2

1.

```
library(itsmr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:itsmr':
##
##     deaths
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.4
```

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

```
##
## Attaching package: 'tseries'
```

```
## The following object is masked from 'package:itsmr':
##
##     arma
```

```
library(nortest)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.4
```

```
##
## Attaching package: 'forecast'
```

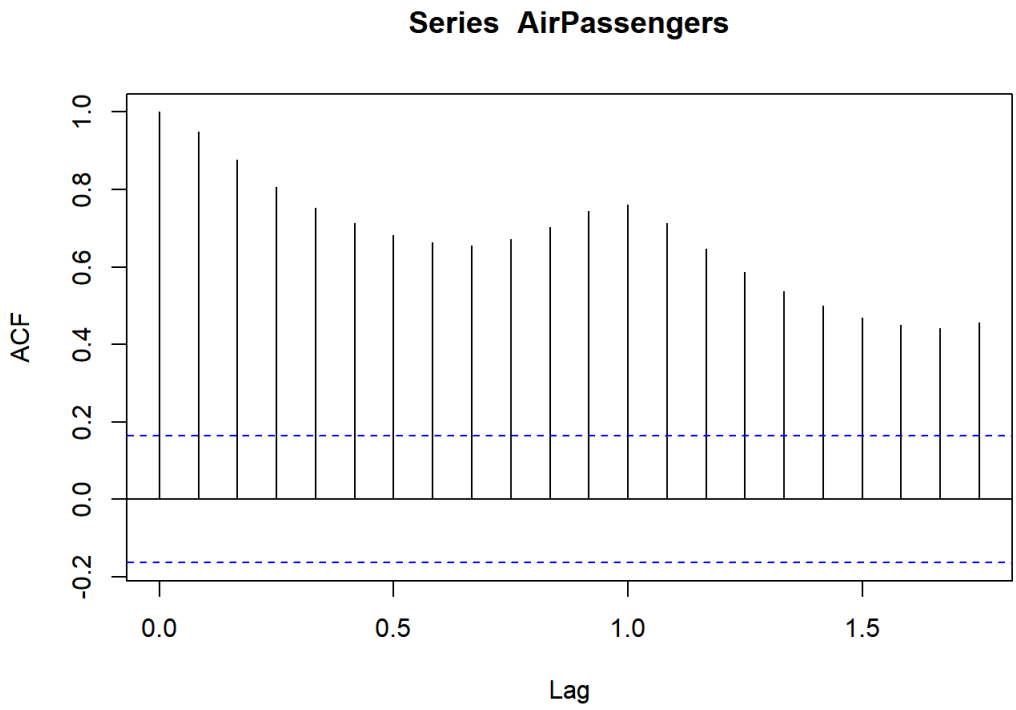
```
## The following object is masked from 'package:itsmr':
##
##     forecast
```

2. ACF, PACF

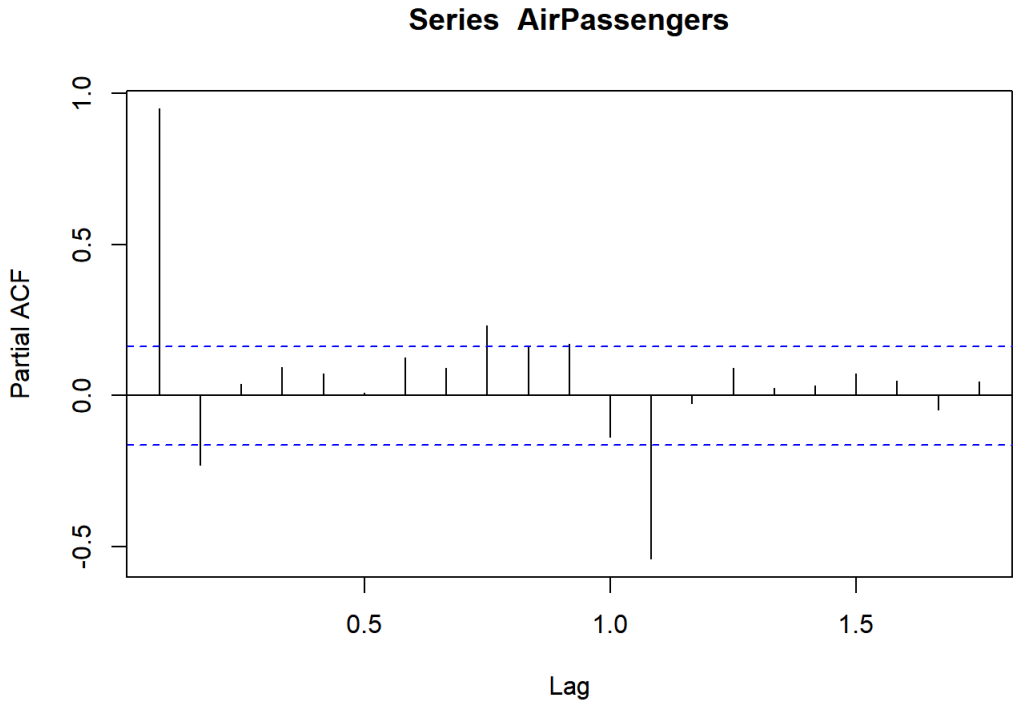
ACF, PACF .

`AirPassengers` ACF PACF

```
acf(AirPassengers)
```




```
pacf(AirPassengers)
```



3. AR, MA

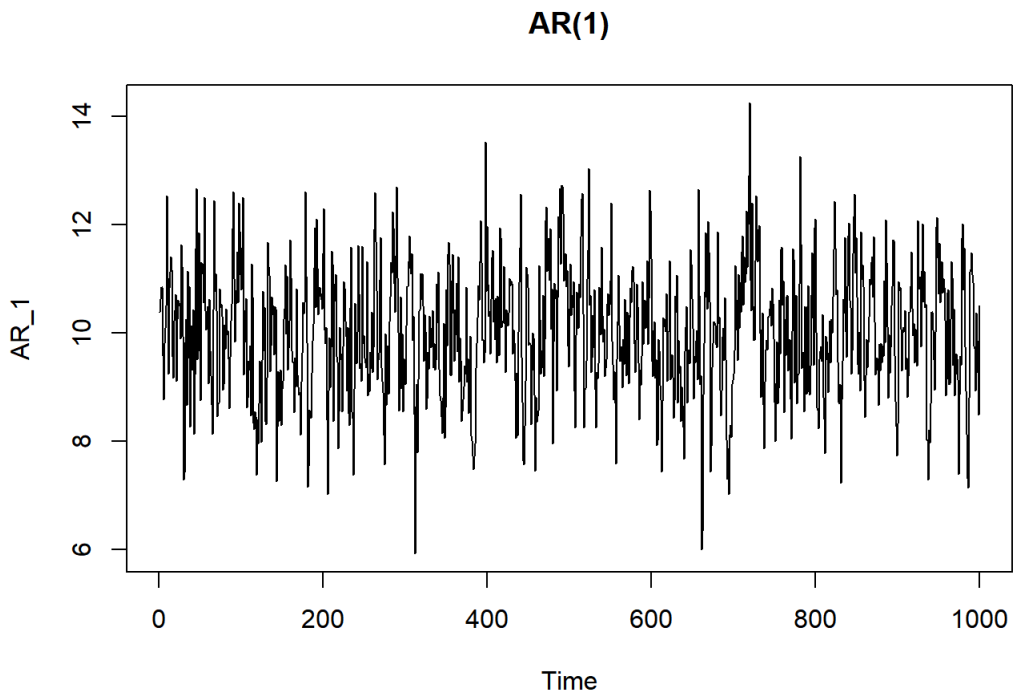
AR MA

`arima.sim()` AR, MA, .

