econ144hw3

Sijia Hua

5/6/2019

```r
library(timeSeries)
```

```
## Loading required package: timeDate
```

```r
library(marima)
library(strucchange)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following object is masked from 'package:timeSeries':
##
##     time<-

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

## Loading required package: sandwich
```

```r
library(seasonal)
```

```
##
## Attaching package: 'seasonal'

## The following objects are masked from 'package:timeSeries':
##
##     outlier, series
```

```r
library(dynlm)
```

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

```r
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

##

## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.

##
## Attaching package: 'gdata'

## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
require(graphics)
library("readxl")
```

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

```
library('xts')
```

```
##
## Attaching package: 'xts'

## The following objects are masked from 'package:gdata':
##
##     first, last
```

```
library('forecast');
```

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

```
library('fma')
library('expsmooth')
library('lmtest')
library('tseries')
library('Quandl')
library('fpp');
library('urca')
library(Hmisc)
```

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

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
setwd("/Users/Renaissance/Desktop/econ144/econ144hw3")
```

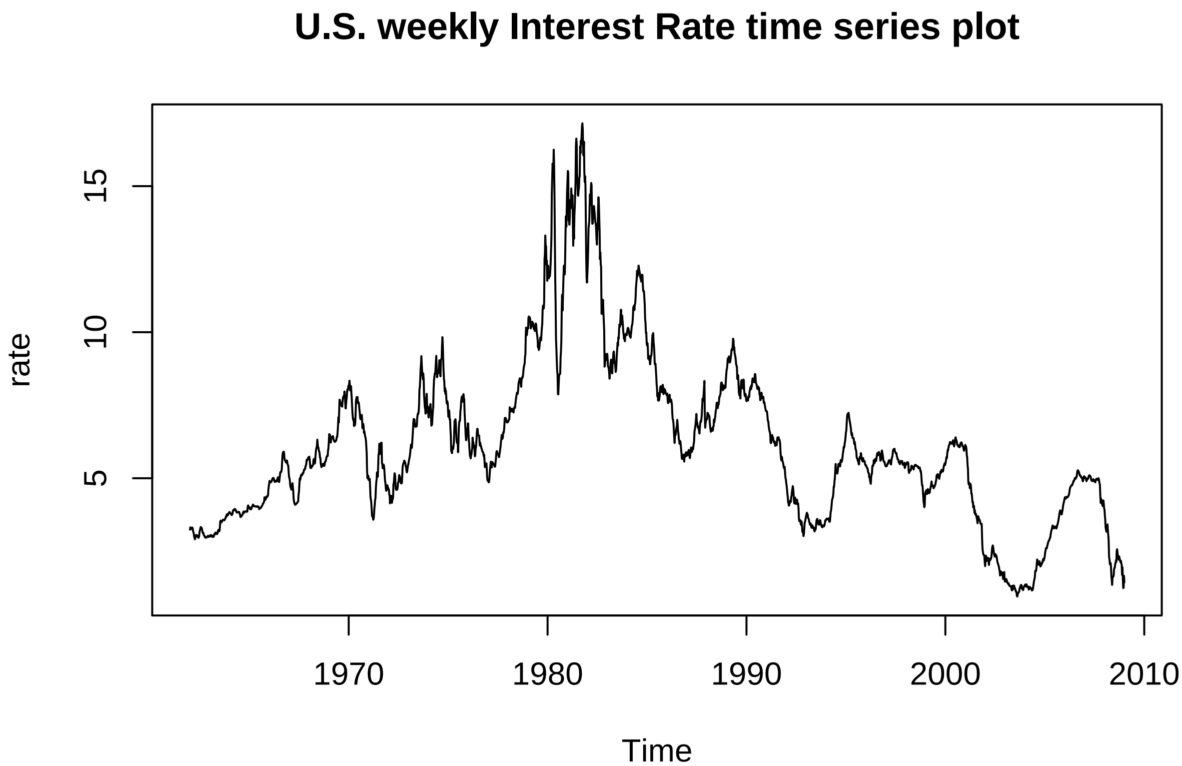
first question

```
hw31 <- read.table("w-gs1yr.txt")
inte <- interpNA(hw31$V4,method="linear")
```

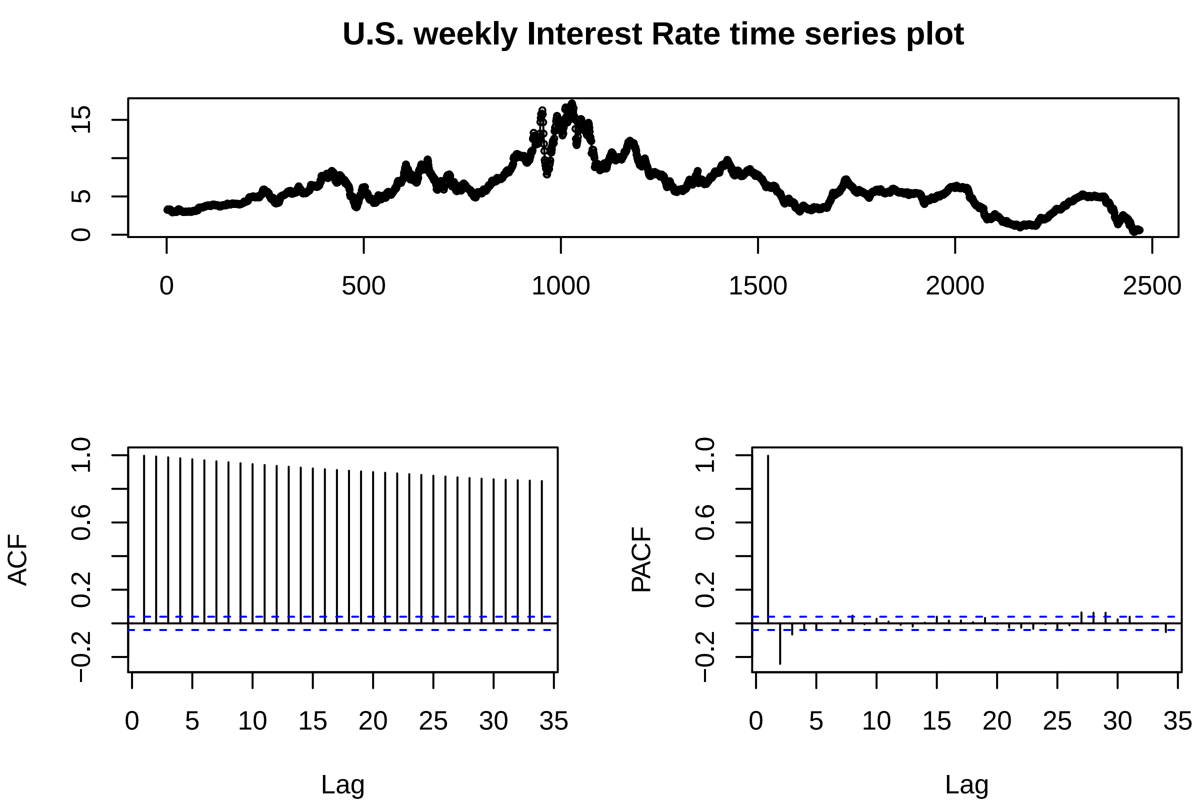
```
## Warning in xy.coords(x, y, setLab = FALSE): NAs introduced by coercion
```

```
inte2 <- as.numeric(inte)
## a
```

```r
t1<-ts(inte,start=1962,2009,freq = 52)
plot(t1,ylab="rate",main="U.S. weekly Interest Rate time series plot")
```




```r
tsdisplay(inte2,main = "U.S. weekly Interest Rate time series plot")
```



```r
# ACF of this dataset is graudally decreasing in a very slow trend, in PACF graph there is only first t
## b
# try ar2
t12<-as.numeric(t1[2:2445])
ar2_q1 <- arma(t12,order=c(2,0))
summary(ar2_q1)
```

```
##
## Call:
## arma(x = t12, order = c(2, 0))
##
## Model:
## ARMA(2,0)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.5675255 -0.0590173  0.0008931  0.0625970  1.4208073
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1        1.342579    0.018991   70.696   <2e-16 ***
## ar2       -0.345195    0.018996  -18.172   <2e-16 ***
## intercept  0.015525    0.008424    1.843   0.0653 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 0.03178,  Conditional Sum-of-Squares = 77.57,  AIC = -1487.67
```

```r
# try ar1 +ma2
ar1ma2_q1 <- arma(t12,order=c(1,2))
summary(ar1ma2_q1)
```

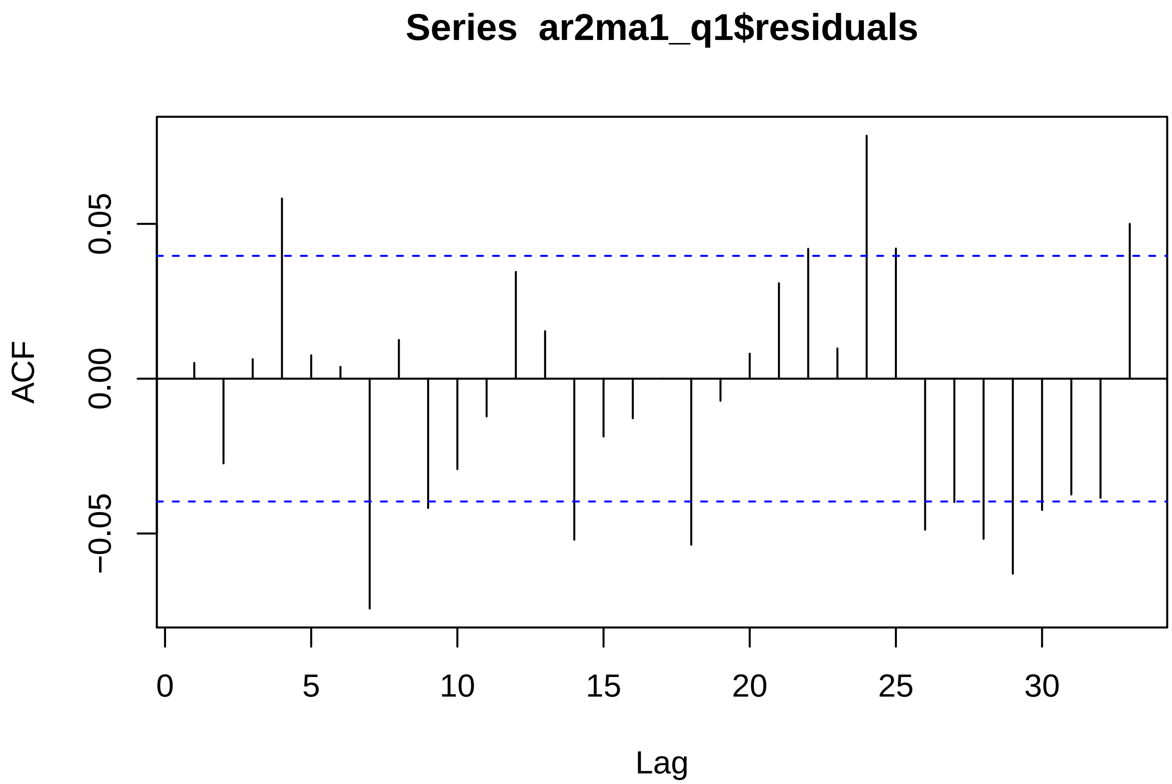
```
##
## Call:
## arma(x = t12, order = c(1, 2))
##
## Model:
## ARMA(1,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56965 -0.05907  0.00131  0.06342  1.42291
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1        0.996452    0.001797  554.597  < 2e-16 ***
## ma1        0.323103    0.020092   16.081  < 2e-16 ***
## ma2        0.122156    0.018829    6.488 8.72e-11 ***
## intercept  0.020984    0.012216    1.718   0.0859 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
```

```
## sigma^2 estimated as 0.03191,  Conditional Sum-of-Squares = 77.89,  AIC = -1475.37
```

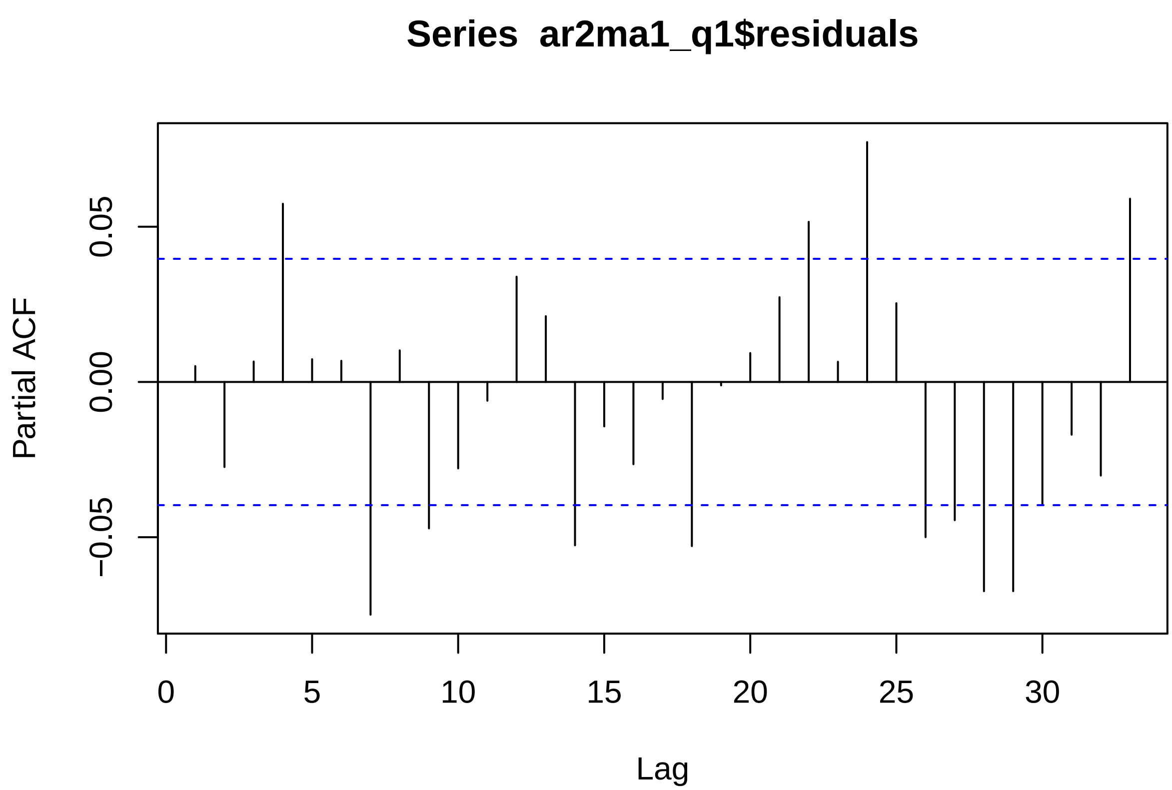
```r
# ar2 +ma1
ar2ma1_q1 <- arma(t12,order=c(2,1))
summary(ar2ma1_q1)
```

