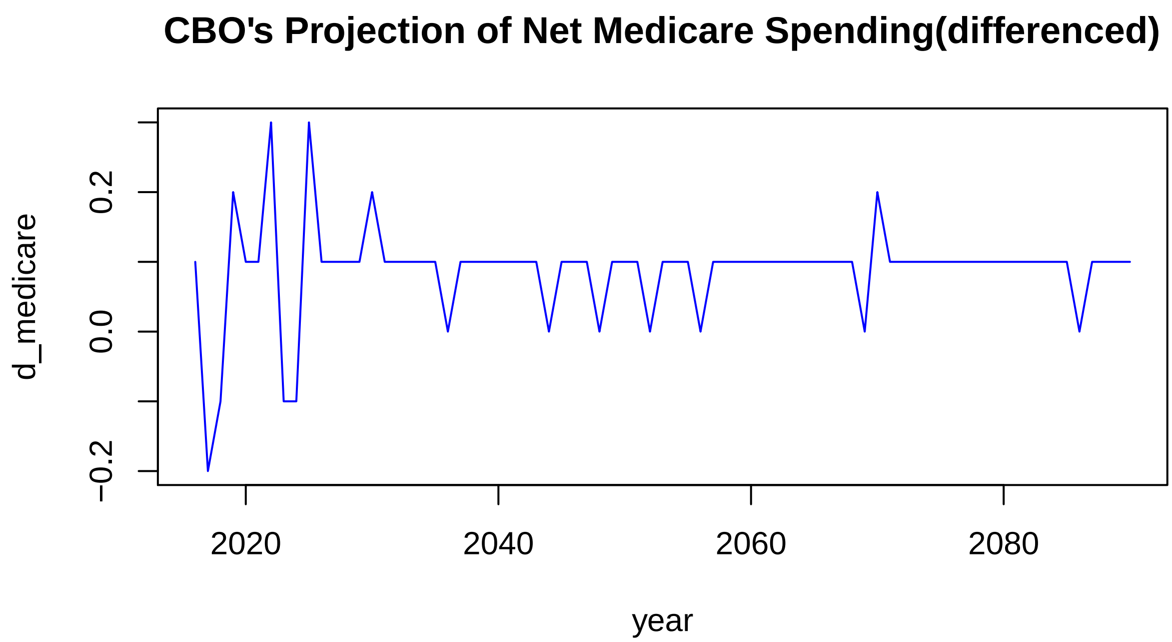
```
## Warning in adf.test(medi.d): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  medi.d
## Dickey-Fuller = -5.8565, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

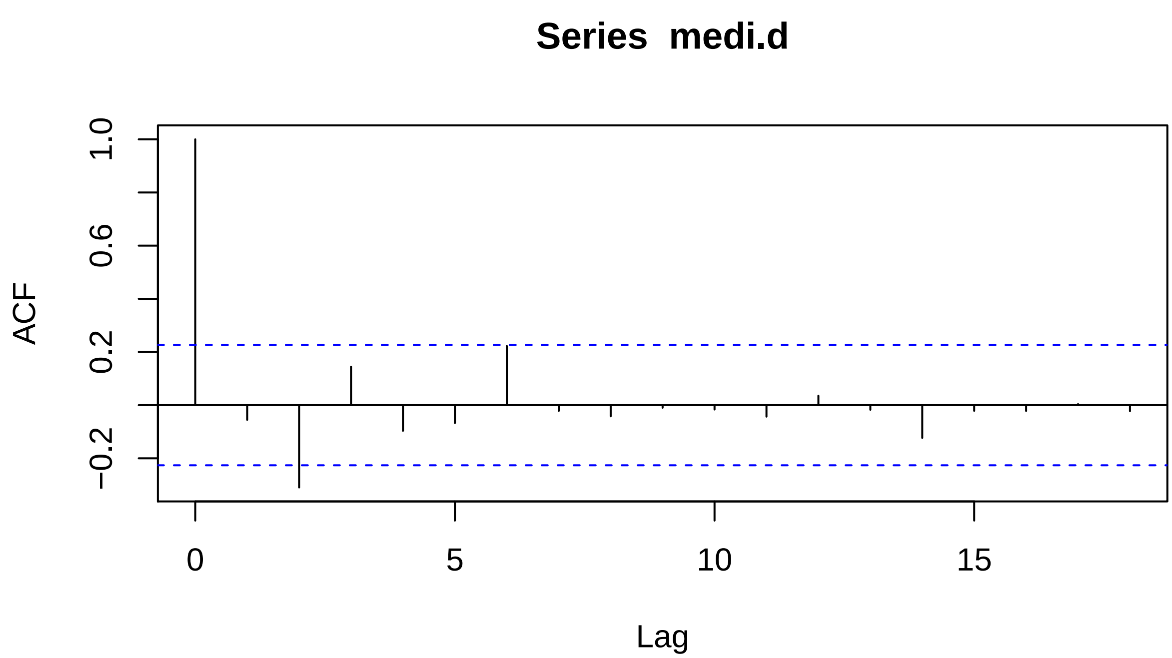