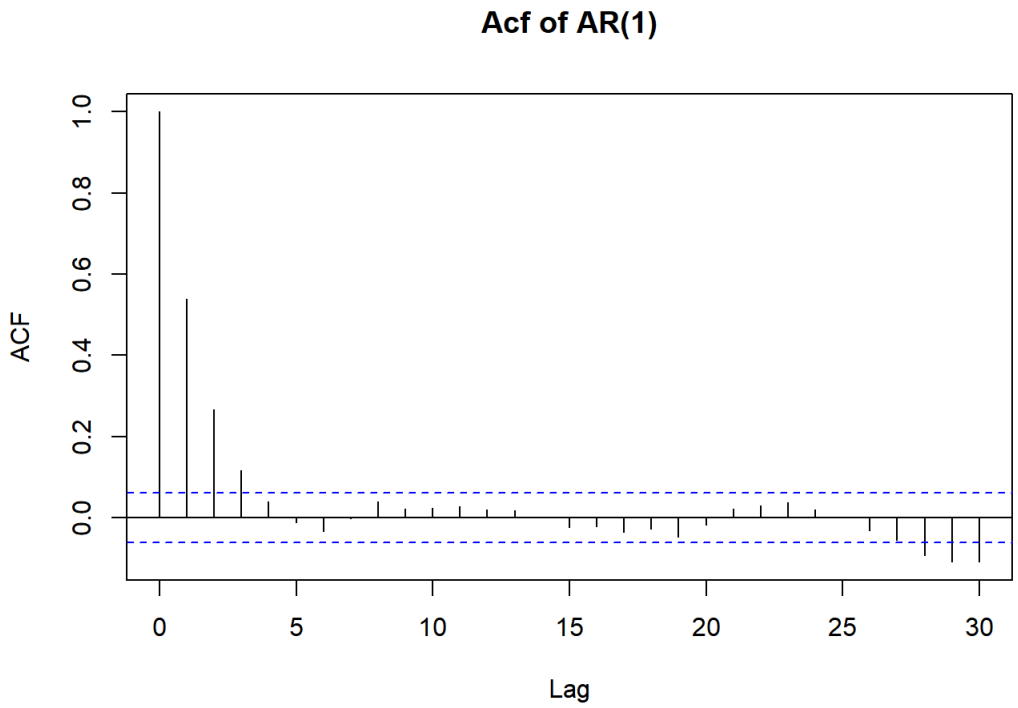
acf pacf

AR(1)

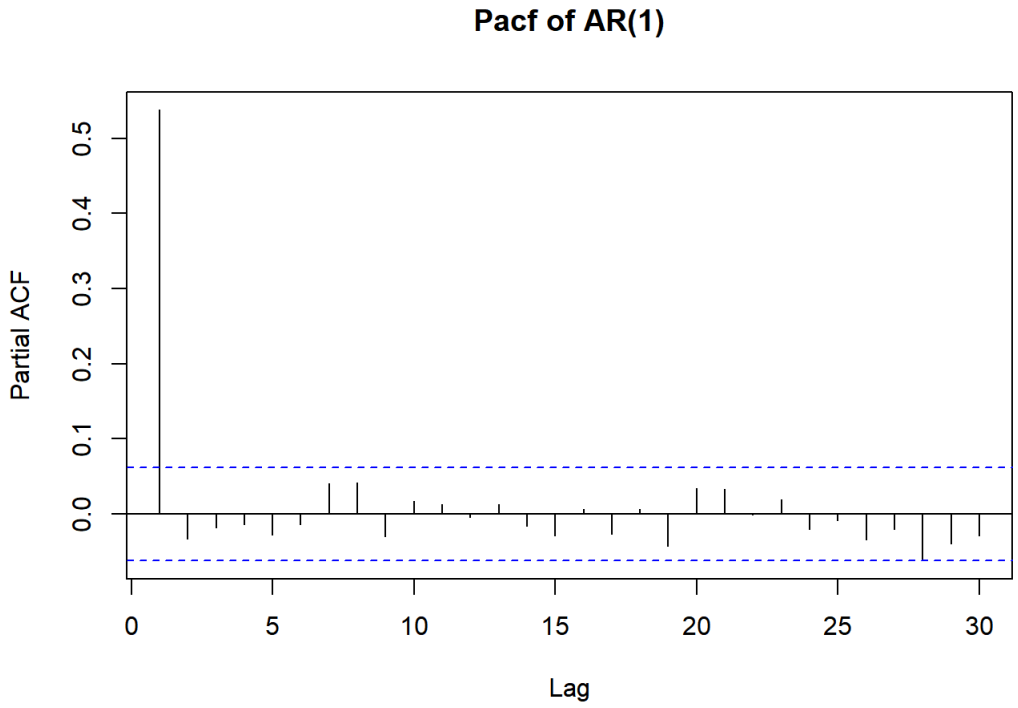
```
AR_1 <- arima.sim(model = list(order = c(1, 0, 0), ar = c(0.5)), n =1000, mean=5)
plot(AR_1,main="AR(1)")
```




```
acf(AR_1,main="Acf of AR(1)")
```



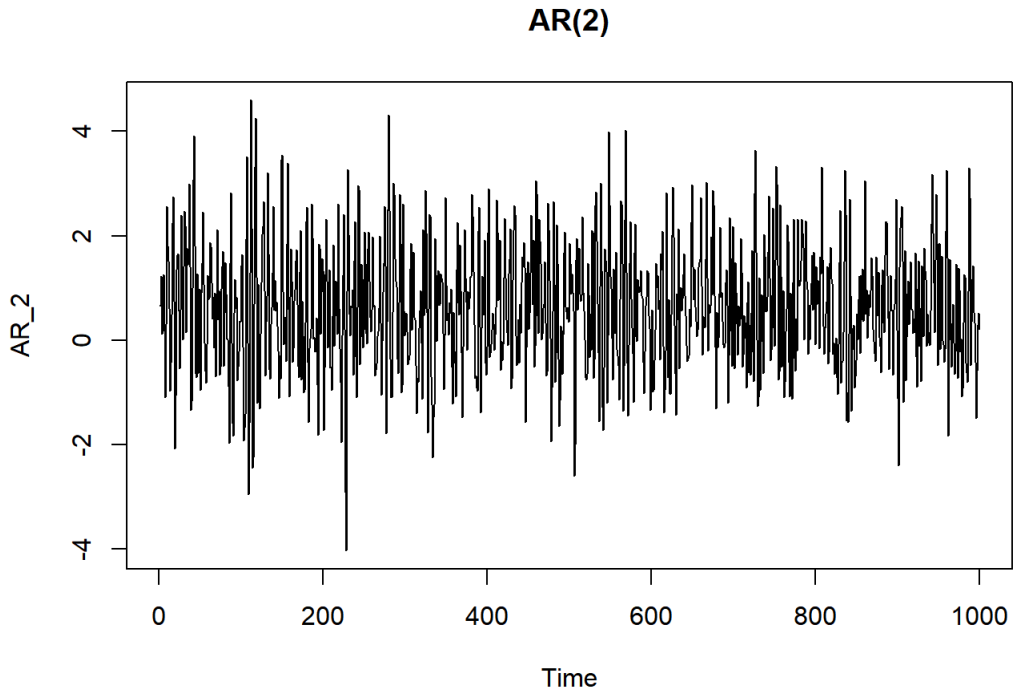

```
pacf(AR_1,main="Pacf of AR(1)")
```



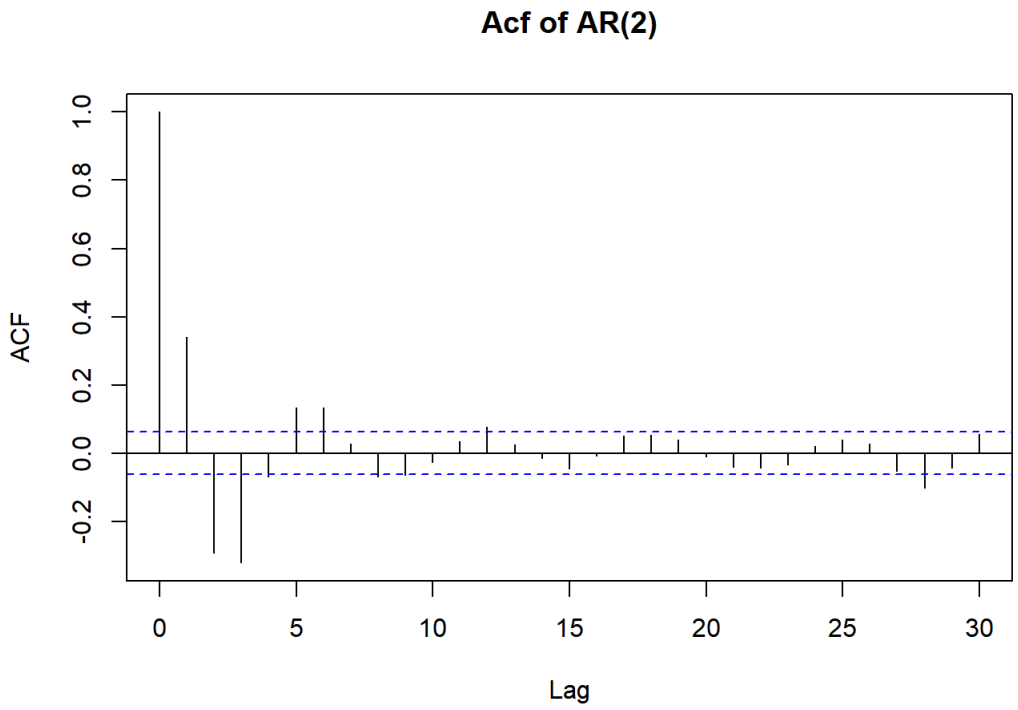
- acf .
- pacf 2 .