```
##
## Call:
## arma(x = t12, order = c(2, 1))
##
## Model:
## ARMA(2,1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.5995074 -0.0585295  0.0005104  0.0612440  1.4332955
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1        1.536201    0.053559   28.683  < 2e-16 ***
## ar2       -0.538464    0.053460  -10.072  < 2e-16 ***
## ma1       -0.223032    0.062927   -3.544 0.000394 ***
## intercept  0.013537    0.006563    2.063 0.039158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 0.03163,  Conditional Sum-of-Squares = 77.21,  AIC = -1496.98
```

```r
# I prefer arma(2,1) model. Because as it is shown in the sumamry, all coefficients are of significant.
## c
Acf(ar2ma1_q1$residuals)
```

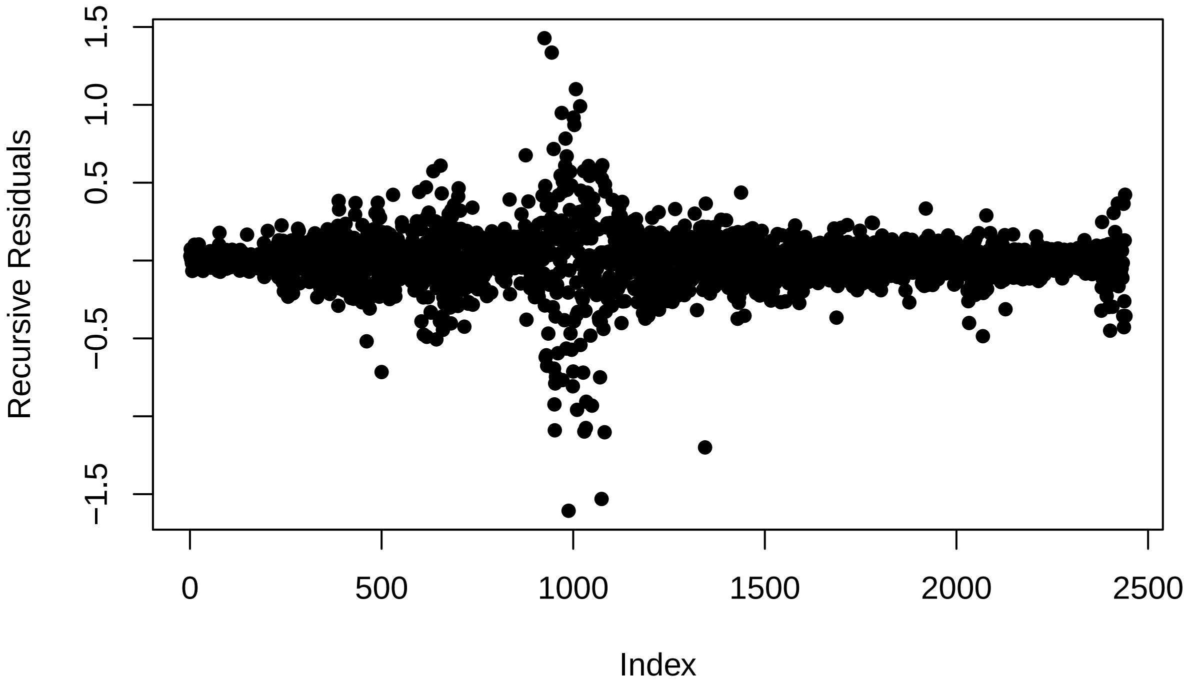



```r
Pacf(ar2ma1_q1$residuals)
```

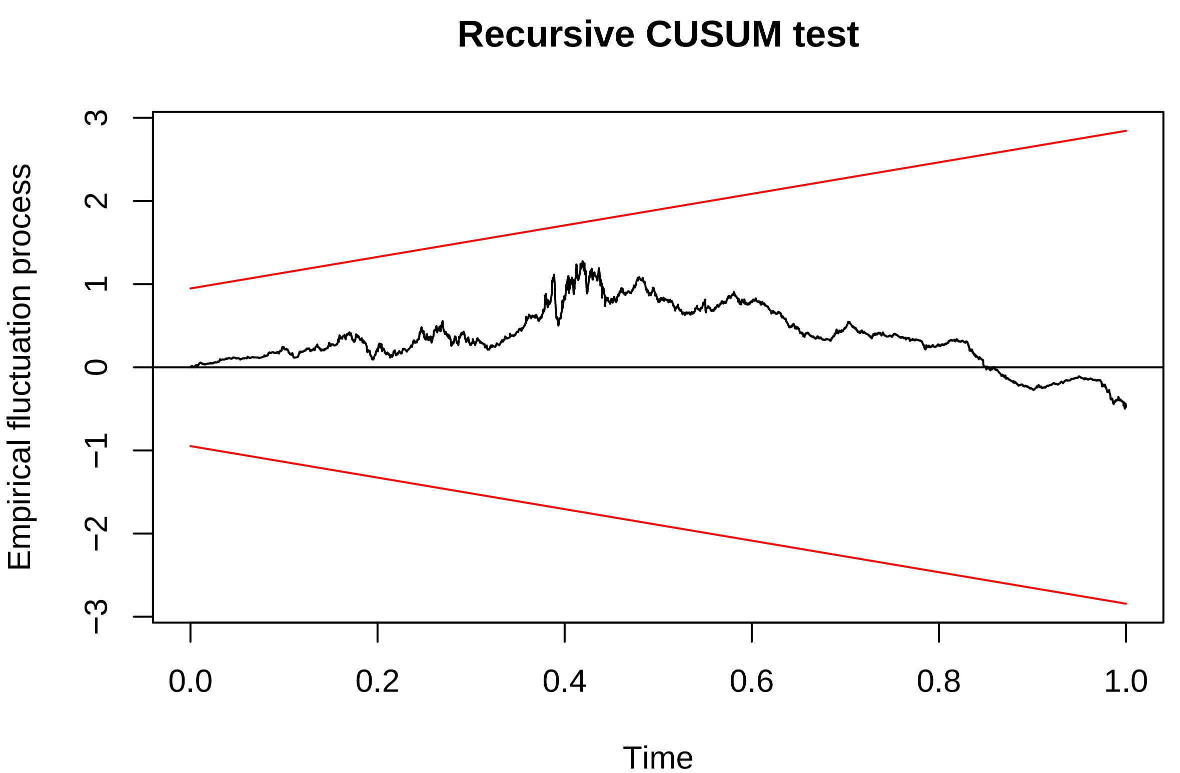



```r
# In acf and pacf graph, there are some residuals of significance. Hence arma(2,1) may not be a good fi
## d
```

```r
rec_q1=recresid(ar2ma1_q1$residuals~1)
plot(rec_q1, pch=16,ylab="Recursive Residuals")
```



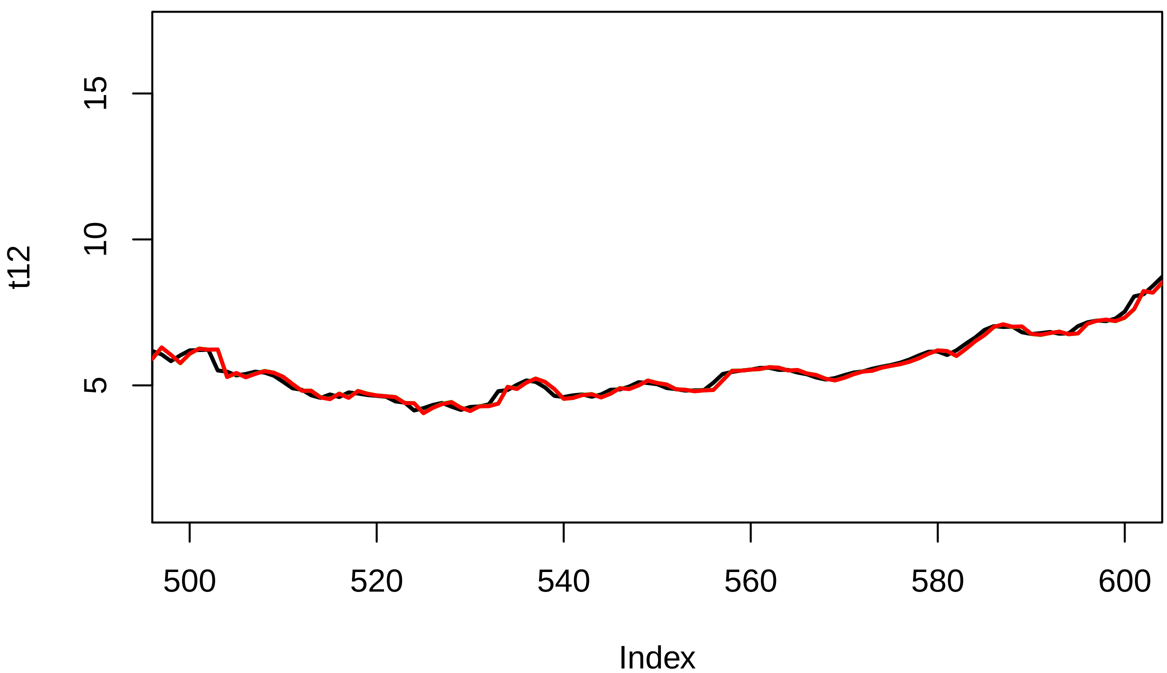

```r
# residuals are bounced around 0 randomly. But there are large residuals around index= 1000.
## e
plot(efp(ar2ma1_q1$residuals~1, type = "Rec-CUSUM"))
```



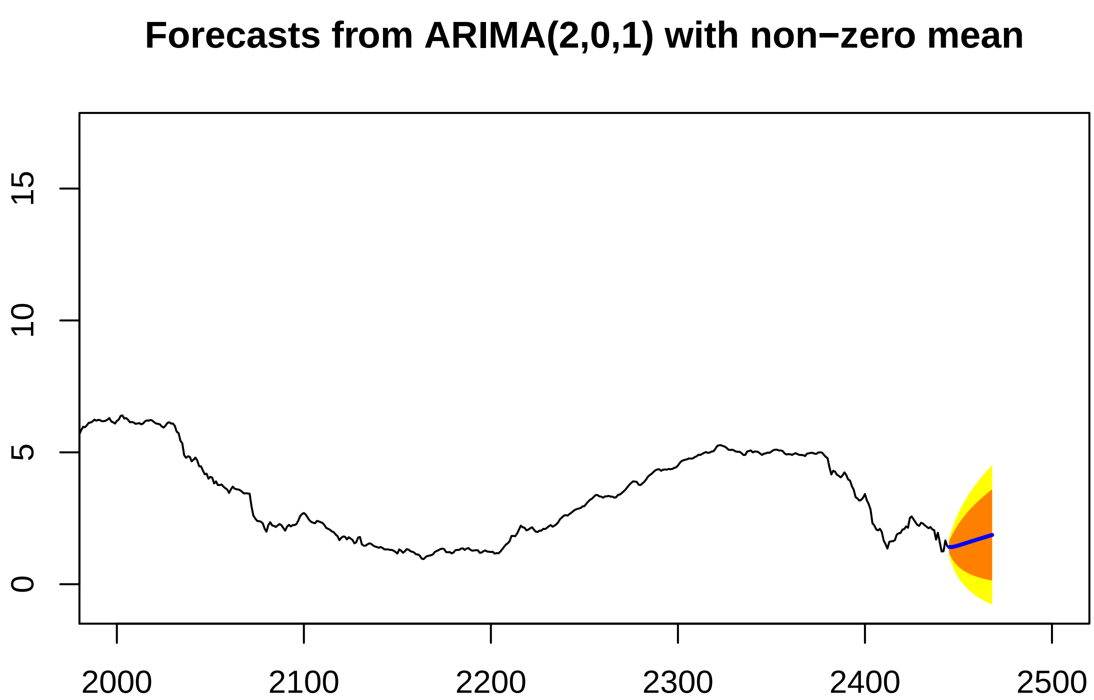
```
# this series passes the recursive cusum test.
## f
auto_p1 <- auto.arima(t12)
summary(auto_p1)
```

```
## Series: t12
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.6265  -0.3053  -0.0517
## s.e.  0.0650   0.0684   0.0301
##
## sigma^2 estimated as 0.03168:  log likelihood=751.7
## AIC=-1495.4   AICc=-1495.38   BIC=-1472.19
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE
## Training set -0.0004550749 0.1778408 0.1051573 -0.03434224 1.758331
##                  MASE           ACF1
## Training set 0.937794 -0.0004524371
```

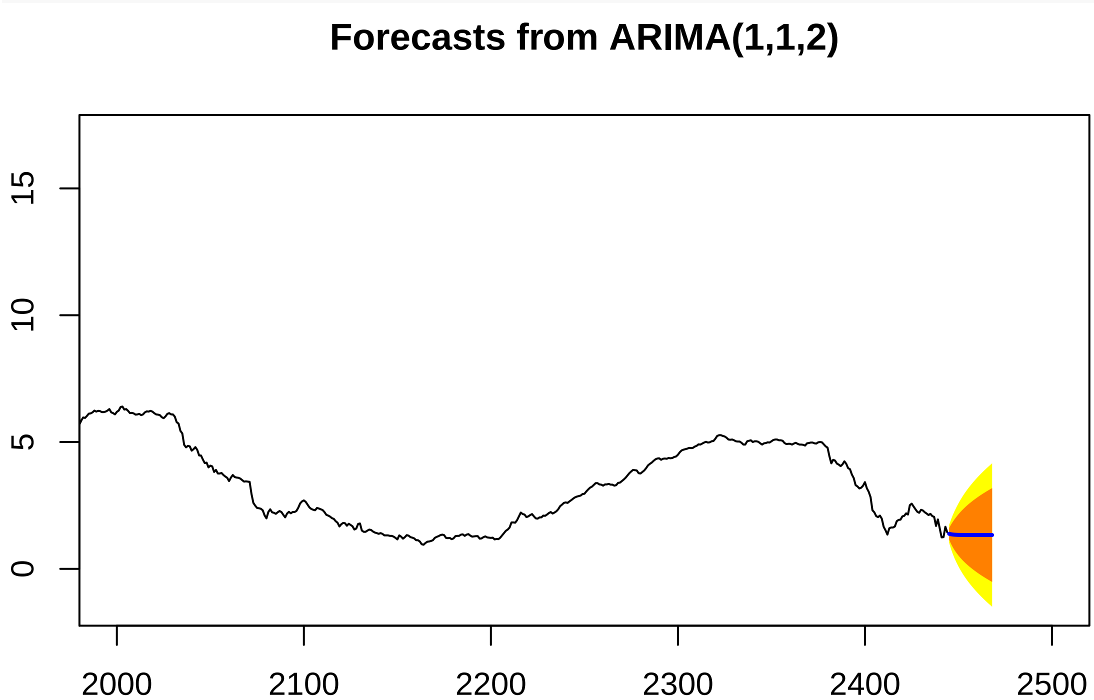
```
# the best fit in r is arima(1,1,2): that is ar order = 1, level of differencing = 1, ma order = 2
plot(t12,type='l',xlim=c(500,600),lwd=2)
lines(ar2ma1_q1$fitted.values,type='l',col='green',lwd=2)
lines(auto_p1$fitted,type='l',col='red',lwd=2)
```