It is now stationary.

plot

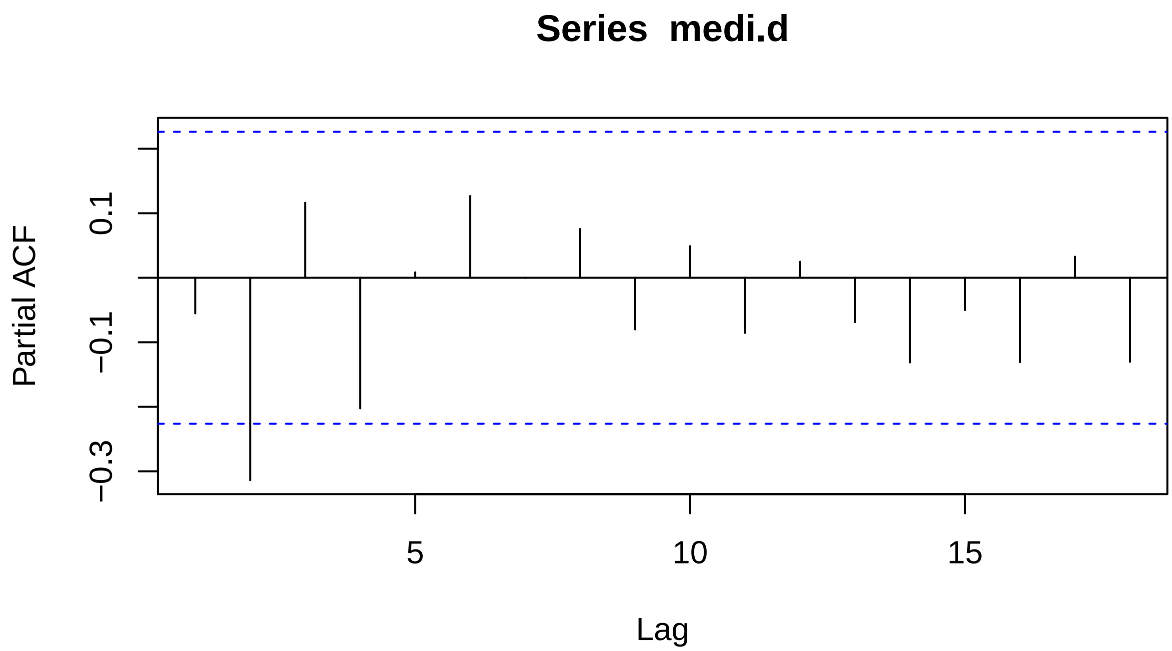
```
plot(2016:2090,medi.d,type = "l",xlab = "year",ylab = "d_medicare",
     main = "CBO's Projection of Net Medicare Spending(differenced)",col="blue")
```