AR(2)

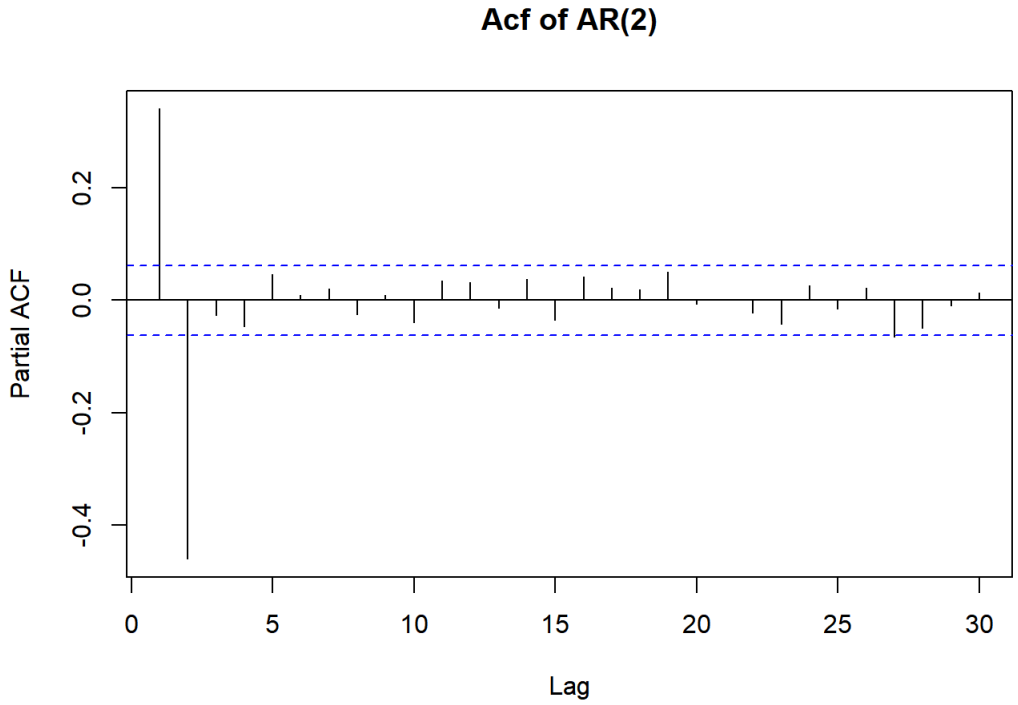
```
AR_2 <- arima.sim(model = list(order = c(2, 0, 0), ar = c(0.5, -.45)), n =1000, mean=0.6)

plot(AR_2,main="AR(2)")
```




```
acf(AR_2,main="Acf of AR(2)")
```



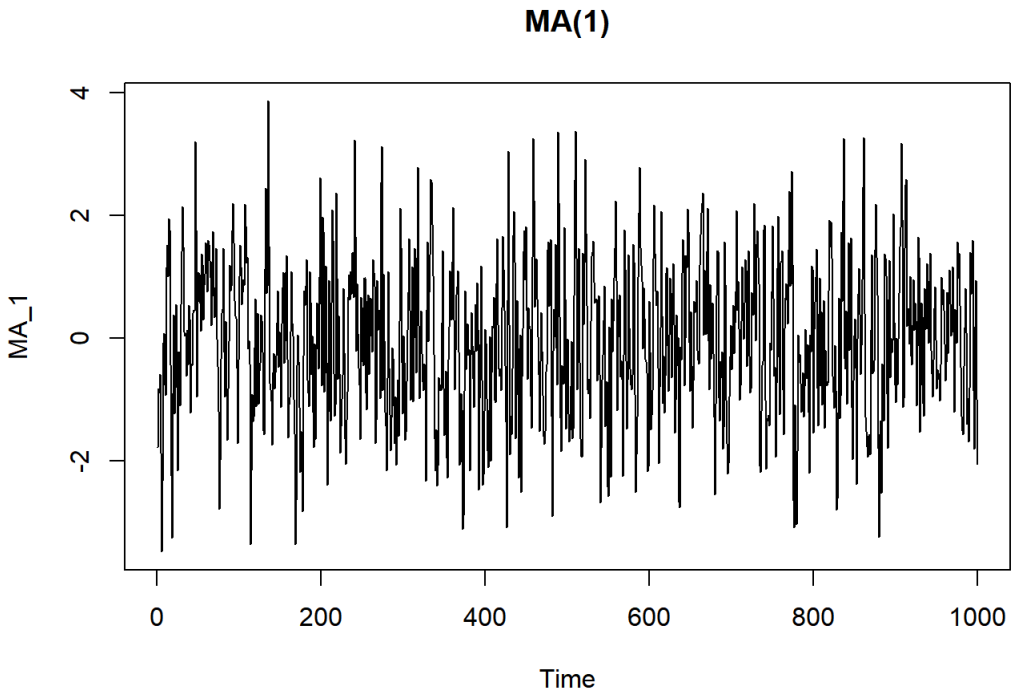

```
pacf(AR_2,main="Acf of AR(2)")
```



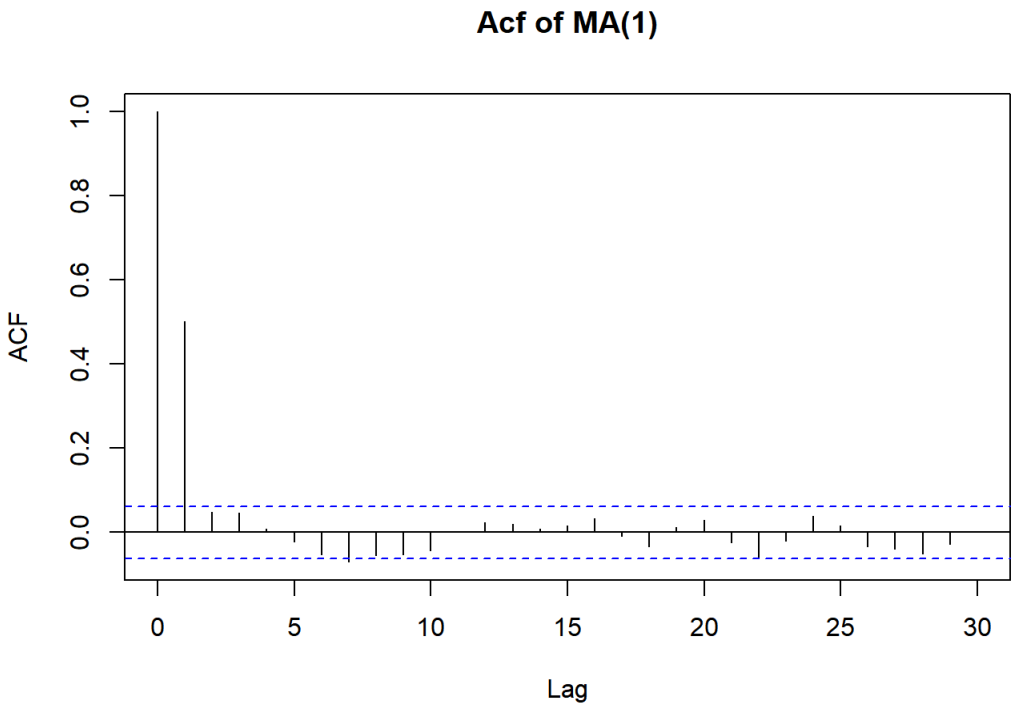
- acf .
- pacf 3 .