```
# these two simulations coincide each other.
## g
f_q1 = Arima(t12,order=c(2,0,1))
plot(forecast(f_q1,h=24),shadecols="oldstyle",xlim=c(2000,2500))
```



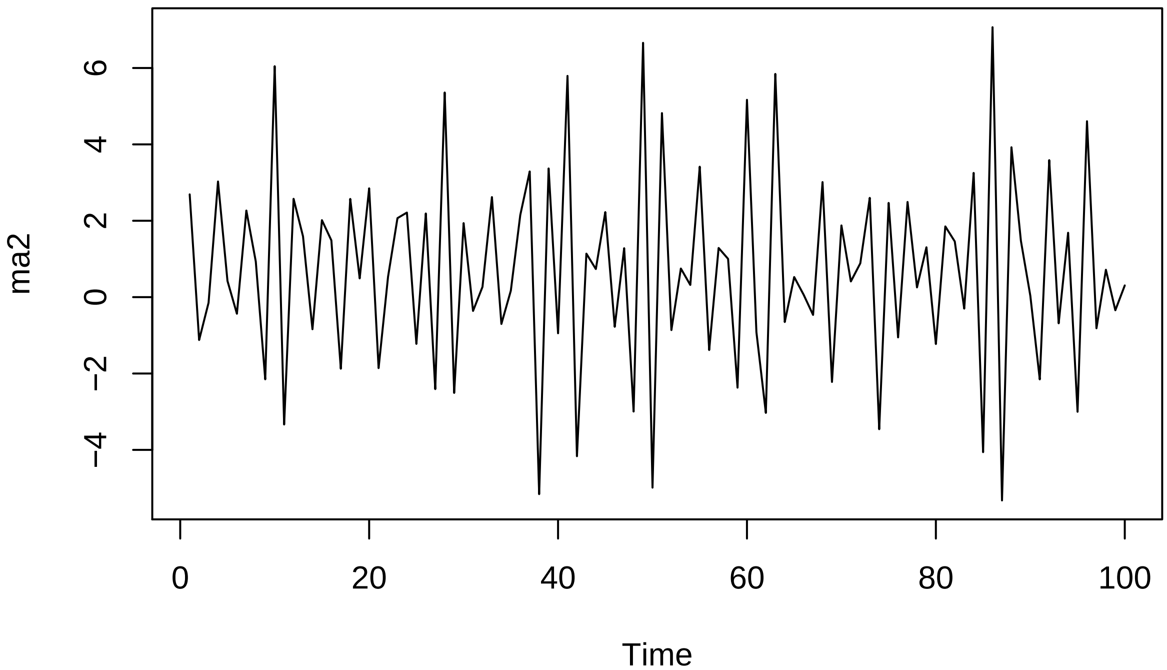

```
plot(forecast(auto_p1,h=24),shadecols="oldstyle",xlim=c(2000,2500))
```



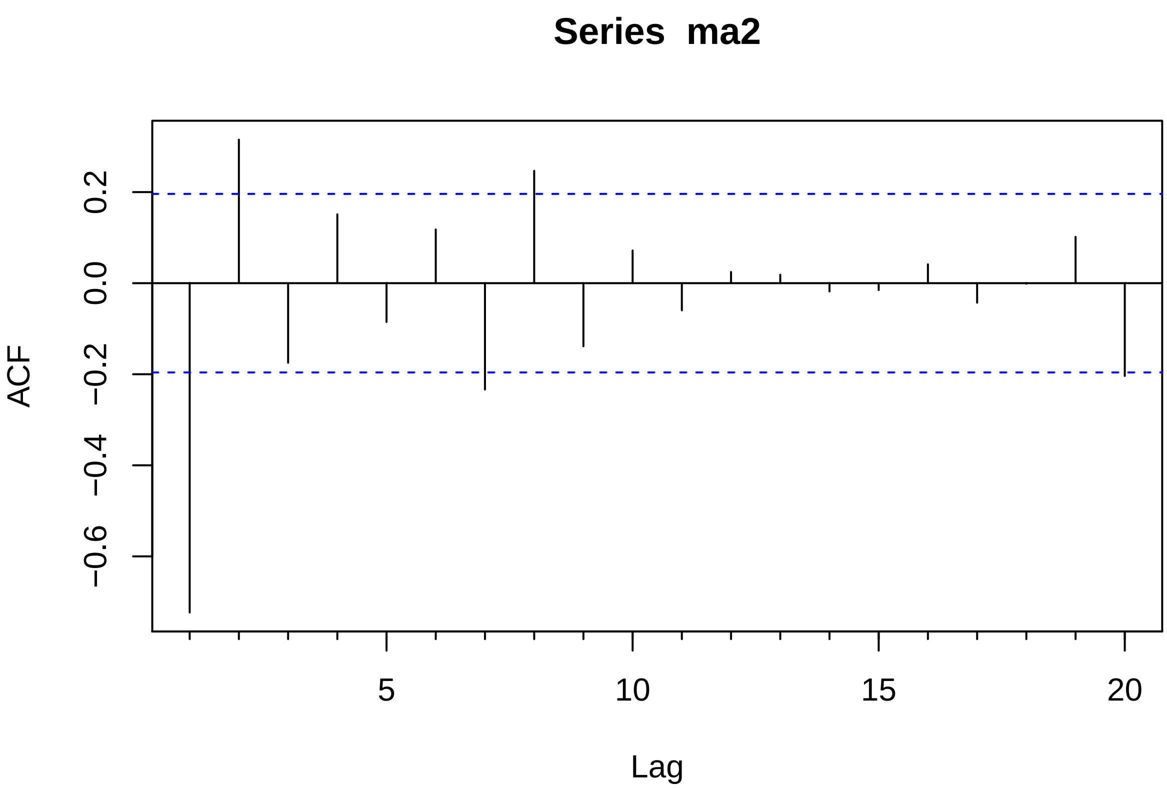

```
# arima(2,0,1) gives a upward trend approxiamtion, while arima(1,1,2) generated by autoarima gives a re
```

6.5 from book a

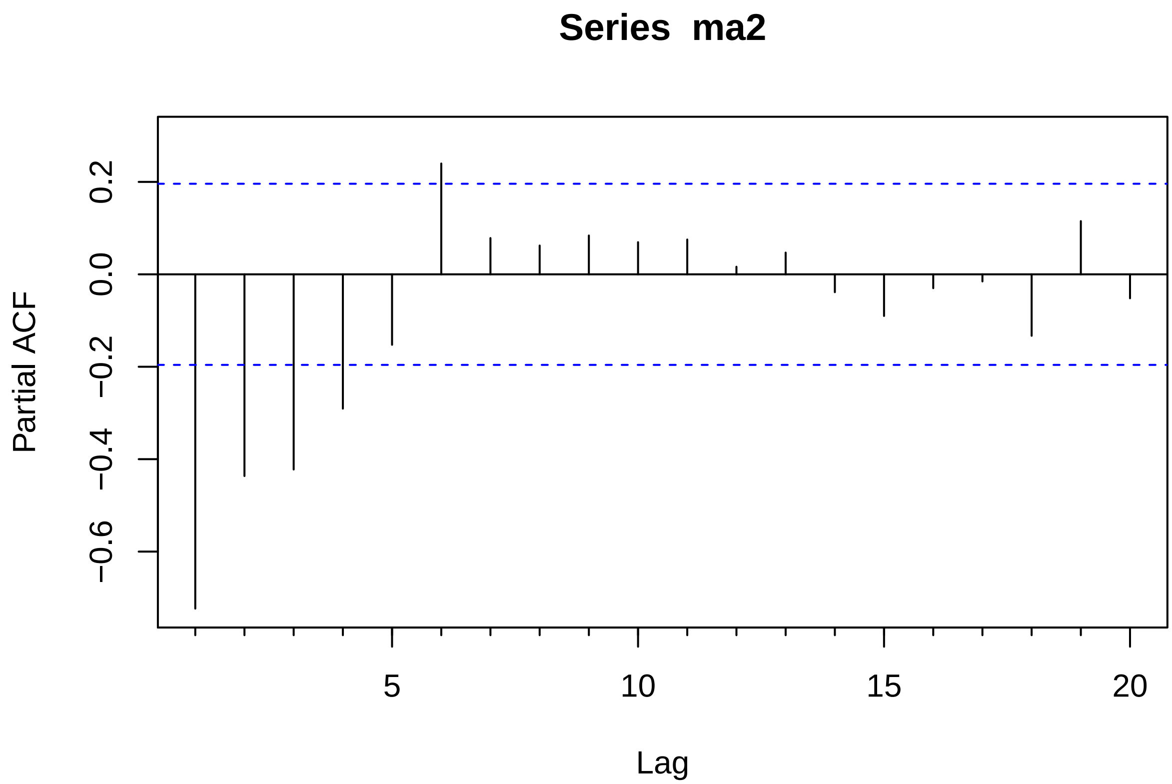
```
# generate data set in 6.4
ma2 <- arima.sim(model=list(ma=c(-2,1.35)),n=100)+0.7
plot(ma2)
```




```
Acf(ma2)
```




```
Pacf(ma2)
```

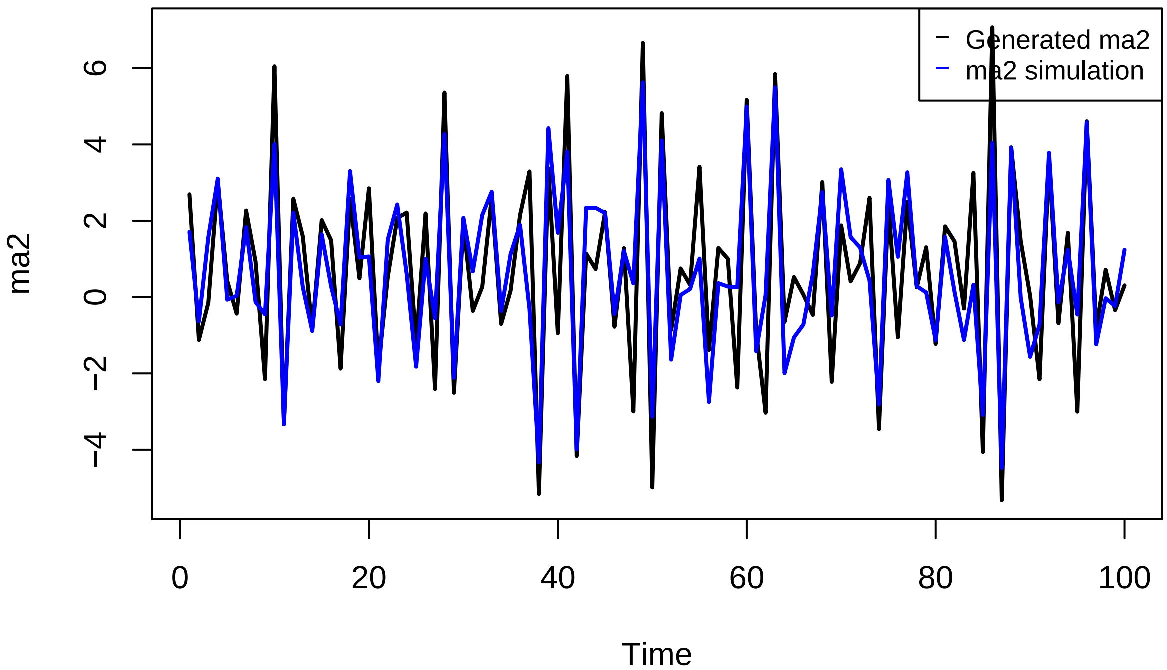



```r
## a
ma2_summary=arma(ma2,order=c(0,2))
summary(ma2_summary)
```

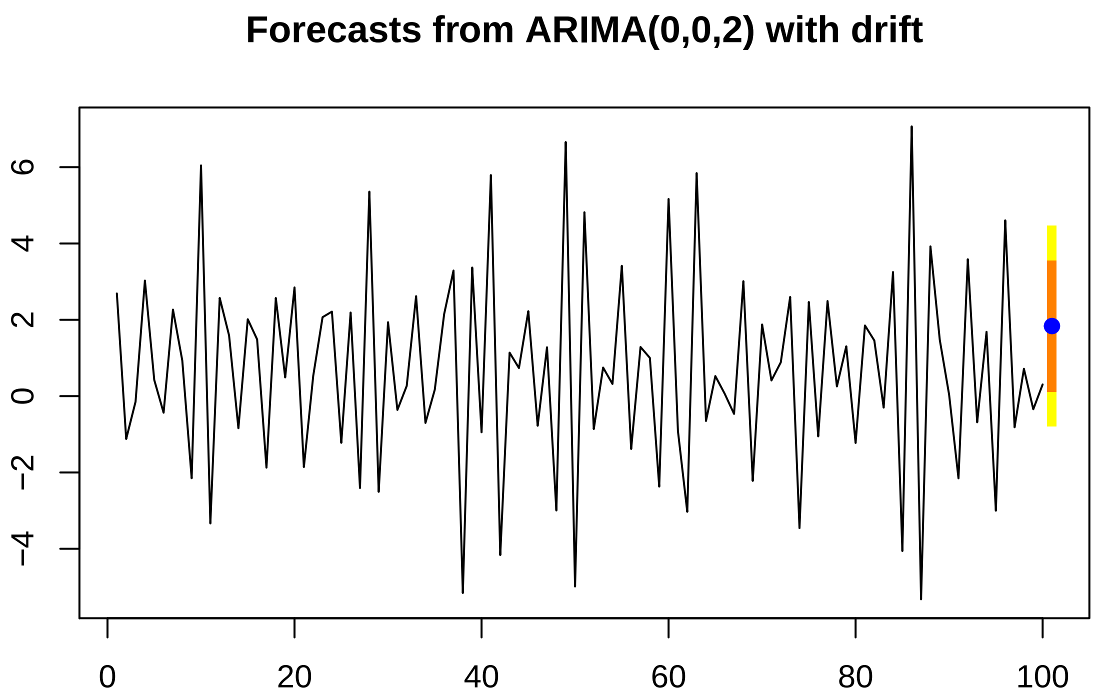
```
##
## Call:
## arma(x = ma2, order = c(0, 2))
##
## Model:
## ARMA(0,2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.2617417 -0.8255906  0.0009851  0.9141638  3.8073091
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ma1        -1.44618     0.08253   -17.52   <2e-16 ***
## ma2         0.71614     0.07547     9.49   <2e-16 ***
## intercept   0.69900     0.03677    19.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 1.892,  Conditional Sum-of-Squares = 183.55,  AIC = 353.56
```

```r
# We can see that ma1, ma2 and intercept are all of significant, so this is a ma2 process.