```r
acf(medi.d)
```




```r
pacf(medi.d)
```



It's hard to tell by these graph what model it is. But it should be around 2nd-3rd lags.

Use auto.arima() from forecast package. It will choose the best model based on AIC.

```r
auto.arima(medi.d,ic="aic")
```

```
## Series: medi.d
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##           ma1      ma2  intercept
##       -0.0401  -0.4166     0.0897
## s.e.   0.1050   0.1038     0.0043
##
## sigma^2 estimated as 0.004496:  log likelihood=97.59
## AIC=-187.18   AICc=-186.61   BIC=-177.91
```

```r
res.medi <- arima(medi.d,order = c(0,0,2))
Box.test(res.medi$residuals,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  res.medi$residuals
## X-squared = 0.00061549, df = 1, p-value = 0.9802
```

MA(2) is the best model.

Problem 5

```r
library(forecast)
library(tseries)
library(car)
GDPDEF <- read.csv("D:/Dropbox/16summer/Macroeconometrics/Problem Set1/GDPDEF-2.csv")
UNRATE.2 <- read.csv("D:/Dropbox/16summer/Macroeconometrics/Problem Set1/UNRATE-2.csv")
```

Calculate the inflation and annualize it

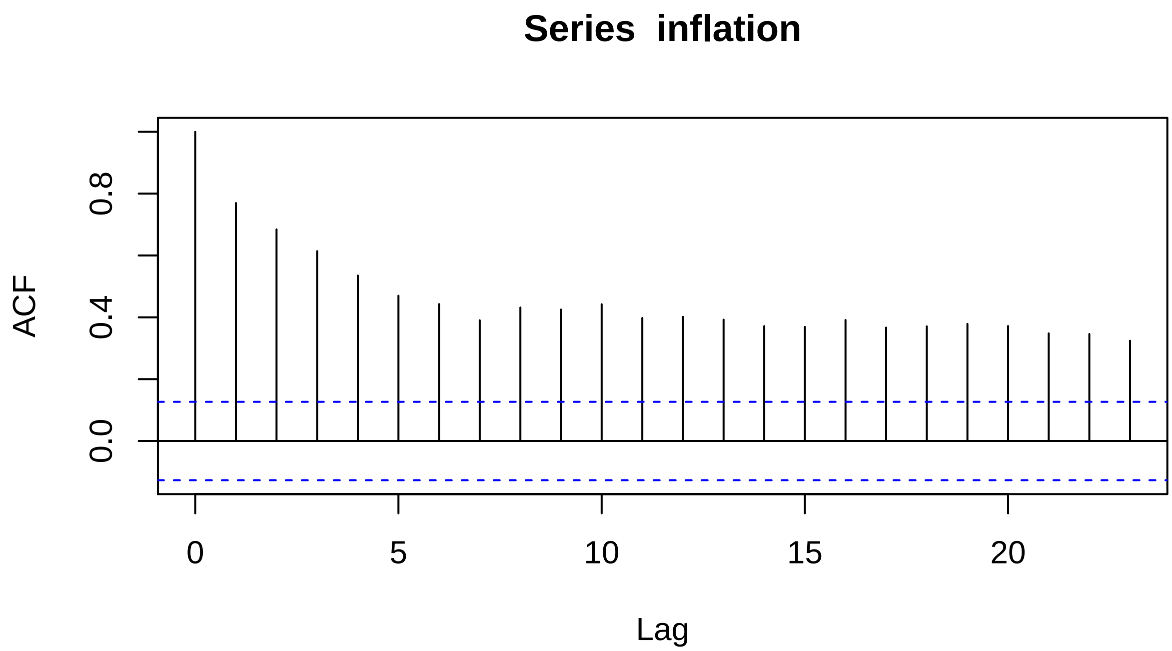
```r
inflation <- rep(NA,239)
for(i in 1:239){
  inflation[i] <- ((1+(GDPDEF[i+1,2]-GDPDEF[i,2])/GDPDEF[i,2])^4)-1
}
inflation.date <- cbind.data.frame(GDPDEF$DATE[-1],inflation)
colnames(inflation.date)[1] <- "DATE"
inflation.date$DATE<- as.Date(inflation.date$DATE,format="%m/%d/%Y")
```

(a)

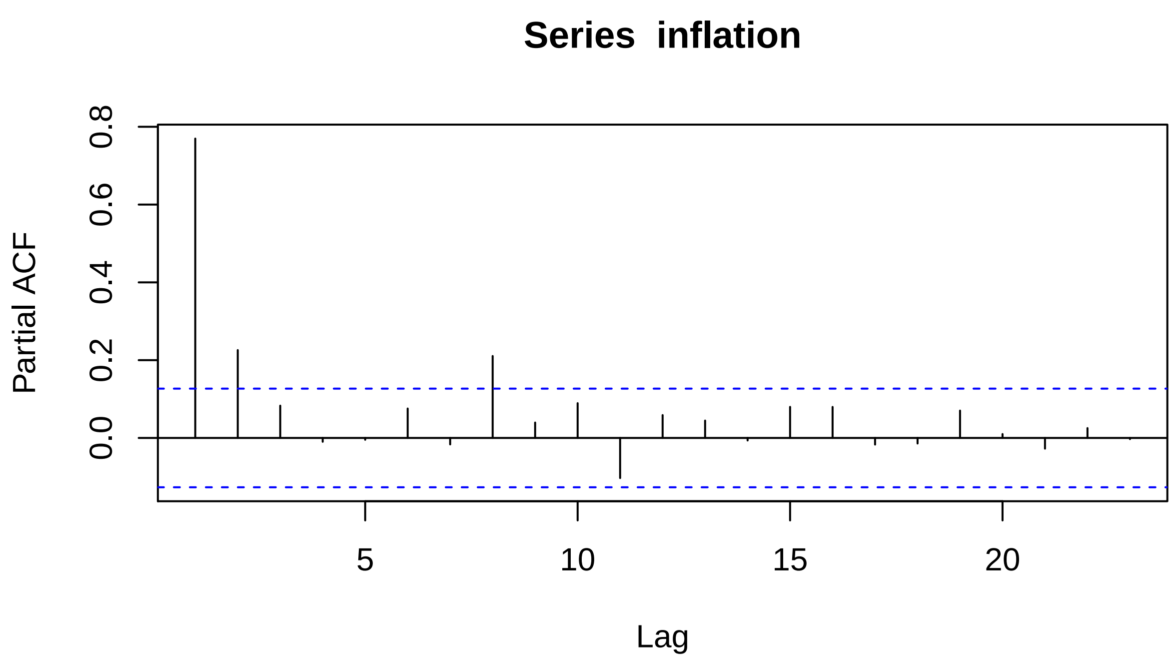
```r
adf.test(inflation)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  inflation
## Dickey-Fuller = -3.1665, Lag order = 6, p-value = 0.09419
## alternative hypothesis: stationary
```