MA(1)

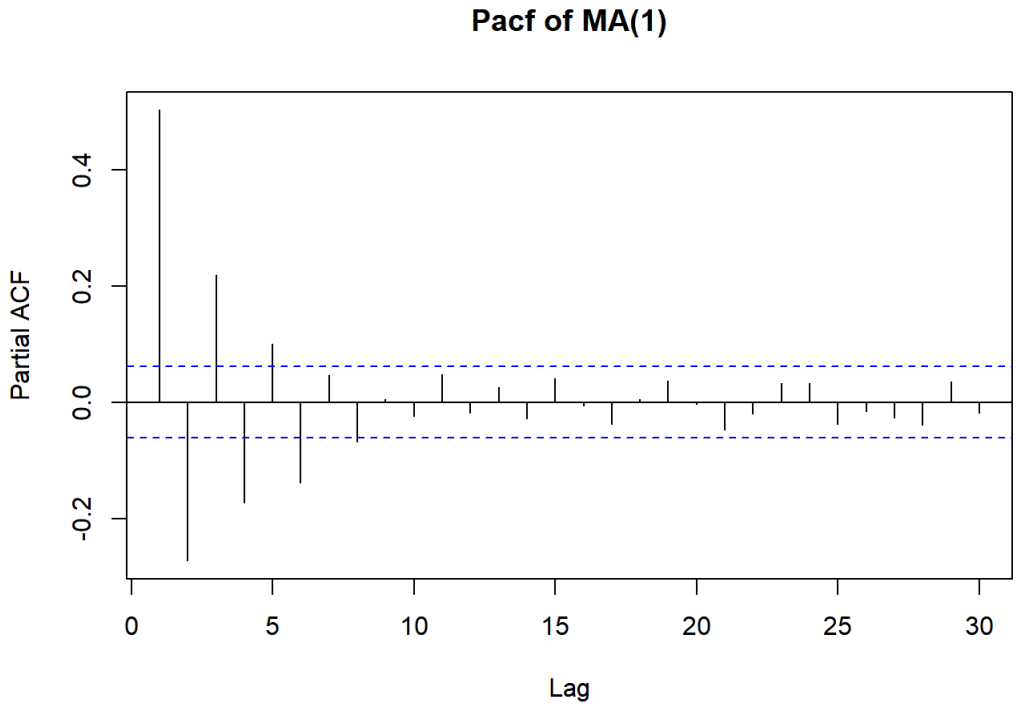
```
MA_1<-arima.sim(model=list(ma=0.8), n=1000)
plot(MA_1, main="MA(1)")
```




```
acf(MA_1,main="Acf of MA(1)")
```



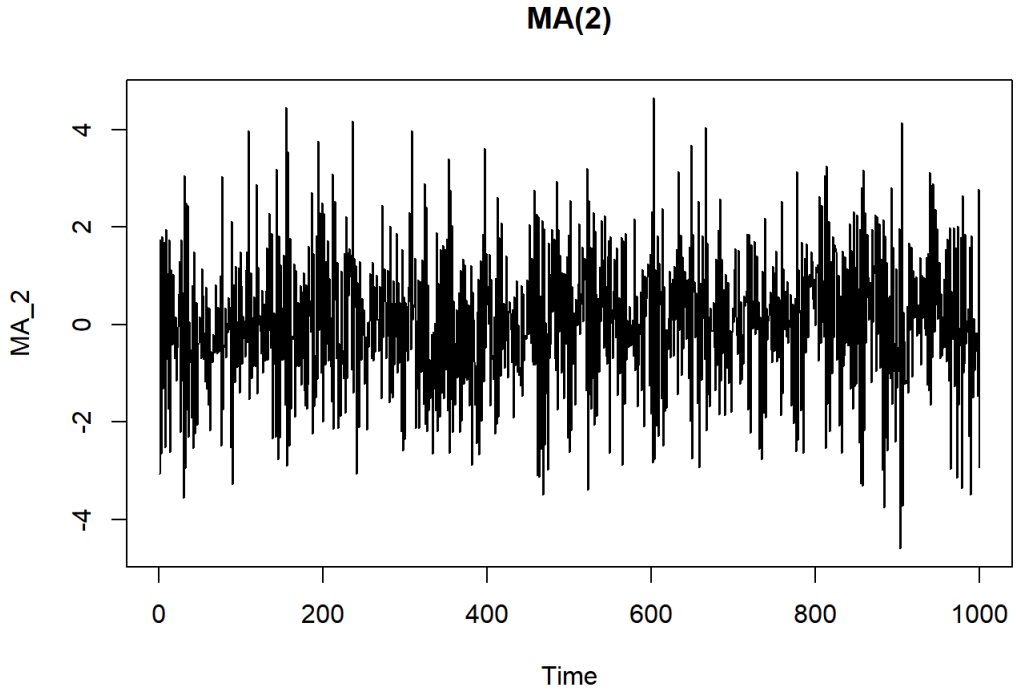

```
pacf(MA_1,main="Pacf of MA(1)")
```



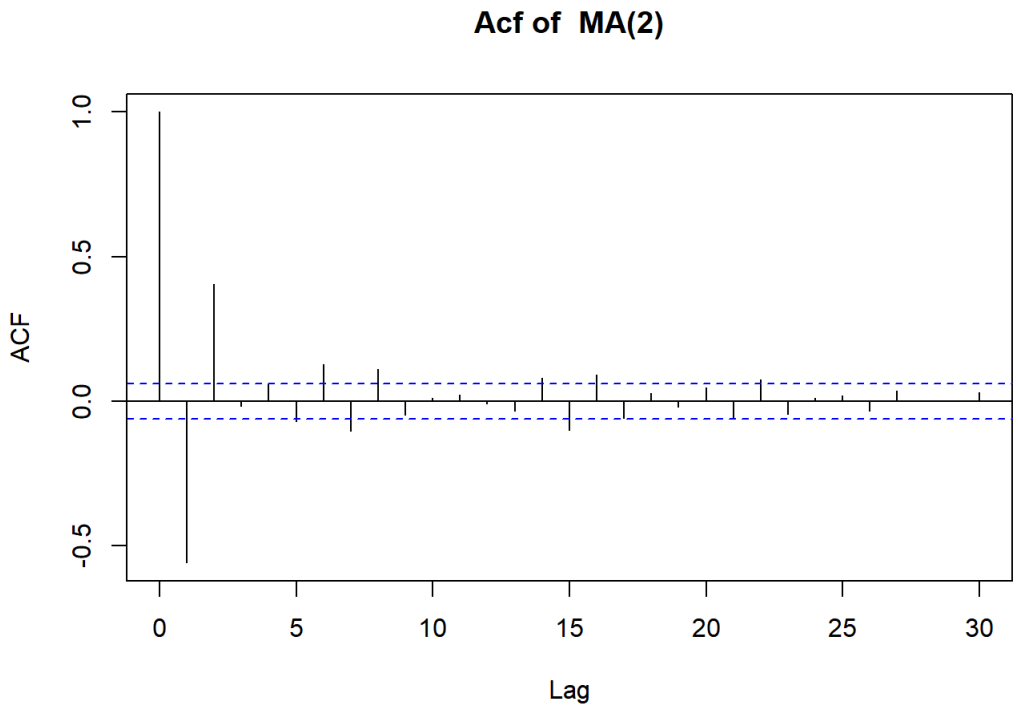
- acf 2 .
- pacf .