ma2_sim <- Arima(ma2,order=c(0,0,2),include.drift=TRUE)
plot(ma2,lwd=2)
```

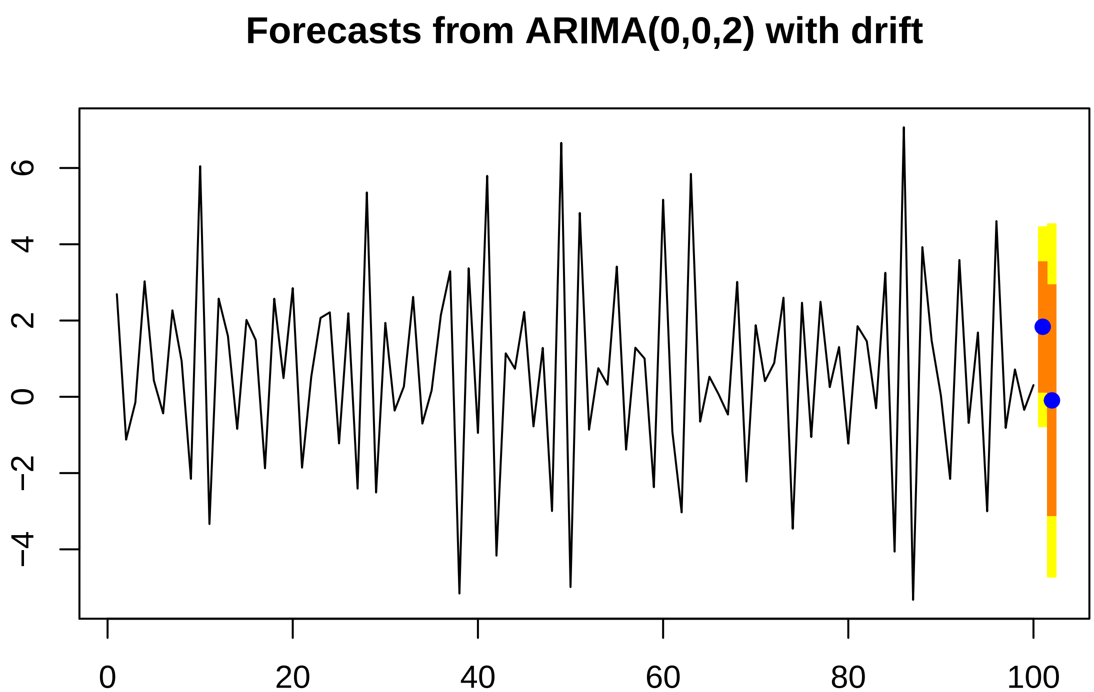
```r
lines(ma2_sim$fitted,col="blue",lwd=2)
legend("topright",pch= c("-","-"),legend=c("Generated ma2", "ma2 simulation"),col=c("black", "blue"),
```



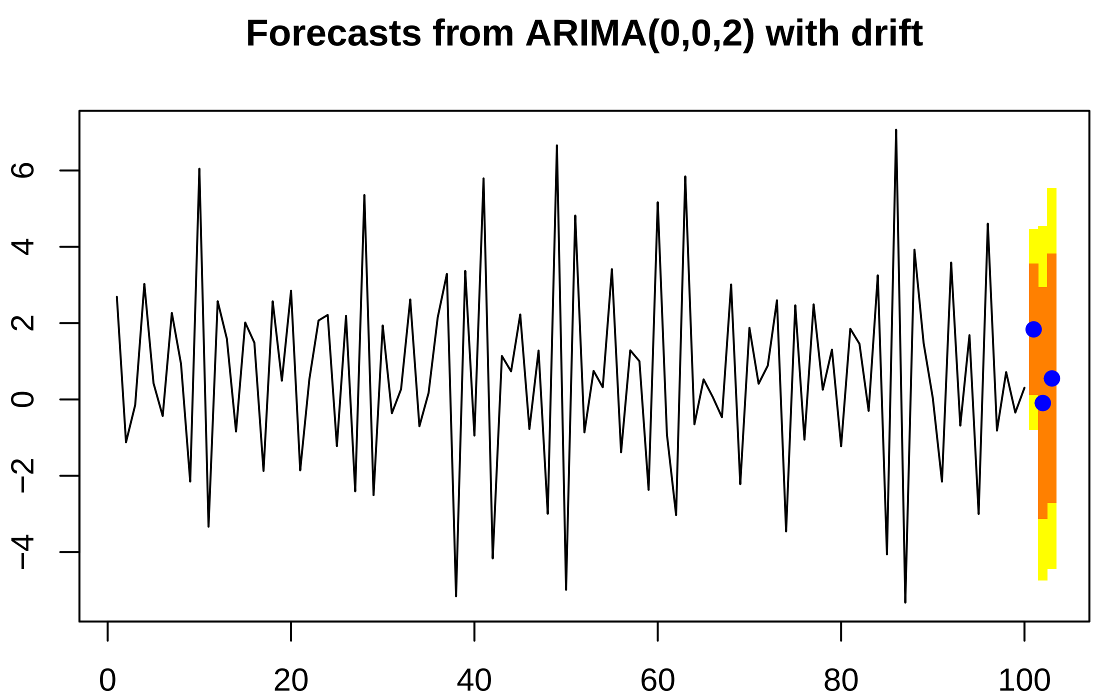

```r
# From the graph we can see that simulated ma2 model is very close to the theoretical model, but the ma

## b
# 1-step ahead
plot(forecast(ma2_sim,h=1),shadecols="oldstyle")
```




```r
# 2-step ahead
plot(forecast(ma2_sim,h=2),shadecols="oldstyle")
```



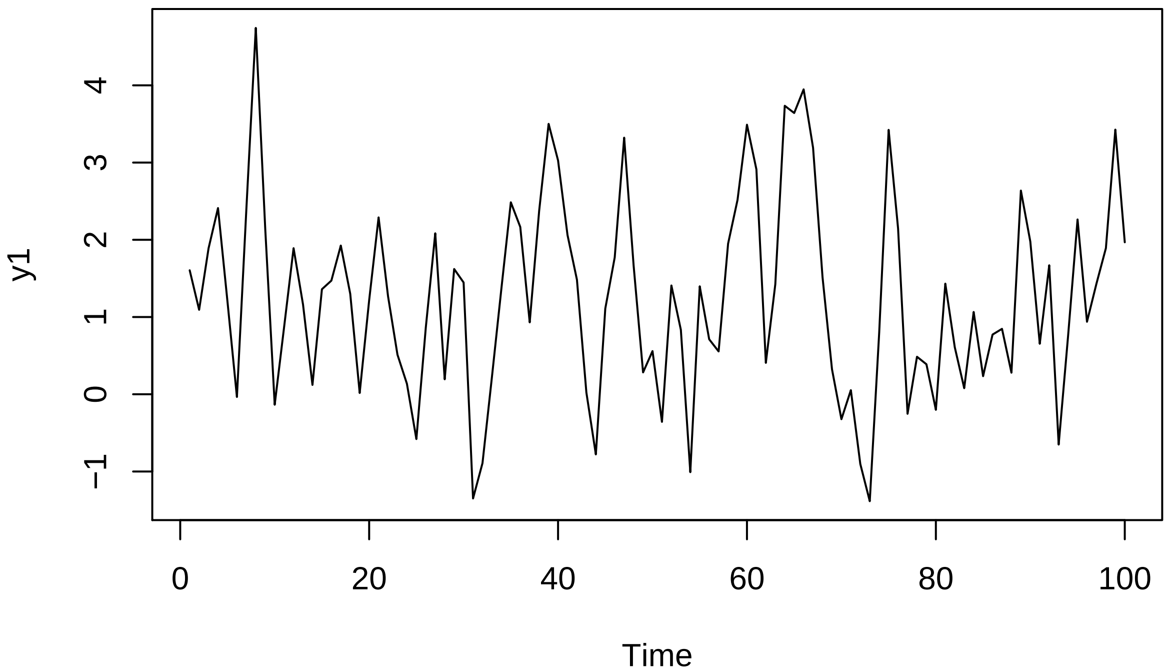

```
# 3-step ahead
plot(forecast(ma2_sim,h=3),shadecols="oldstyle")
```



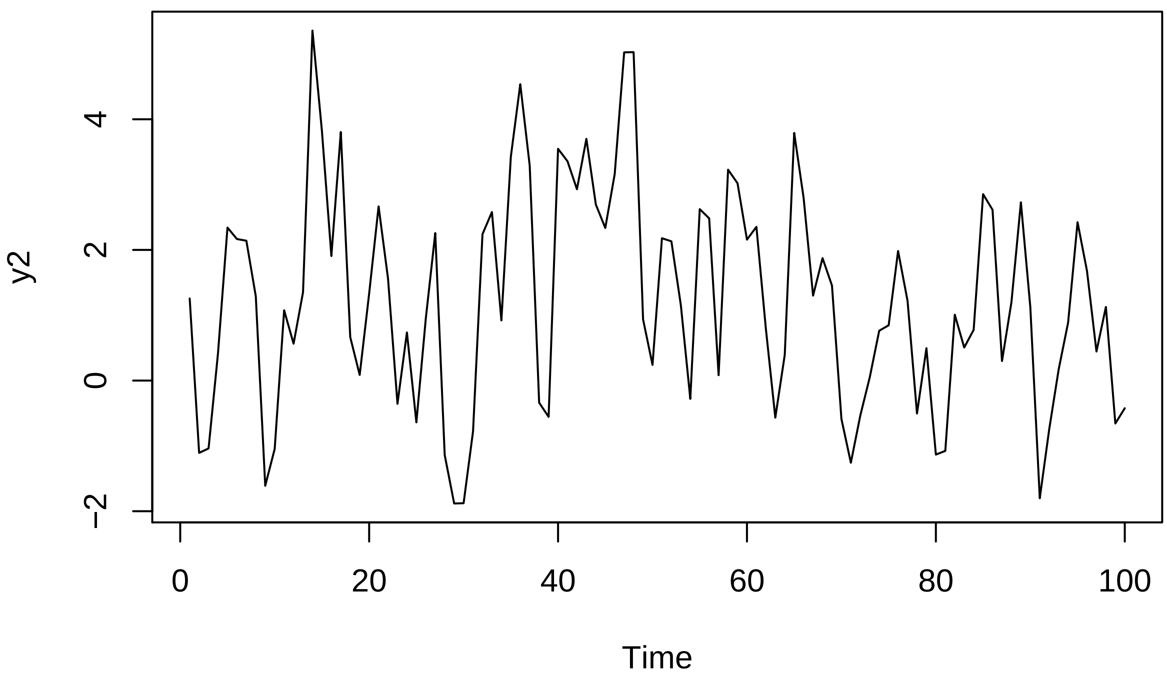
6.6 from book a

```
# first
y1 <- arima.sim(model = list(ma=c(0.8)),n=100)+1.2
# autoregressive:
# Y_t = 1.2 + 0.8*(Y_t-1 - 1.2) - 0.8^2*(Y_t-2 - 1.2) +... + (-1)^t*0.8^(t-1)*(Y_1 - 1.2) + e_t
```

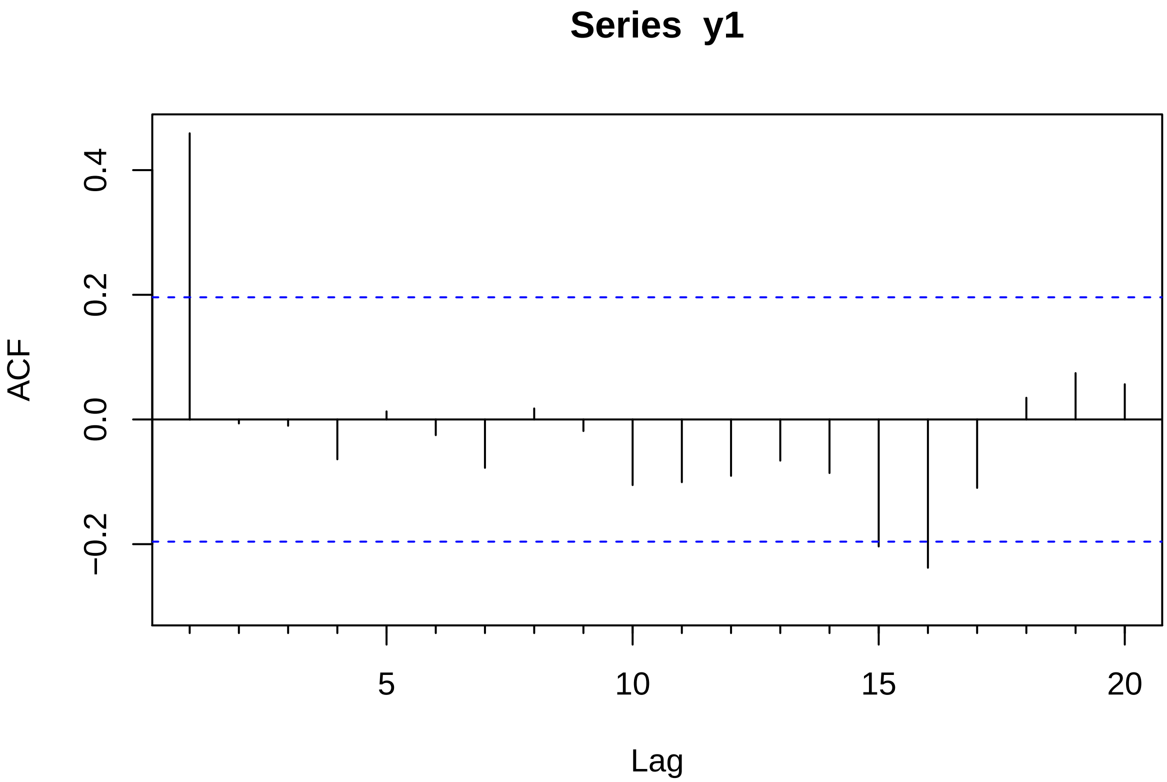
```r
# second
y2 <- arima.sim(model = list(ma=c(1.25)),n=100)+1.2
# autoregressive:
# -1/0.8 * Y_t+1 = 1/0.8^2 * Y_t+2 + 1/0.8^3 * Y_t+3 +... + e_t