```r
acf(inflation)
```




```r
pacf(inflation)
```




```r
ar(inflation,aic = TRUE)$aic
```

```
##          0          1          2          3          4          5
```

```
## 224.984615  12.486778   1.968600   2.315351   4.292468   6.287355
##          6          7          8          9         10         11
##   6.916481   8.850678   0.000000   1.625248   1.706980   1.154876
##         12         13         14         15         16         17
##   2.327071   3.849555   5.839550   6.310409   6.786814   8.717856
##         18         19         20         21         22         23
##  10.669504  11.484591  13.460376  15.279818  17.126058  19.124016
```

```r
ar(inflation,aic = TRUE)
```

```
##
## Call:
## ar(x = inflation, aic = TRUE)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
##  0.5829   0.1603   0.0919  -0.0152  -0.0627   0.0477  -0.1387   0.2107
##
## Order selected 8  sigma^2 estimated as  0.000271
```

```r
Box.test(ar(inflation)$resid,type="Ljung-Box")
```

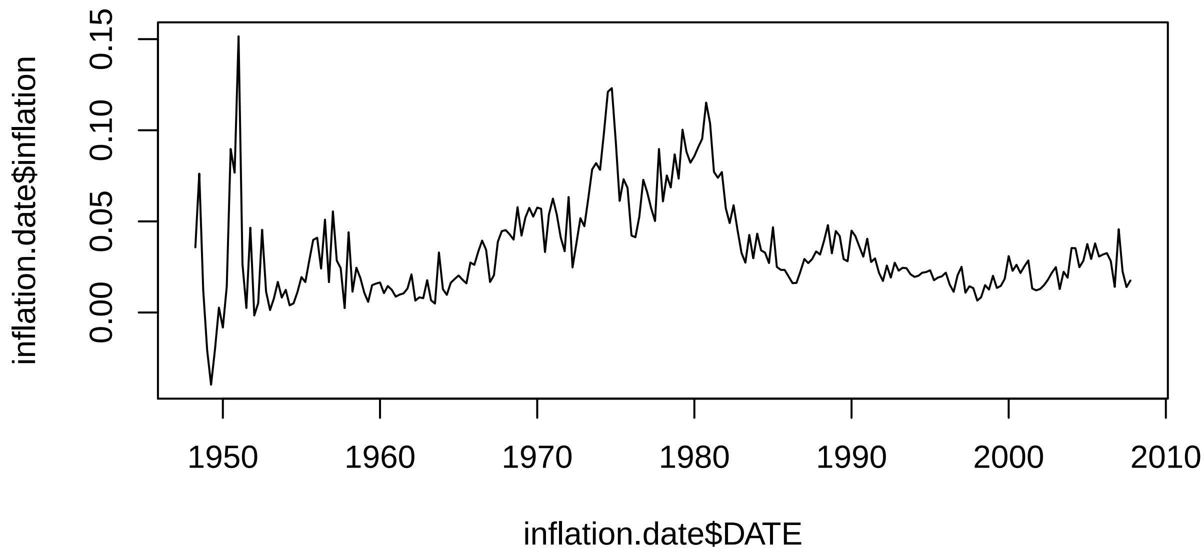
```
##
##  Box-Ljung test
##
## data:  ar(inflation)$resid
## X-squared = 0.53964, df = 1, p-value = 0.4626
```

By acf and pacf we can see that the 1st, 2nd and 8th length of pacf is significant, and acf is a geometric decay. So it might be a AR(2) or AR(8). Using the ar() funtion and set aic=TRUE, it will select the best AR model based on AIC. And the result is AR(8) is the best. It is a very long lag length. Maybe it captures something in the longrun business cycles, or it might be experienceing an overfitting problem, the significance of the 8th lag may be caused by a random error. The best way to trade off between bias-variance is to use a cross validation set, maybe using bootstrap is a good idea. It remains further studies.

(b)

Plot the inflation data

```r
plot(inflation.date$DATE,inflation.date$inflation,type="l")
```



We can see that the patterns before and after 1984 are kind of different. After 1984 the inflation is more "stable"" and seems like having a lower mean. It might be related to the new monetary policy during the Age of Reagan.So the structure change test is needed.

Construct chow test

```
ar.inflation1 <- ar(inflation[1:143],aic = FALSE,order.max = 8)
ar.inflation2 <- ar(inflation[144:239],aic = FALSE,order.max = 8)
SSRp <- sum(ar(inflation)$resid[-(1:8)]^2)
SSR1 <- sum(ar.inflation1$resid[-(1:8)]^2)
SSR2 <- sum(ar.inflation2$resid[-(1:8)]^2)
chow <- ((SSRp-SSR1-SSR2)/(SSR1+SSR2))*((239-2*8)/8)
print(chow)
```

```
## [1] 0.7958668
```

```
qf(0.95,8,(239-8*2))
```

```
## [1] 1.980087
```

Unfortunately the test is not significant. There is no actual structure change here.

(c)

Regress the Phillips curve with lagged unemployment rate symmetrically. We think the model won't be bigger than 5th lag because otherwise the model will be too big. So we truncate the data for 5 periods so that every model will have the same obs number and then we can compare the AIC correctly.