MA(2)

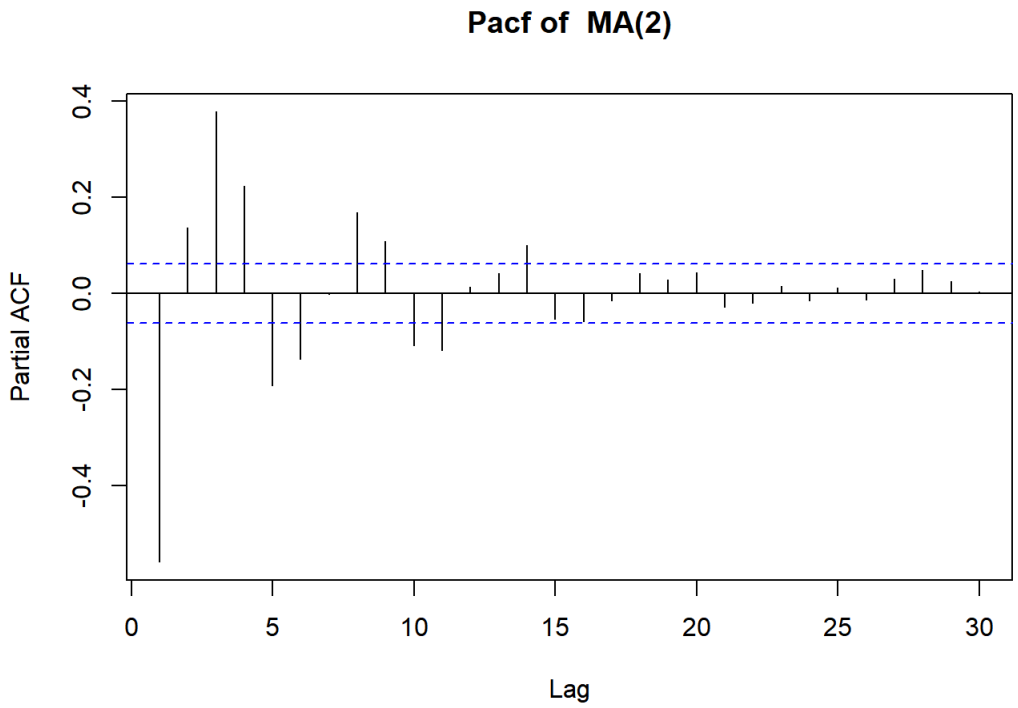
```
MA_2<-arima.sim(model=list(ma=c(-0.6,0.8)),n=1000)
plot(MA_2, main="MA(2)")
```




```
acf(MA_2,main="Acf of  MA(2)")
```




```
pacf(MA_2,main="Pacf of  MA(2)")
```



- acf 3 .
- pacf .

4. ARMA

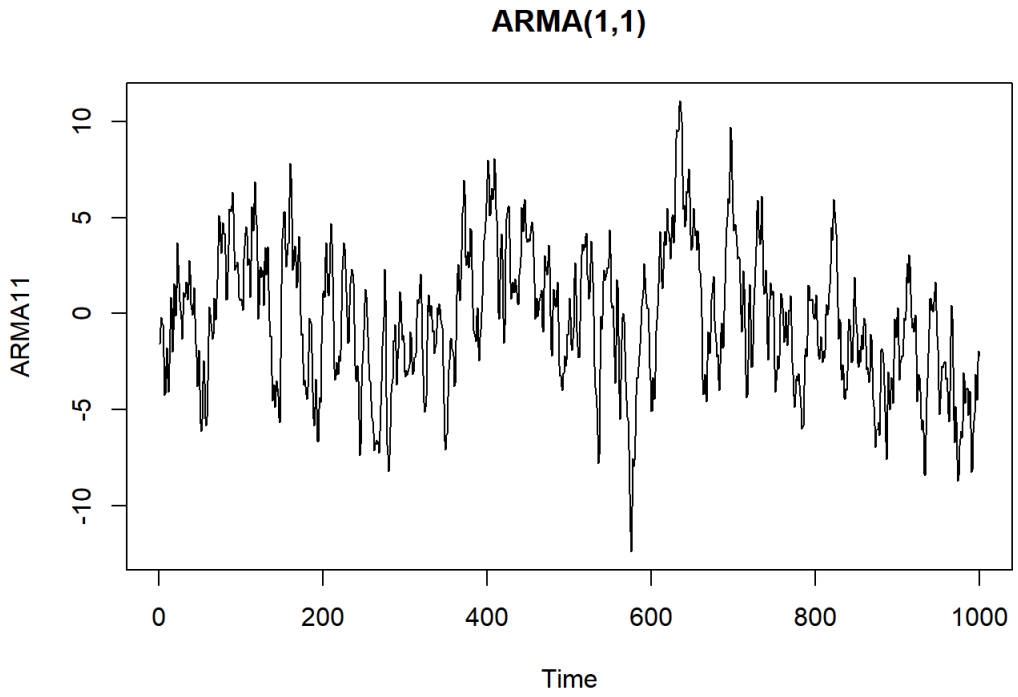
ARMA

`arima.sim()` ARMA .

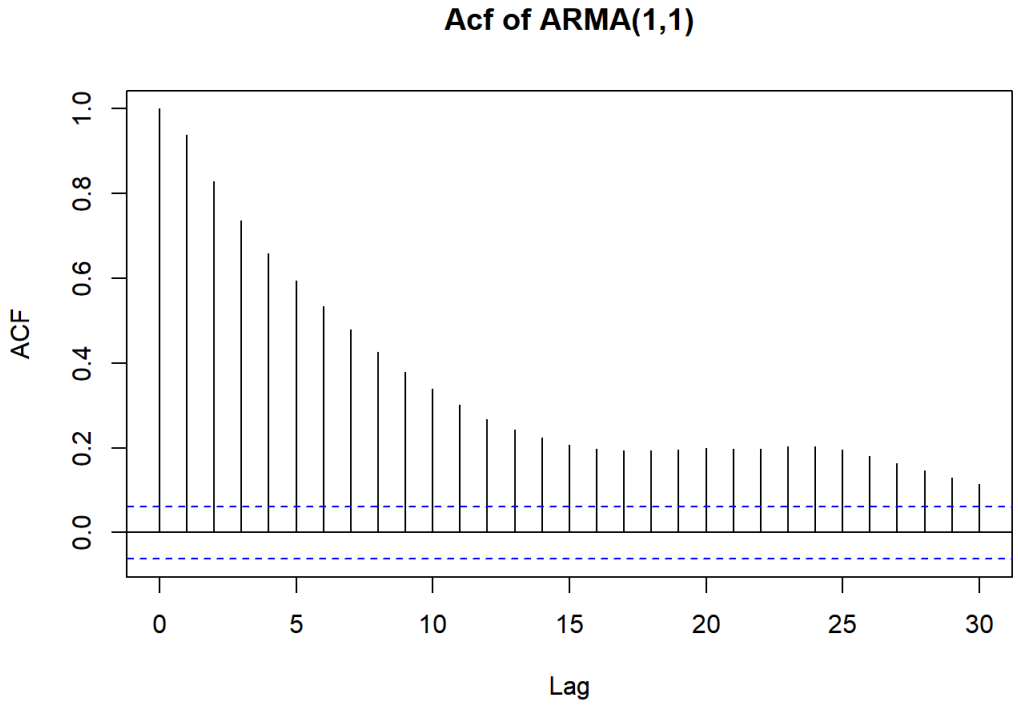
acf pacf

ARMA(1,1)