# plot
plot(y1)
```



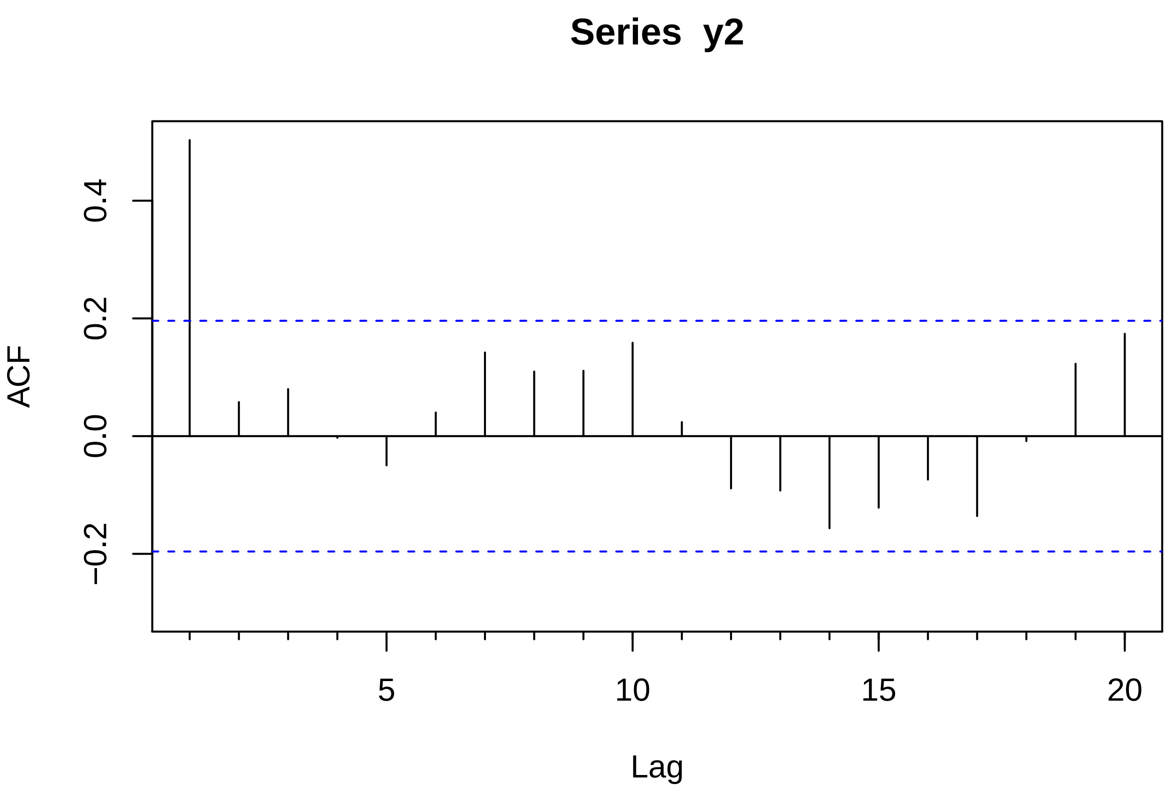

```r
plot(y2)
```



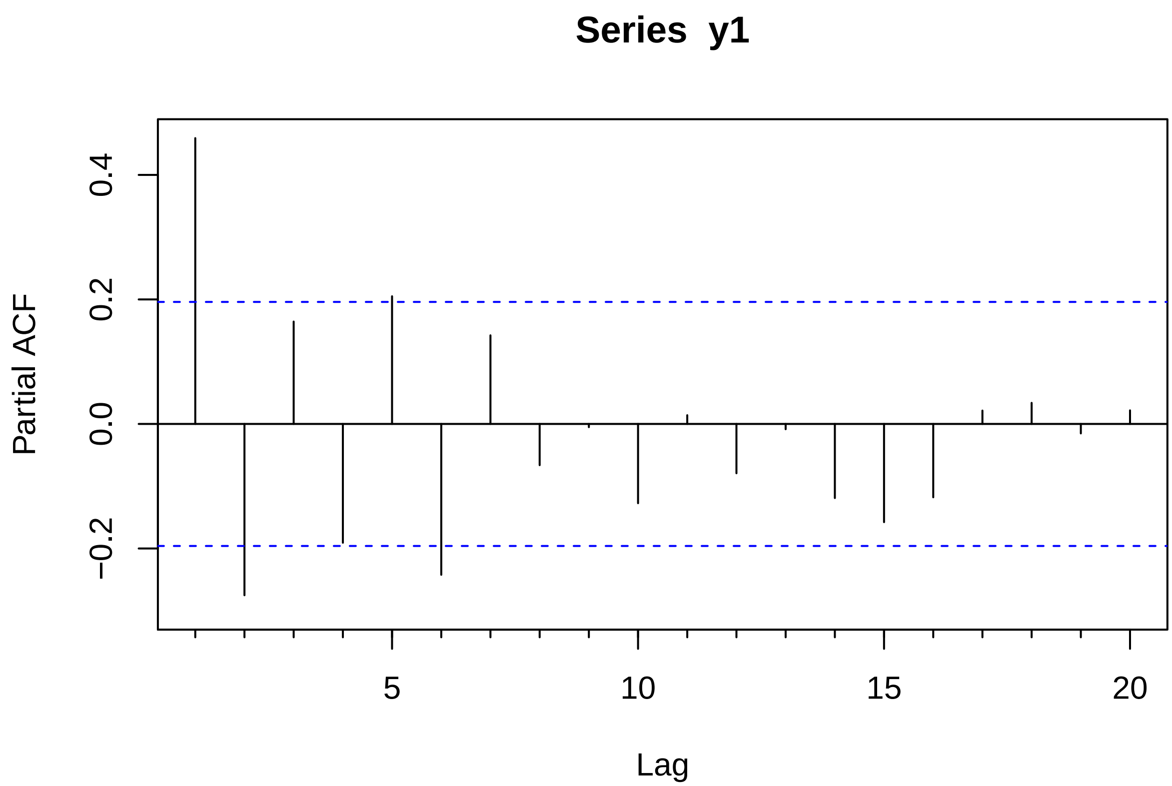
```r
Acf(y1)
```



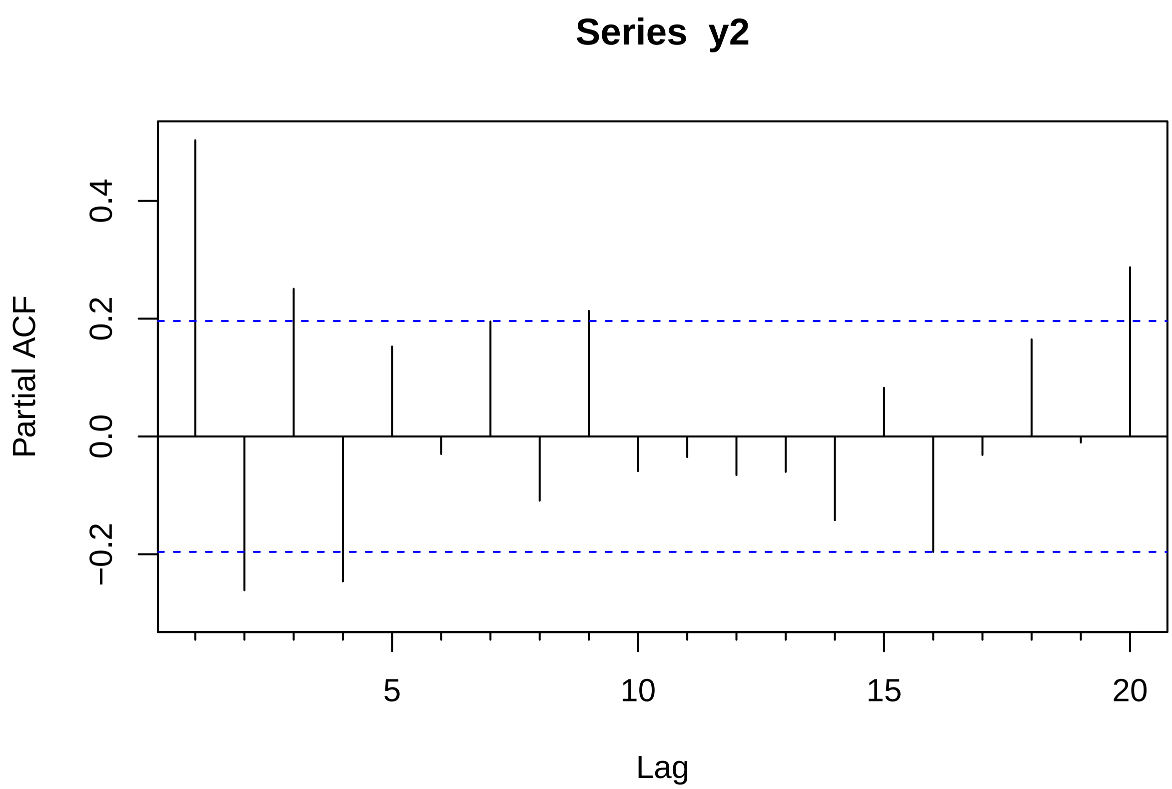

```r
Acf(y2)
```



```r
Pacf(y1)
```



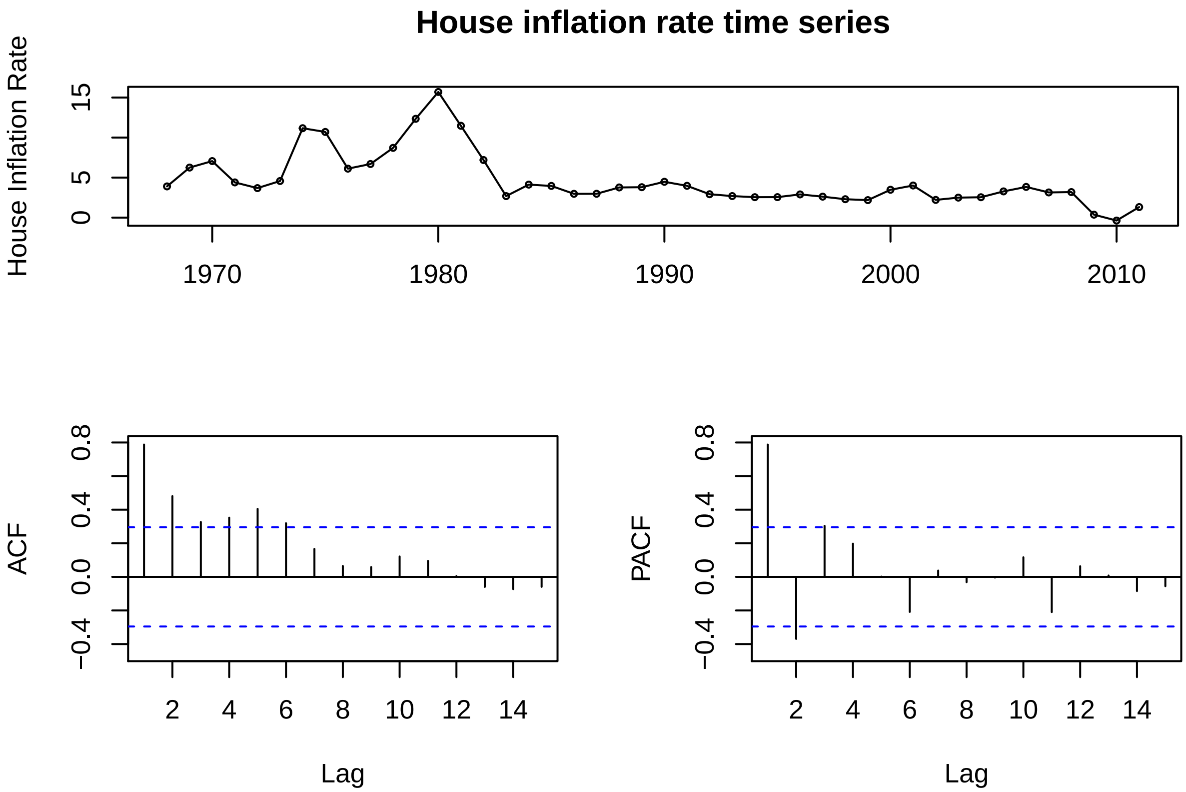

```r
Pacf(y2)
```



```
# I prefer the first model. Because its invertibility, we can convert MA(1) process into its own past, 
```

7.6 from book a

```
hw31 <- read.xls('hw37a.xls',sheet = 3)
# delete na
hw31 <- hw31[-c(1),]
# house inflation
hinf <-hw31$housing.Inflation....
thinf<-ts(hinf,start=1968,2011,freq = 1)
tsdisplay(thinf,ylab="House Inflation Rate",main="House inflation rate time series")
```

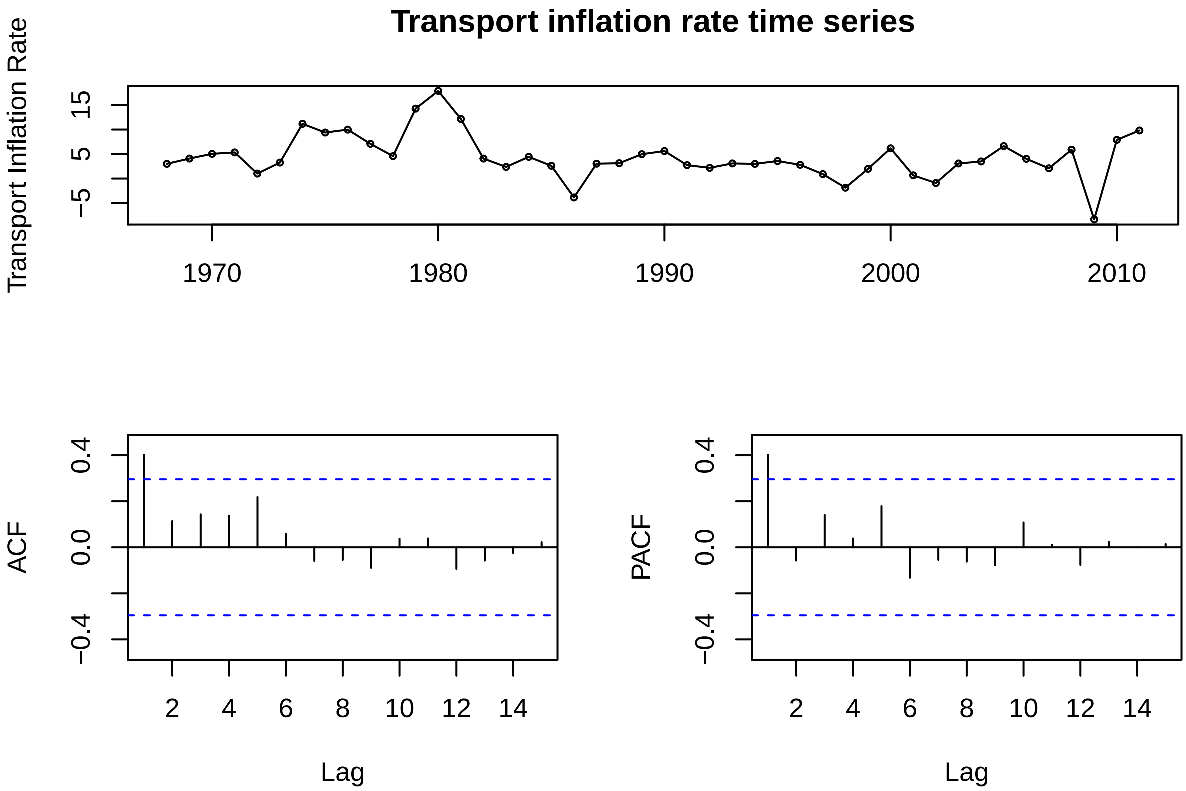



```
# from acf and pacf, housing inflation rate can be modeled as ar2
ar21 <- arma(hinf,order=c(2,0)) #Same as MA(1) = AR(0) + MA(1)
summary(ar21)
```

```
##
## Call:
## arma(x = hinf, order = c(2, 0))
##
## Model:
## ARMA(2,0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1746 -0.8164 -0.3451  0.4210  6.3067
##
```

```
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1          1.0997      0.1382    7.958 1.78e-15 ***
## ar2         -0.3749      0.1422   -2.636  0.00838 **
## intercept    1.2157      0.5125    2.372  0.01769 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 3.493,  Conditional Sum-of-Squares = 143.22,  AIC = 185.9
```

```
# from the t test, all coefficients are of significances. Hence house inflation rate can be modeled as 