```r
unrate <- UNRATE.2[,2]
L.unrate <- c(unrate[-240])
L2.unrate <- c(NA,L.unrate[-239])
L3.unrate <- c(NA,L2.unrate[-239])
L4.unrate <- c(NA,L3.unrate[-239])
L5.unrate <- c(NA,L4.unrate[-239])
arima(inflation[-(1:4)],order = c(1,0,0),xreg =cbind(L.unrate)[-c(1:4),])$aic
```

```
## [1] -1251.171
```

```r
arima(inflation[-(2:4)],order = c(2,0,0),xreg =cbind(L.unrate,L2.unrate)[-c(2:4),])$aic
```

```
## [1] -1266.001
```

```r
arima(inflation[-(3:4)],order = c(3,0,0),xreg =cbind(L.unrate,L2.unrate,L3.unrate)[-c(3:4),])$aic
```

```
## [1] -1269.46
```

```r
arima(inflation[-4],order = c(4,0,0),xreg =cbind(L.unrate,L2.unrate,L3.unrate,L4.unrate)[-4,])$aic
```

```
## [1] -1266.034
```

```r
arima(inflation,order = c(5,0,0),xreg =cbind(L.unrate,L2.unrate,L3.unrate,L4.unrate,L5.unrate))$aic
```

```
## [1] -1268.086
```

The best model is AR(3) with 3 lags of unemployment rate.

```r
arima(inflation,order = c(3,0,0),xreg =cbind(L.unrate,L2.unrate,L3.unrate))
```

```
##
## Call:
## arima(x = inflation, order = c(3, 0, 0), xreg = cbind(L.unrate, L2.unrate, L3.unrate))
##
## Coefficients:
##          ar1     ar2     ar3  intercept  L.unrate  L2.unrate  L3.unrate
##       0.5486  0.1768  0.1414     0.0324   -0.0010    -0.0062     0.0071
## s.e.  0.0648  0.0745  0.0665     0.0148    0.0036     0.0052     0.0036
##
## sigma^2 estimated as 0.0002466:  log likelihood = 647.57,  aic = -1279.14
```

```r
Box.test(arima(inflation,order = c(3,0,0),xreg =cbind(L.unrate,L2.unrate,L3.unrate))$resid,
         type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  arima(inflation, order = c(3, 0, 0), xreg = cbind(L.unrate, L2.unrate,     L3.unrate))$resid
## X-squared = 0.026935, df = 1, p-value = 0.8696
```

(d)

Test in each subsample whether the sum of the unemployment rate equals to zero

```
philps1 <- arima(inflation[c(1:167)],order = c(3,0,0),
                 xreg =cbind(L.unrate,L2.unrate,L3.unrate)[c(1:167),])
linearHypothesis(philps1,c(0,0,0,0,1,1,1))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L.unrate  + L2.unrate  + L3.unrate = 0
##
## Model 1: restricted model
## Model 2: philps1
##
##   Df  Chisq Pr(>Chisq)
## 1
## 2  1 0.0174     0.8951
```

```
philps2 <- arima(inflation[c(168:239)],order = c(3,0,0),
                 xreg =cbind(L.unrate,L2.unrate,L3.unrate)[c(168:239),])
linearHypothesis(philps2,c(0,0,0,0,1,1,1))
```

```
## Linear hypothesis test
##
## Hypothesis:
## L.unrate  + L2.unrate  + L3.unrate = 0
##
## Model 1: restricted model
## Model 2: philps2
##
##   Df  Chisq Pr(>Chisq)
## 1
## 2  1 0.3373     0.5614
```

Yes. Both equals to zero. This means that the long run propensity of unemployment rate equals to zero. In the long run, the inflation will not depend on the unemployment rate any more. In both sample, the slopes of the Phillips curve become zero.