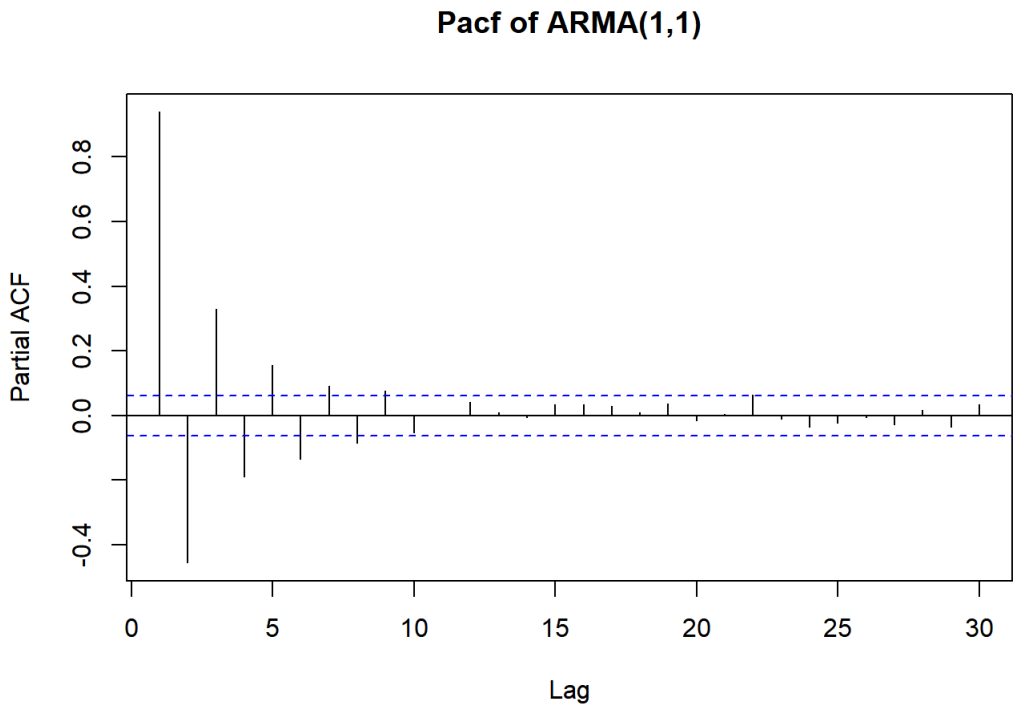
```
ARMA11 <- arima.sim(model = list(order = c(1, 0, 1), ar = 0.9, ma = .8), n = 1000)
plot(ARMA11,main="ARMA(1,1)")
```




```
acf(ARMA11,main="Acf of ARMA(1,1)")
```



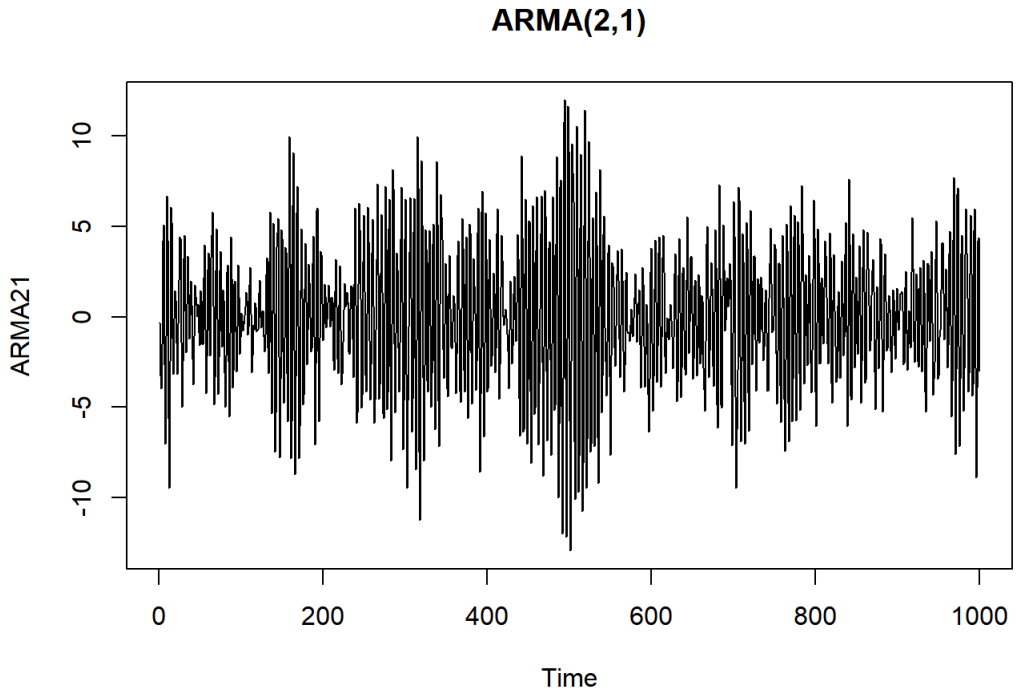

```
pacf(ARMA11,main="Pacf of ARMA(1,1)")
```



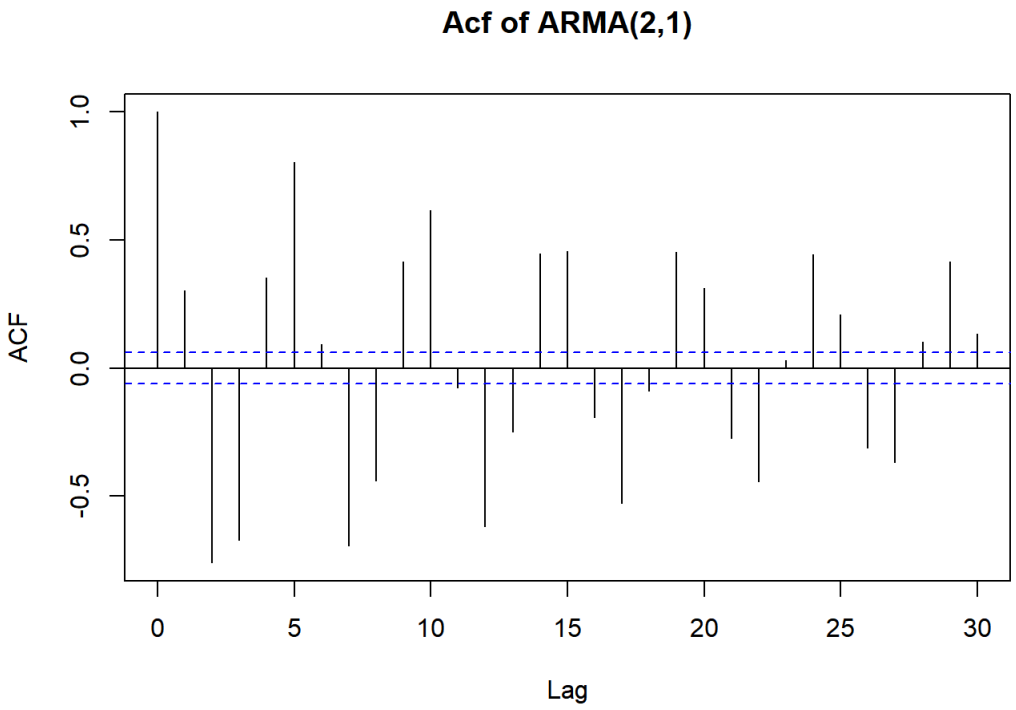
- acf pacf AR, MA
- .

ARMA(2,1)

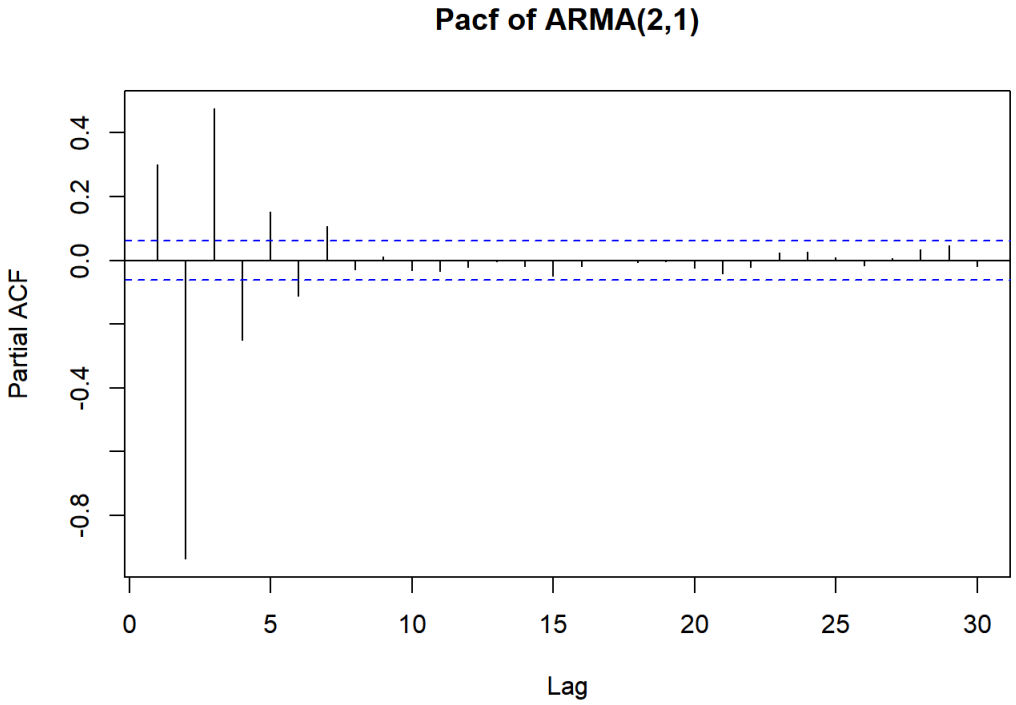
```
ARMA21 <- arima.sim(model = list(order = c(2, 0, 1), ar=c(0.5, -.9), ma= .8), n=1000)
plot(ARMA21, main="ARMA(2,1)")
```




```
acf(ARMA21, main = "Acf of ARMA(2,1)")
```




```
pacf(ARMA21, main = "Pacf of ARMA(2,1)")
```



- acf pacf AR, MA
- .