# transportation inflation
tinf <-hw31$transportation.Inflation....
ttinf<-ts(tinf,start=1968,2011,freq = 1)
tsdisplay(ttinf,ylab="Transport Inflation Rate",main="Transport inflation rate time series")
```



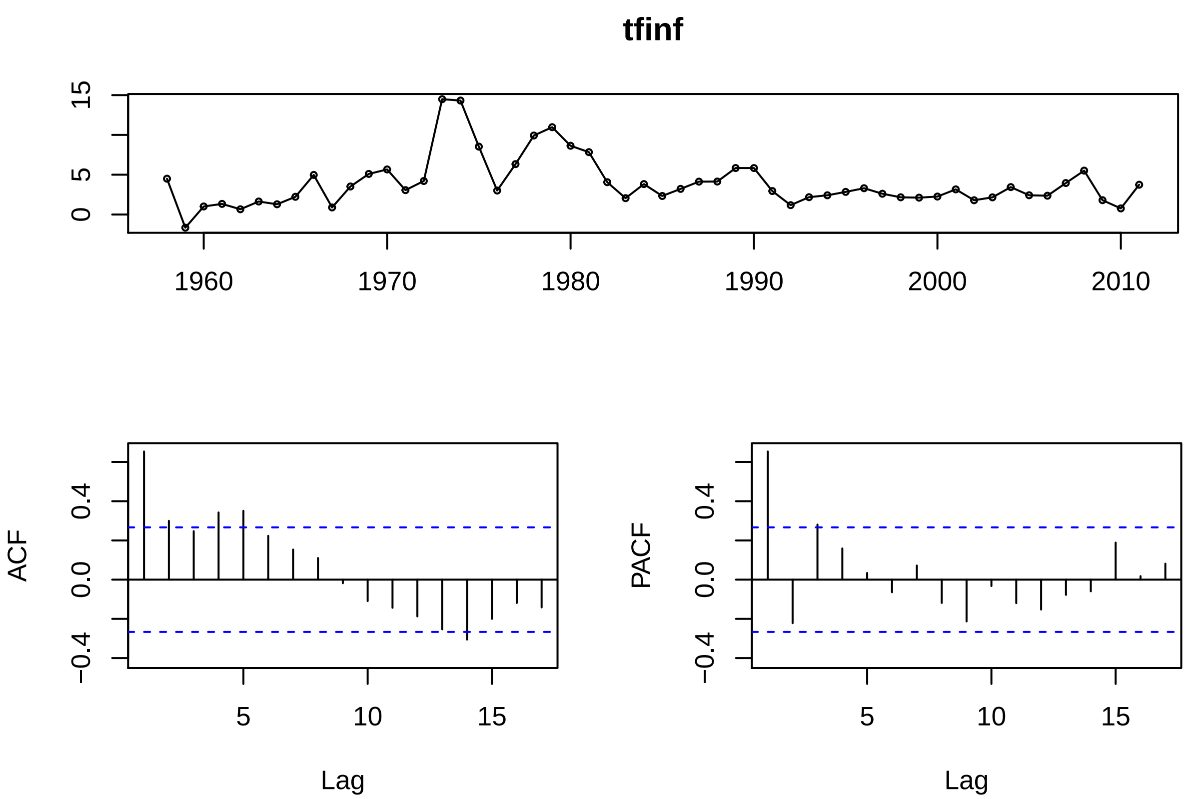

```
# from acf and pacf, transport inflation rate can not be modeled as ar2
ar22 <- arma(tinf,order=c(2,0))
summary(ar22)
```

```
##
## Call:
## arma(x = tinf, order = c(2, 0))
##
## Model:
## ARMA(2,0)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -13.5817  -1.8746  -0.1395   2.0365   9.8649
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1          0.4391      0.1522    2.886  0.00391 **
## ar2         -0.0553      0.1532   -0.361  0.71820
## intercept    2.7798      0.9489    2.930  0.00339 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 17.78,  Conditional Sum-of-Squares = 728.81,  AIC = 257.49
# from the t test, coefficient of ar2 is not of significance. Hence transportation inflation rate can n
```

7.7 from book a

```
hw32<-read.xls('hw37a.xls',sheet = 4)
hw32<-hw32[-c(1),] # delete na row

# food inflation
finf <- hw32$food.Inflation....
tfinf <- ts(finf,1958,2011,freq=1)
tsdisplay(tfinf)
```




```
ffit <- auto.arima(finf)
summary(ffit) # ar(1)+ma(1) process
```

```
## Series: finf
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1     ma1    mean
##       0.3709  0.5361  4.0364
## s.e.  0.1677  0.1632  0.7306
##
## sigma^2 estimated as 5.287:  log likelihood=-120.47
## AIC=248.94   AICc=249.75   BIC=256.89
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.03008777 2.234634 1.526076 -19.88114 60.99807 0.8351221
##                     ACF1
## Training set -0.01990706
```

```r
ffit2 <- arma(finf,order=c(1,0))# try ar1
summary(ffit2) # compare aic and find ffit works better
```

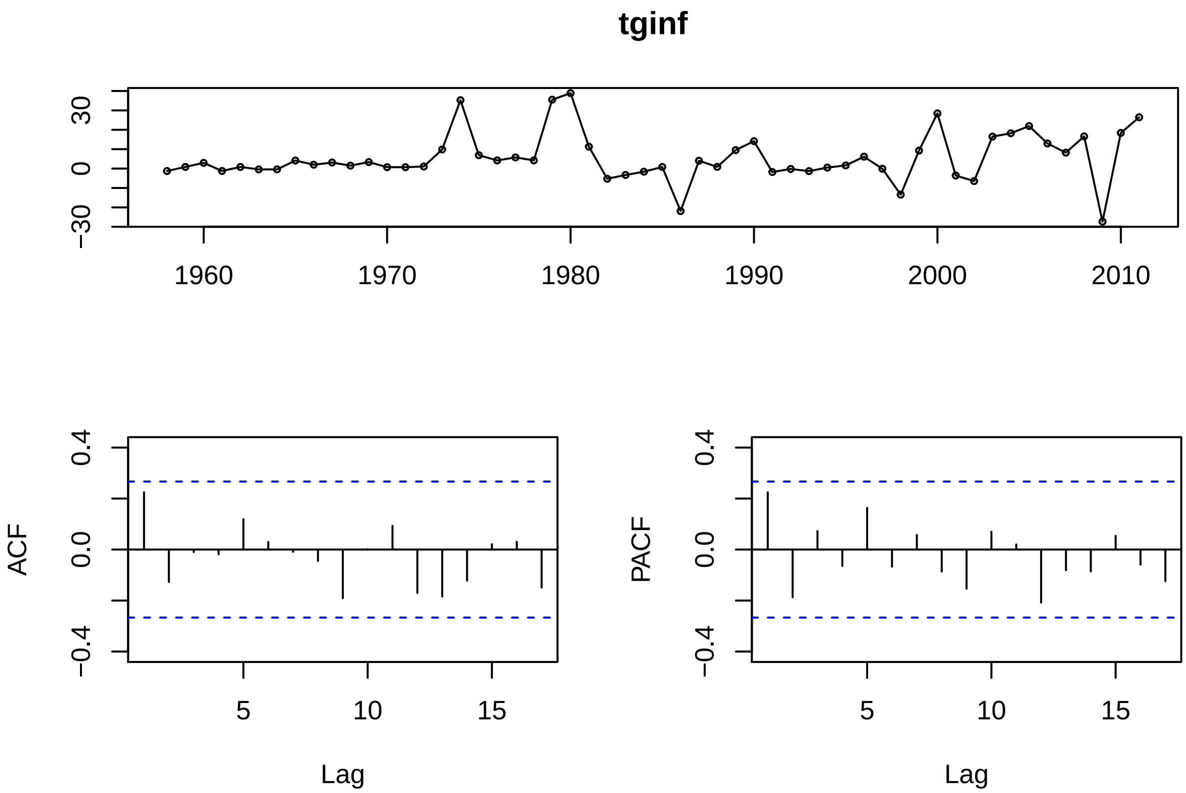
```
##
## Call:
## arma(x = finf, order = c(1, 0))
##
## Model:
## ARMA(1,0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.94928 -1.52663  0.02819  0.97395 10.38545
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1          0.6537      0.1029     6.35 2.16e-10 ***
## intercept    1.3533      0.5205     2.60  0.00932 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 5.824,  Conditional Sum-of-Squares = 302.85,  AIC = 252.39
```

```r
ffitr = Arima(finf,order=c(1,0,1),include.drift=TRUE) # (1,1) fit
# forecast
p1 <-forecast(ffit,h=1) # 1 step ahead
p2 <-forecast(ffit,h=2) # 2 step ahead
p3 <-forecast(ffit,h=3) # 3 step ahead
# forecast error
e1 <- recresid(ffitr$res~1)
e1
```

```
##  [1] -3.972323338  3.702457647 -1.144104112  0.502066183  0.773833165
##  [6] -0.175201771  1.312931889  2.645990676 -3.236547510  4.055600800
## [11]  0.517795012  2.225096852 -1.574164909  2.541114018  9.769020328
## [16]  1.421478091  0.018057585 -2.597697955  4.143908148  2.706706459
## [21]  3.035665545  0.018102093  1.639172565 -2.708150057 -0.895413848
## [26]  0.671203509 -2.262647667  0.823427494 -0.243383744  0.039118185
```

```
## [31]  1.572105276  0.105939084 -1.954407156 -1.485531651 -0.035306991
## [36] -0.902090064 -0.049714685 -0.177665709 -0.931889140 -0.675130900
## [41] -0.654254898 -0.469670199  0.311457385 -1.769188914  0.262175606
## [46]  0.335786090 -1.170368003 -0.004709248  0.990420844  1.428449276
## [51] -3.065797076 -0.262159508  1.582856512
```

```
# forecast uncertainty
# sigma^2 is 5.353

# gas inflation
ginf <- hw32$Gas.Inflation....
tginf <-ts(ginf,1958,2011,freq=1)
tsdisplay(tginf) # white noise
```




```
gfit <- auto.arima(ginf)
summary(gfit) # ma1 process
```

```
## Series: ginf
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##       0.3389  5.5646
## s.e.  0.1523  2.1390
##
## sigma^2 estimated as 144.5:  log likelihood=-209.94
## AIC=425.87   AICc=426.35   BIC=431.84
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
```

```
## Training set 0.02835135 11.7948 8.357421 190.3038 364.341 0.8604053
##                    ACF1
## Training set -0.0401626
```

```r
gfit2 <- arma(ginf,order=c(1,1)) # try ar1 +ma1 process
summary(gfit2)  # compare aic and find gfit works better
```

```
##
## Call:
## arma(x = ginf, order = c(1, 1))
##
## Model:
## ARMA(1,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -32.9360  -5.5391  -0.9959   6.3187  29.8594
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1        -0.59545     0.11332   -5.255 1.48e-07 ***
## ma1         0.95597     0.07325   13.051  < 2e-16 ***
## intercept   6.88947     3.60509    1.911    0.056 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 138,  Conditional Sum-of-Squares = 7224.22,  AIC = 425.3
```

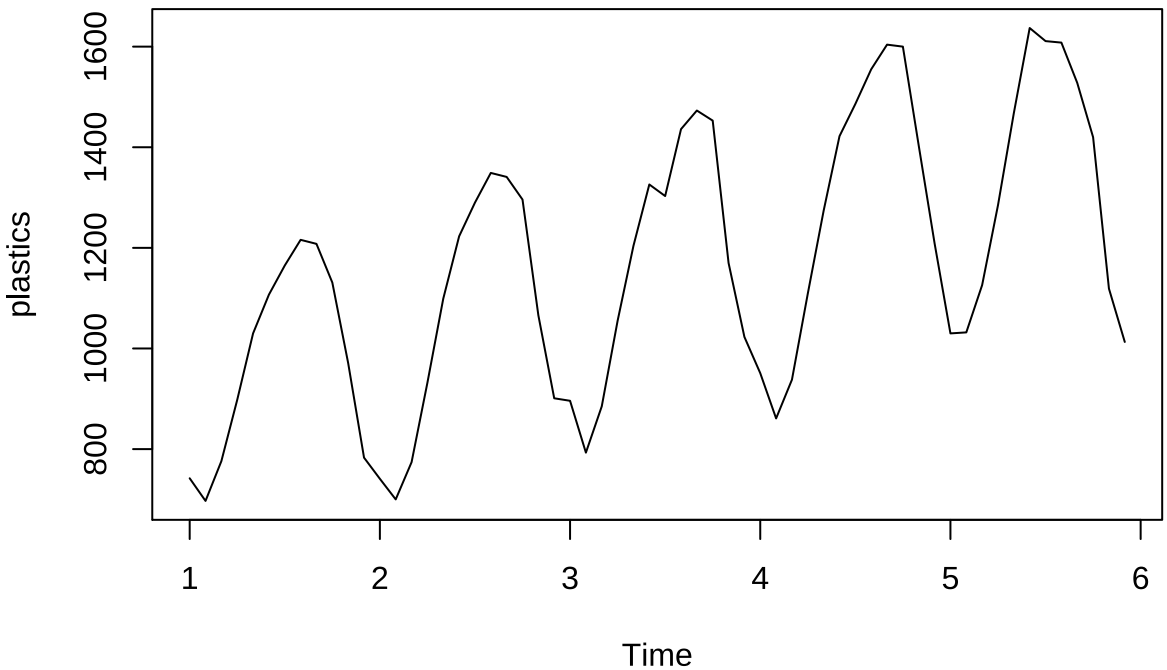
```r
# forecast
p21 <-forecast(ffit,h=1) # 1 step ahead
p22 <-forecast(ffit,h=2) # 2 step ahead
p23 <-forecast(ffit,h=3)#  3 step ahead
# forecast error
e2 <- recresid(gfit$residuals~1)
e2
```

```
##  [1]   2.71568017   2.30067183  -2.25563173   1.45331976  -1.05337720
##  [6]  -0.12202155   3.89922379   0.03699689   2.26783568  -0.24467771
## [11]   2.32249722  -1.26208177  -0.01841867  -0.06897297   8.46545398
## [16]  29.69473228  -6.96875651   2.68662591   0.89333956  -0.07131925
## [21]  30.82080503  22.26609895  -3.27066239 -10.99910310  -5.99307033
## [26]  -5.65211036  -3.10150329 -26.11722964   8.10582709  -6.56385706
## [31]   7.04748593   6.81108939  -9.05646988  -1.92580823  -5.29580013
## [36]  -2.18981805  -2.03234485   2.41192094  -5.26631446 -15.71789008
## [41]  10.67682186  20.57487535 -15.02053551  -5.62080169  14.09784509
## [46]   8.87762588  14.02580085   3.04726326   1.92556374  10.55040191
## [51] -36.12085758  25.61519593  12.44083323
```

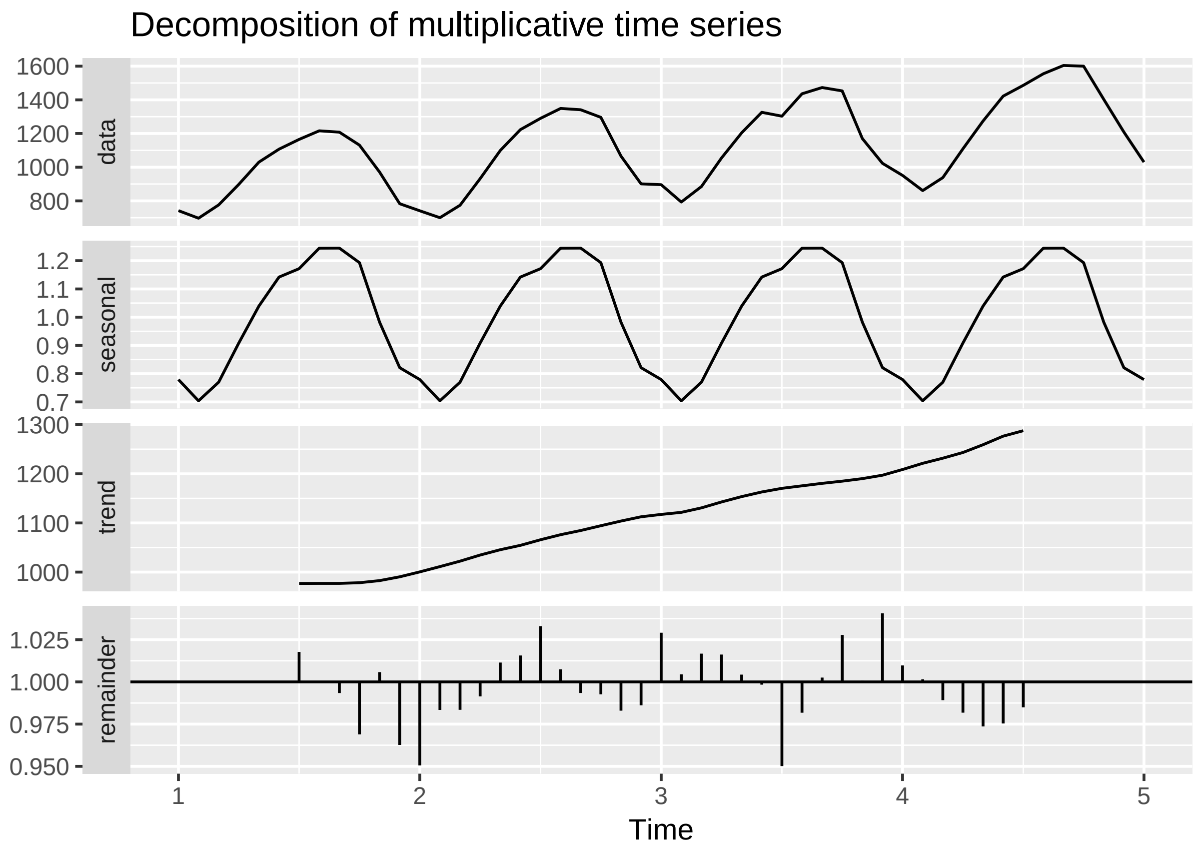
```r
# forecast uncertainty
# sigma^2  is 144.5
```

6.2 from book c

```r
## a
plot(plastics)
```




```r
# There is a seasonal cycle in between each unit time. A peak appears in the middle of the unit time. T
## b
plasticts<-ts(plastics,1,5,frequency = 12)
plasticts %>% decompose(type="multiplicative") %>%autoplot()
```

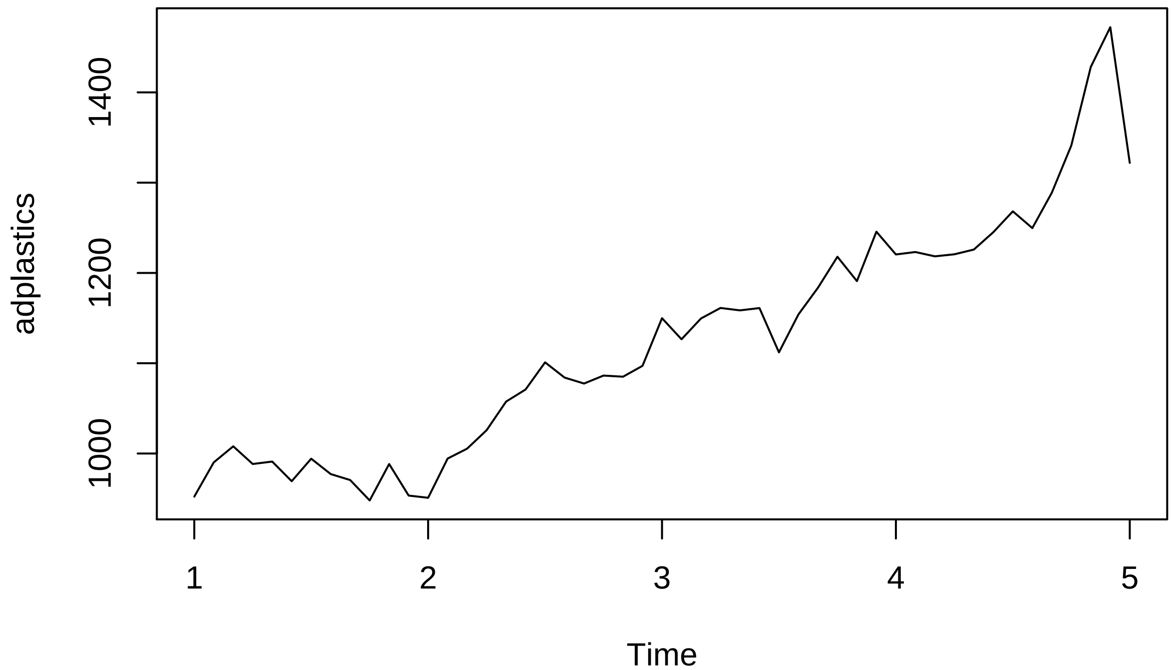



```
splastics<-decompose(plasticts,"multiplicative")
stlplastic <- stl(plasticts, s.window = "periodic")
# trend is in a linear growth
stlplastic # seasonal indices
```

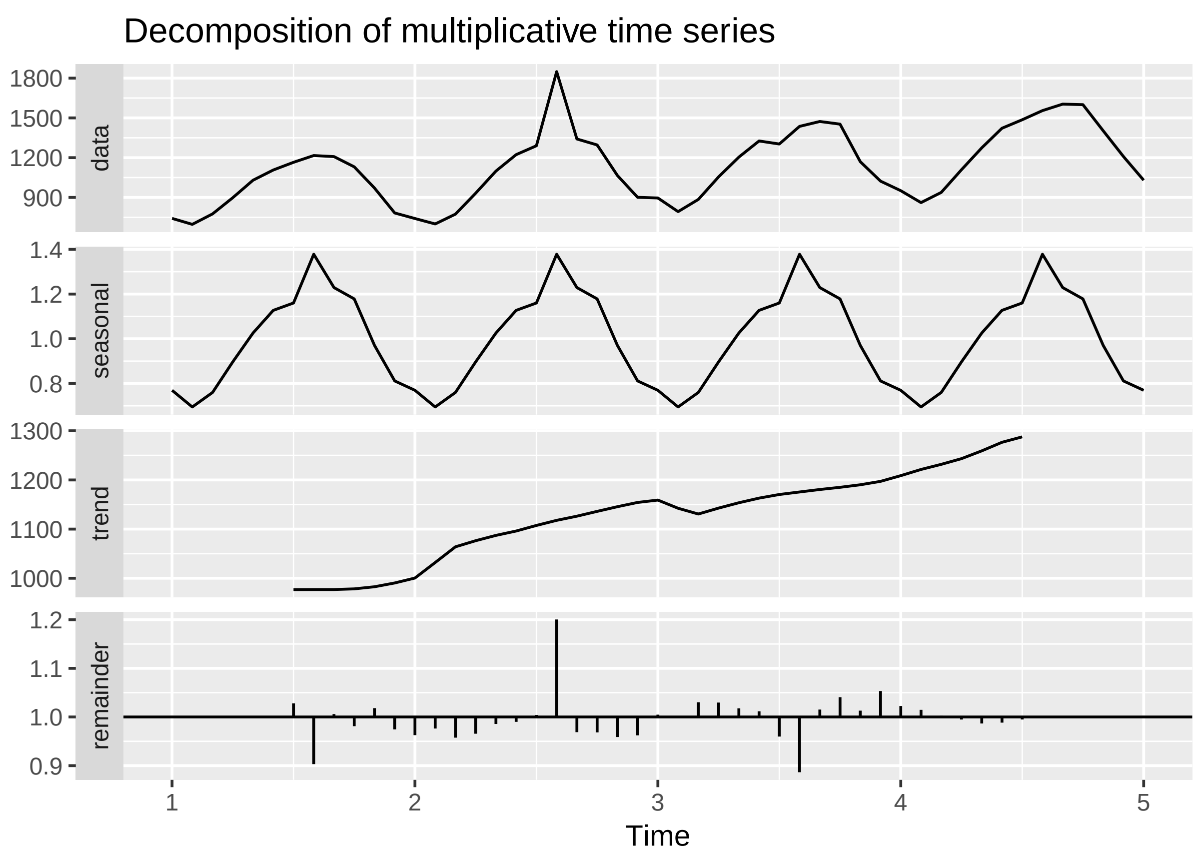
```
##  Call:
##  stl(x = plasticts, s.window = "periodic")
##
## Components
##          seasonal     trend   remainder
## Jan 1 -267.401536 1008.6525   0.7489974
## Feb 1 -328.779407 1003.4665  22.3129389
## Mar 1 -254.910287  998.2804  32.6298894
## Apr 1 -107.081843  994.0930  10.9888880
## May 1   38.746597  989.9055   1.3478902
## Jun 1  150.562280  986.7182 -30.2805216
## Jul 1  186.127941  983.5310  -4.6589110
## Aug 1  259.038125  980.8859 -23.9240554
## Sep 1  271.448290  978.2409 -41.6891809
## Oct 1  228.114417  979.6957 -76.8101479
## Nov 1    3.780506  981.1506 -13.9310760
## Dec 1 -179.644999  990.3793 -27.7343497
## Jan 2 -267.401536  999.6081   8.7934080
## Feb 2 -328.779407 1011.8558  16.9235699
## Mar 2 -254.910287 1024.1035   4.8067409
## Apr 2 -107.081843 1035.7099   3.3719315
## May 2   38.746597 1047.3163  12.9371256
```

```
## Jun 2  150.562280 1056.9442  15.4935119
## Jul 2  186.127941 1066.5721  37.2999206
## Aug 2  259.038125 1075.3960  14.5659196
## Sep 2  271.448290 1084.2198 -14.6680623
## Oct 2  228.114417 1092.8688 -24.9831943
## Nov 2    3.780506 1101.5178 -39.2982873
## Dec 2 -179.644999 1109.1337 -28.4887097
## Jan 3 -267.401536 1116.7496  46.6518993
## Feb 3 -328.779407 1125.2394  -3.4600208
## Mar 3 -254.910287 1133.7292   6.1810680
## Apr 3 -107.081843 1143.4251  18.6567381
## May 3   38.746597 1153.1210  12.1324118
## Jun 3  150.562280 1161.2976  14.1401009
## Jul 3  186.127941 1169.4742 -52.6021877
## Aug 3  259.038125 1175.0520   1.9098819
## Sep 3  271.448290 1180.6297  20.9219705
## Oct 3  228.114417 1186.2888  38.5967548
## Nov 3    3.780506 1191.9479 -25.7284219
## Dec 3 -179.644999 1199.9288   2.7161509
## Jan 4 -267.401536 1207.9098  10.4917551
## Feb 4 -328.779407 1218.7223 -28.9428455
## Mar 4 -254.910287 1229.5347 -36.6244370
## Apr 4 -107.081843 1244.4991 -28.4172294
## May 4   38.746597 1259.4634 -24.2100181
## Jun 4  150.562280 1274.3761  -2.9384245
## Jul 4  186.127941 1289.2889  10.5831916
## Aug 4  259.038125 1304.0947  -8.1327889
## Sep 4  271.448290 1318.9005  13.6512496
## Oct 4  228.114417 1334.1942  37.6913688
## Nov 4    3.780506 1349.4880  49.7315270
## Dec 4 -179.644999 1365.0308  23.6141821
## Jan 5 -267.401536 1380.5737 -83.1721314
```

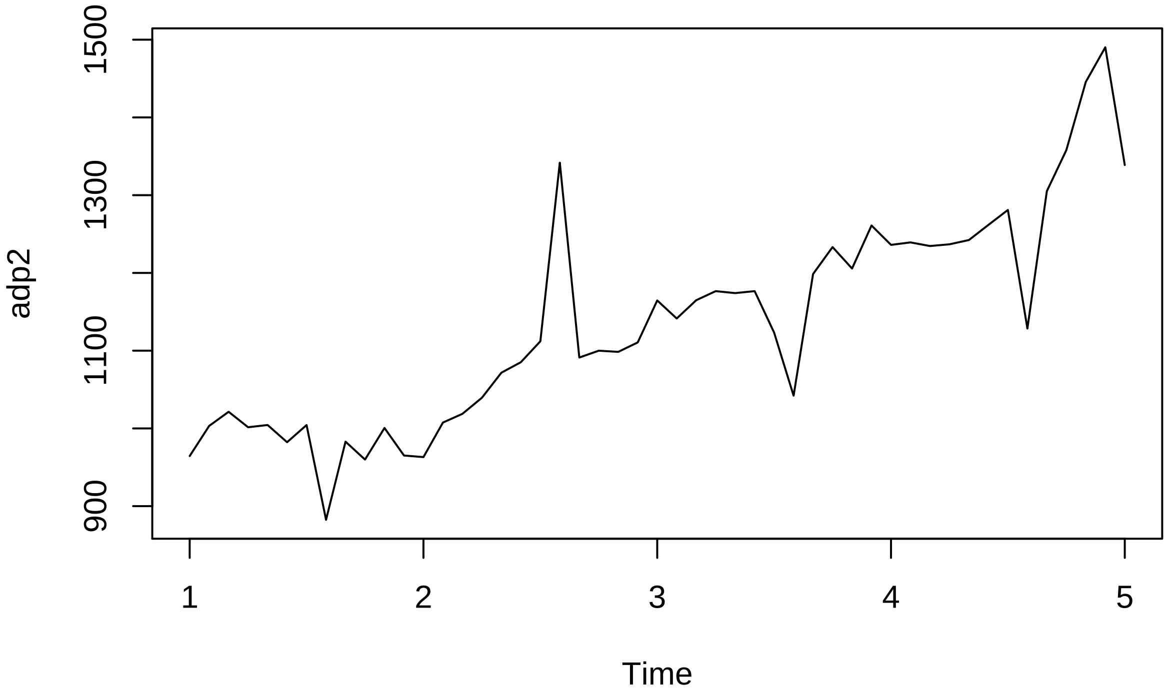
```
## c
# Yes, it supports.