5.

(1)

```
library(tseries) #for kpss.test()
library(itsmr) #for test()
library(astsa) #for acf2(), sarima.for()
```

```
##
## Attaching package: 'astsa'
```

```
## The following object is masked from 'package:forecast':
##
##     gas
```

```
library(lmtest) #for coeftest()
```

```
## Warning: package 'lmtest' was built under R version 4.0.4
```

```
## Loading required package: zoo
```

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

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

```
library(forecast)
library(TSA) #
```

```
## Warning: package 'TSA' was built under R version 4.0.4
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:itsmr':
##
##     periodogram, season
```

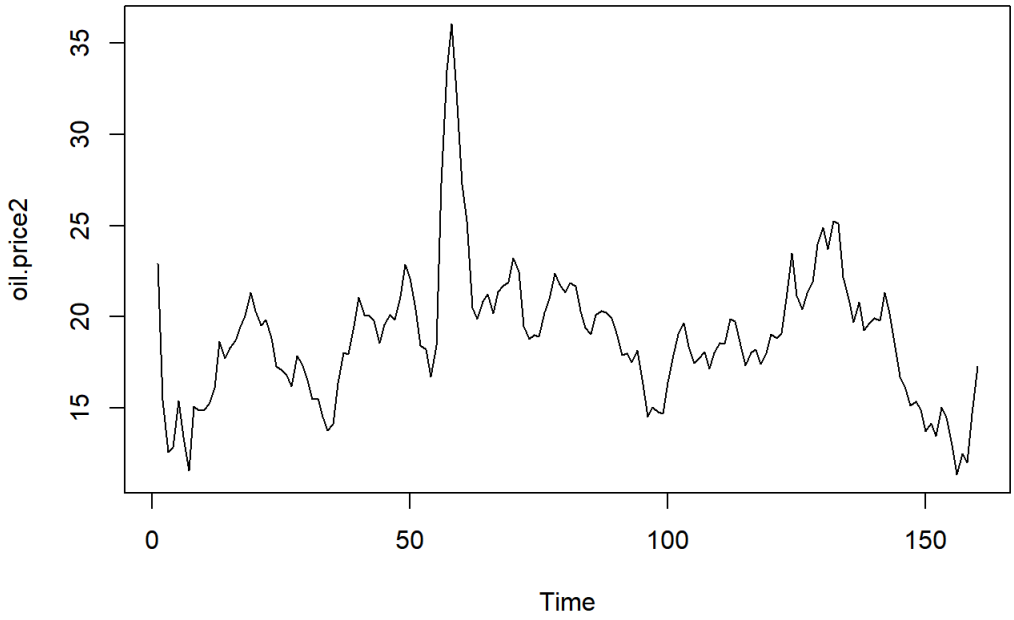
```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

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

```
library(nortest)
```

(2)

```
data(oil.price)
oil.price2 = oil.price[1:160]
ts.plot(oil.price2)
```



(3)

AR, MA, ARMA , .
1 kpss
kpss.test .

```
kpss.test(oil.price2)
```

```
## Warning in kpss.test(oil.price2): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  oil.price2
## KPSS Level = 0.25363, Truncation lag parameter = 4, p-value = 0.1
```

- ->

! 0 , .

```
Box.test(oil.price2, lag=10,type="Ljung-Box")
```

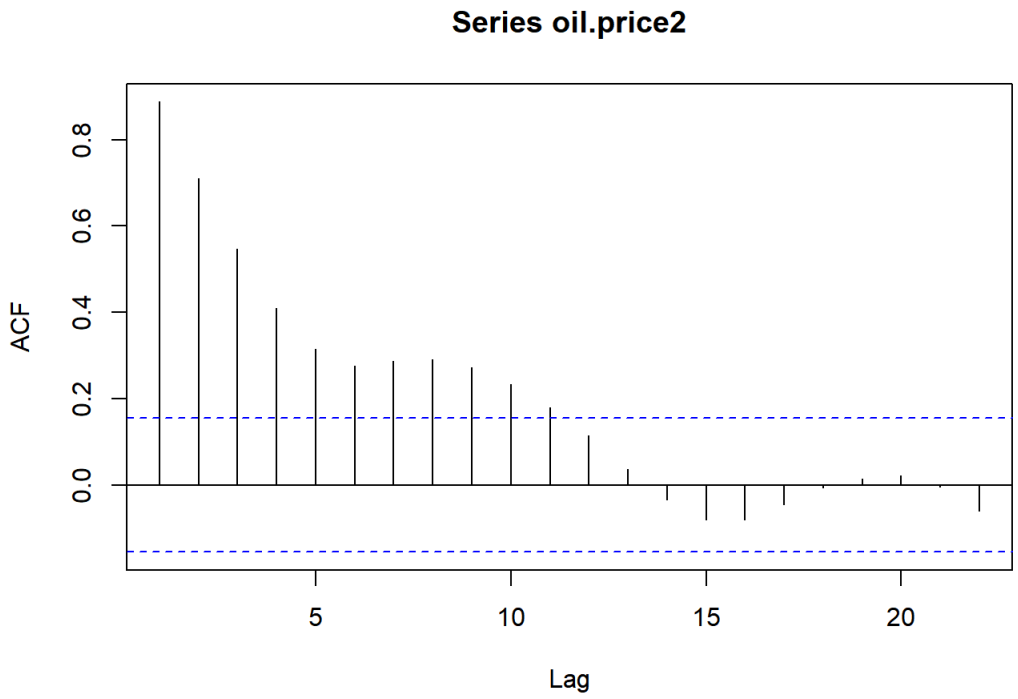
```
##
##  Box-Ljung test
##
## data:  oil.price2
## X-squared = 368.32, df = 10, p-value < 2.2e-16
```

- ->

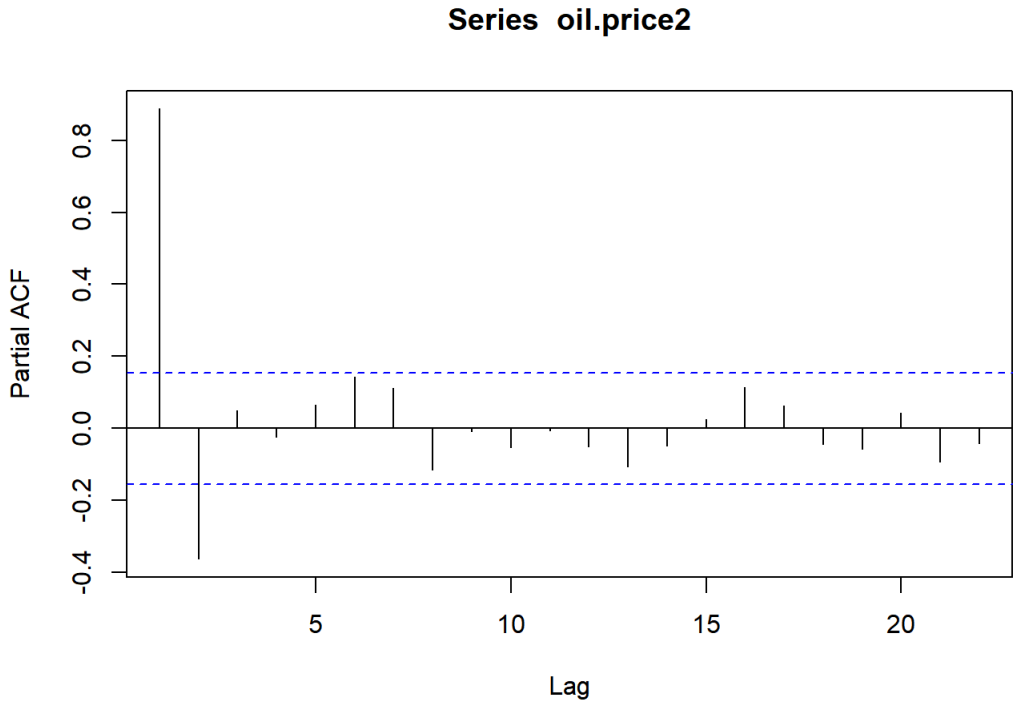
(4)ACF, PACF

, .