## d
adplastics <- plastics/splastics$seasonal
plot(adplastics) # adjust plastics data
```



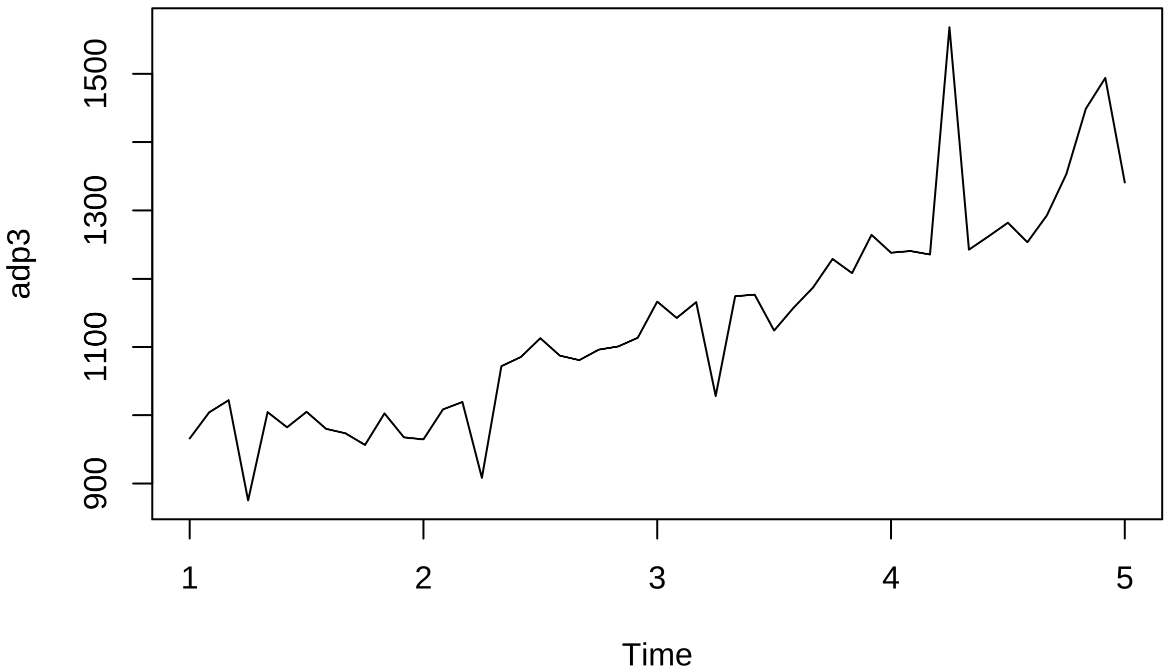

```r
## e
p2<-plastics
p2[20]<-plastics[20]+500
p2ts<-ts(p2,1,5,frequency = 12)
p2ts %>% decompose(type="multiplicative") %>%autoplot()
```



```r
psd<-decompose(p2ts,type="multiplicative")
adp2<-seasadj(psd)
plot(adp2)
```



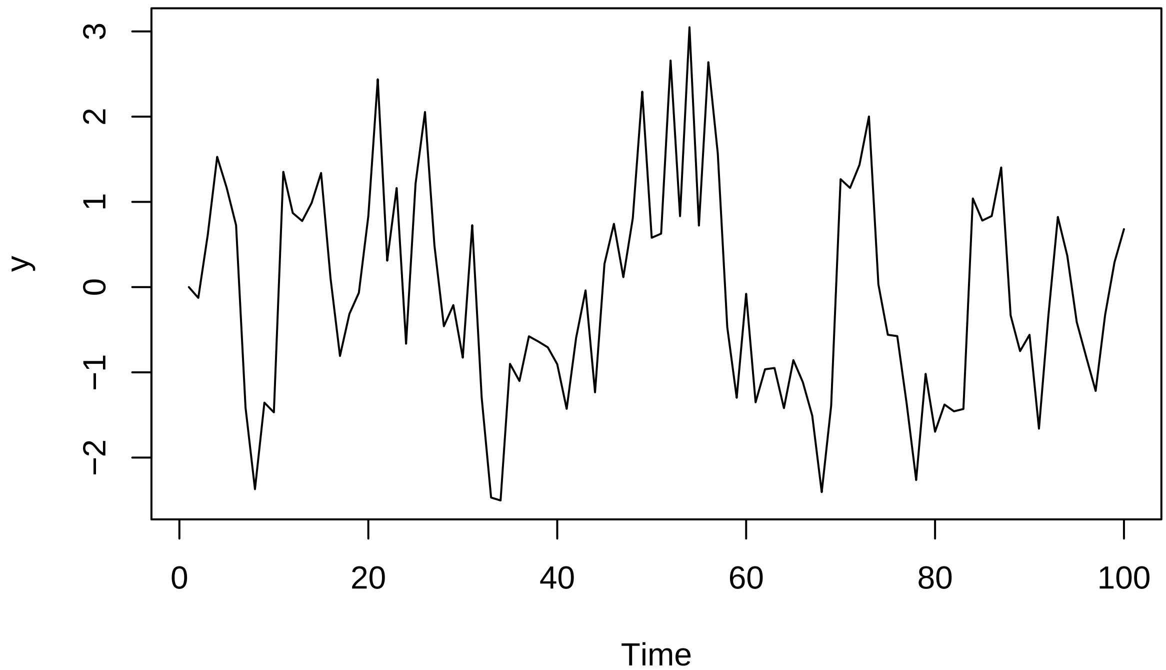

```r
# In e, the new outlier drives the seasonal adjusted time series to have a spike at the point we add 50
## f
p3<-plastics
p3[40]<-plastics[40]+500
p3ts<-ts(p3,1,5,frequency = 12)
psd3 <- decompose(p3ts, type='multiplicative')
adp3 <- seasadj(psd3)
plot(adp3)
```



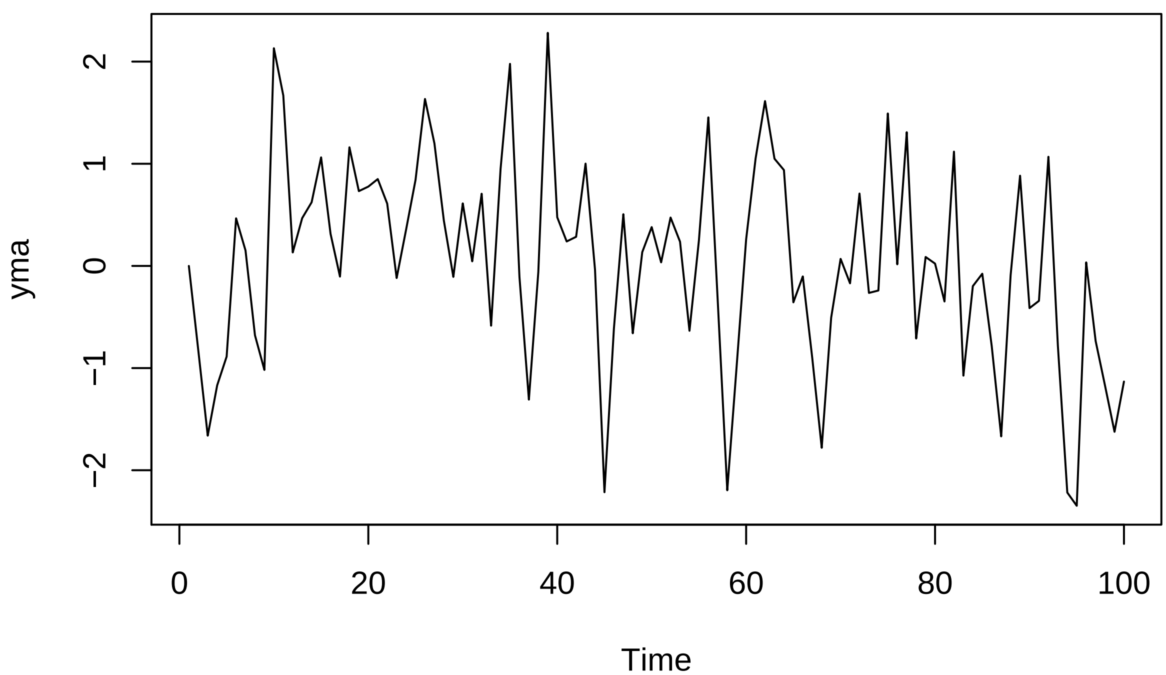

```
# no matter where the outlier exists, it will always generate similar spike but at different location, 
```

8.6 from book c

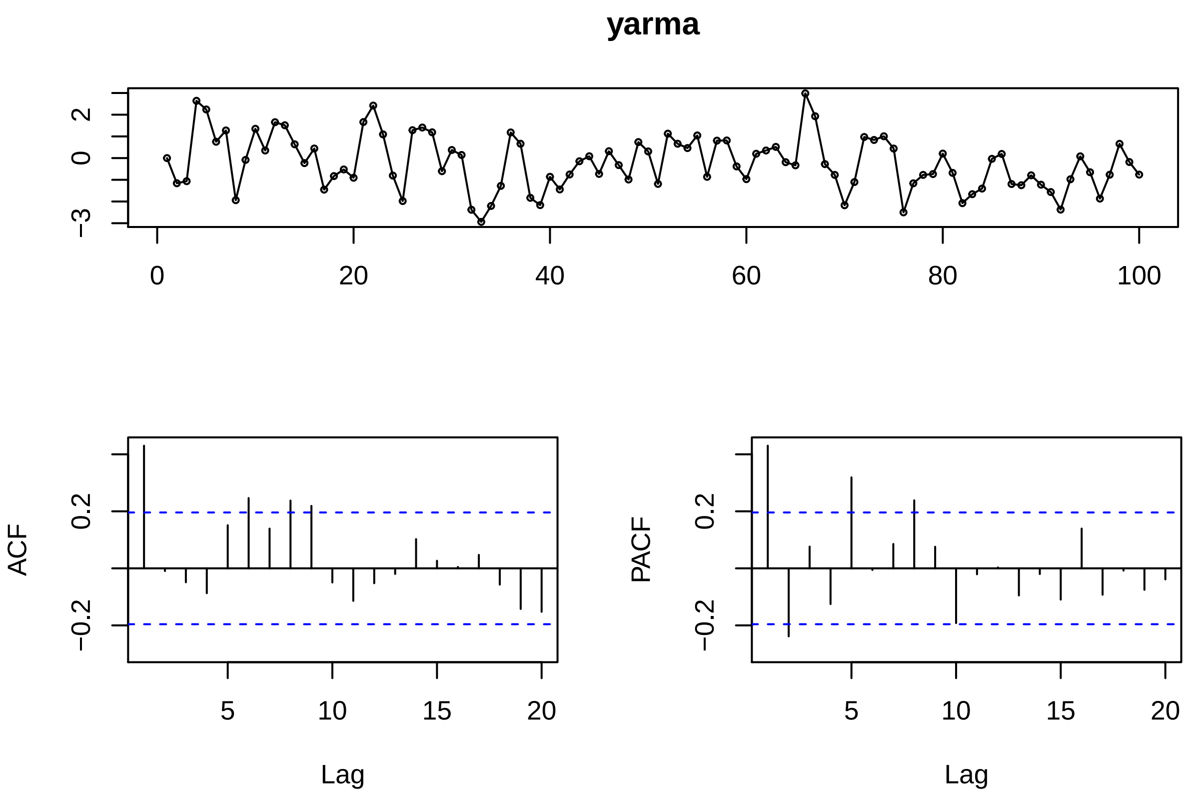
```
## a
# generate ar(1)
y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.6*y[i-1] + e[i]
## b
plot(y)
```



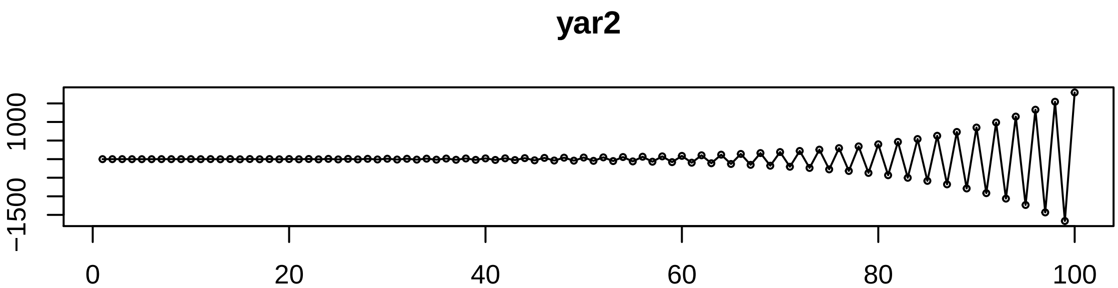

```
# as I change the value of ph1, the graph changes. As ph1 becomes larger, the graph illustrates more pe
## c
# generate ma(1)
yma <- ts(numeric(100))
ema <- rnorm(100)
for (i in 2:100)
  yma[i] <- 0.6*ema[i-1]+ema[i]
## d
plot(yma)
```

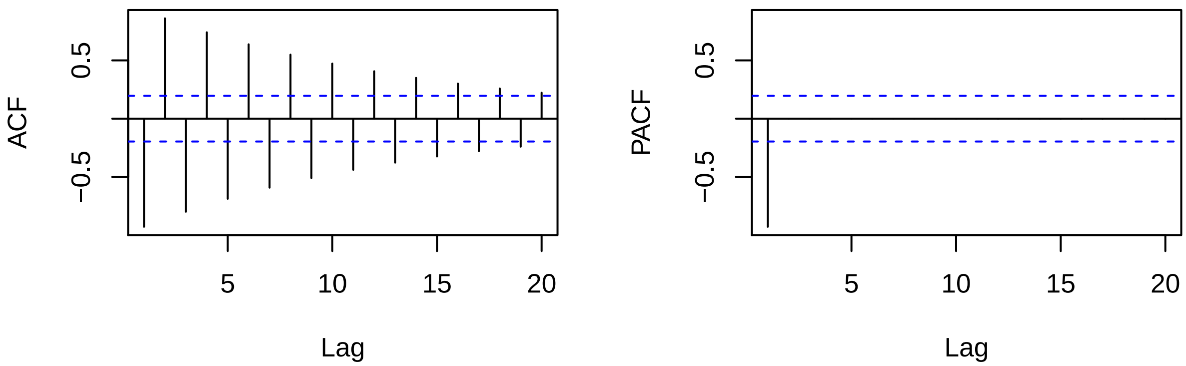


```
#  as I change theta 1 in ma(1) model, the graph changes. As thta 1 becomes smaller, the frequency of f
## e
# generate arma(1,1)
yarma <- ts(numeric(100))
earma <- rnorm(100)
for (i in 2:100)
  yarma[i] <- 0.6*y[i-1]+0.6*earma[i-1]+earma[i]
## f
# generate ar(2)
yar2 <- ts(numeric(100))
ear2 <- rnorm(100)
for(i in 3:100)
  yar2[i] <- -0.8*yar2[i-1] +0.3*yar2[i-2]+ear2[i]
## g
# graph last two and compare
tsdisplay(yarma)
```




```
tsdisplay(yar2)
```






```
# The oscilation magnitude increases as time increases in ar2 model, which is a non-stationary process.
```