```
acf(oil.price2)
```




```
pacf(oil.price2)
```



* acf pcf 3 0 , AR(2)

. * AR(2)

(5)

arima , order .
(p,d,q) , d arima AR, MA d 0 .

```
fit<-arima(oil.price2, order=c(2,0,0))
summary(fit)
```

```
##
## Call:
## arima(x = oil.price2, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       1.3267  -0.4835    19.1463
## s.e.  0.0746   0.0750     0.7397
##
## sigma^2 estimated as 2.205:  log likelihood = -291.35,  aic = 588.71
##
## Training set error measures:
```

```
## Warning in trainingaccuracy(object, test, d, D): test elements must be within
## sample
```

```
##                ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN
```

- : $$Zt=19.1463+1.3267Z_(t−1)−0.4835Z_(t−2)+I_t

(6)

coeftest .

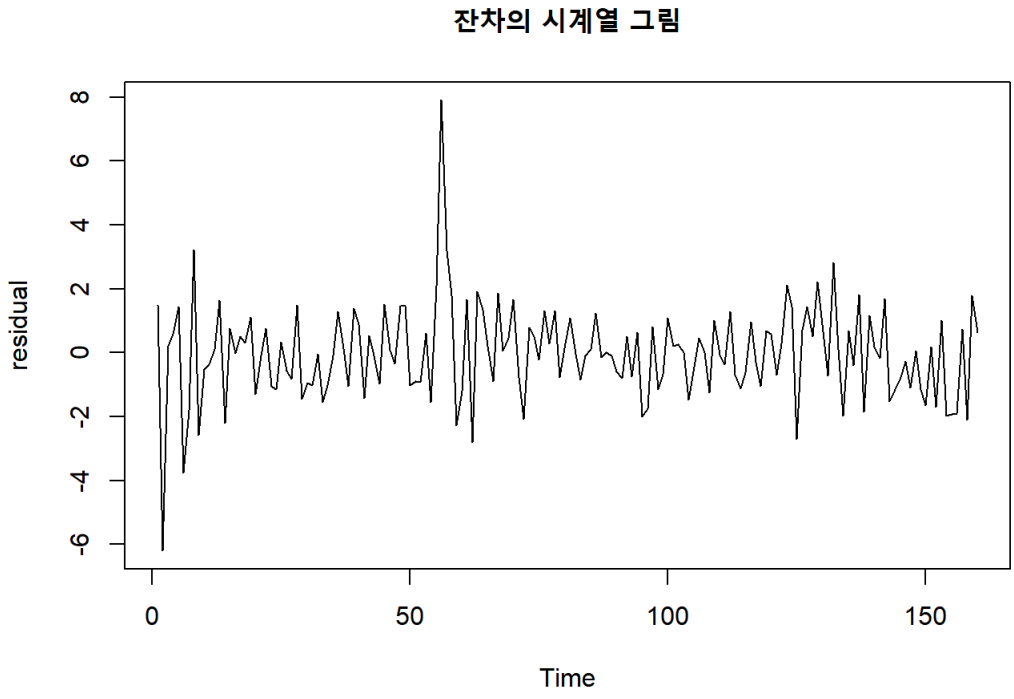
```
coeftest(fit)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar1        1.326711   0.074592 17.7862 < 2.2e-16 ***
## ar2       -0.483522   0.074985 -6.4482 1.132e-10 ***
## intercept 19.146313   0.739656 25.8854 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

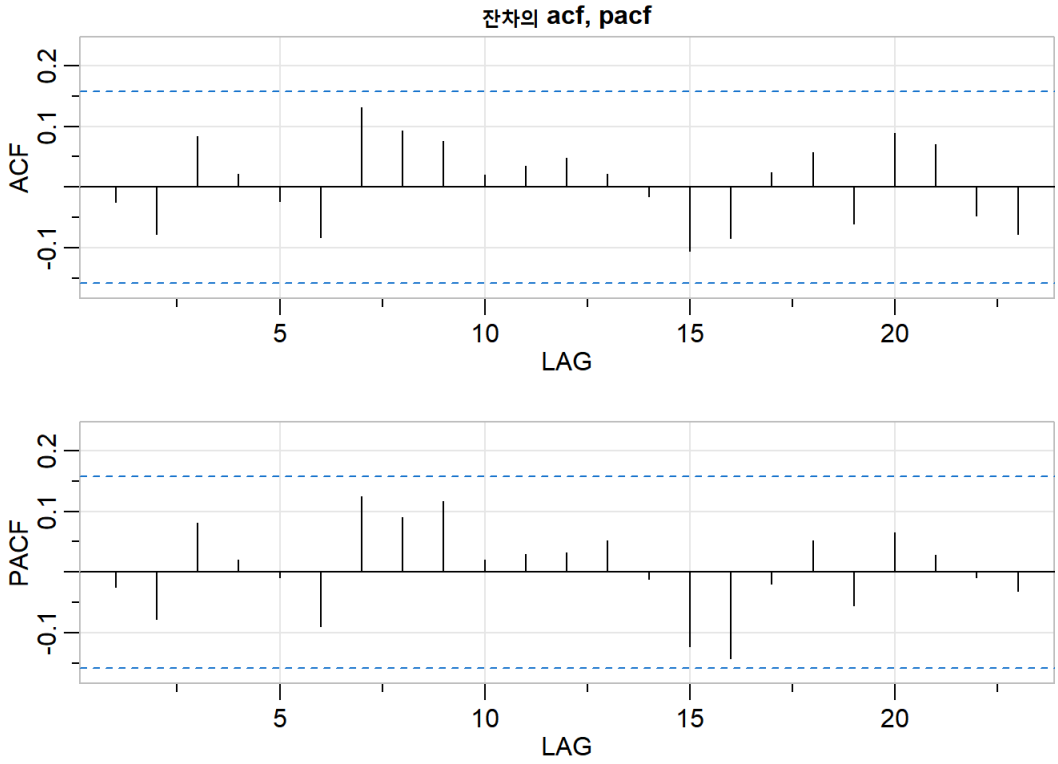
- .

!

```
ts.plot(resid(fit),main="          ", ylab="residual")
```



```
acf2(resid(fit),main="    acf, pacf") #
```




```
##      [,1]  [,2] [,3] [,4]  [,5]  [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.02 -0.08 0.08 0.02 -0.02 -0.08 0.13 0.09 0.08  0.02  0.03  0.05  0.02
## PACF -0.02 -0.08 0.08 0.02 -0.01 -0.09 0.12 0.09 0.12  0.02  0.03  0.03  0.05
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
## ACF  -0.02 -0.11 -0.08  0.02  0.06 -0.06  0.09  0.07 -0.05 -0.08
## PACF -0.01 -0.12 -0.14 -0.02  0.05 -0.06  0.06  0.03 -0.01 -0.03
```

```
Box.test(resid(fit),lag=10,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid(fit)
## X-squared = 9.0404, df = 10, p-value = 0.5283
```

- , . -> .

AR(3), ARMA(2,1) 3

```
fit2<-arima(oil.price2, order=c(3,0,0)) #AR(3)
fit3<-arima(oil.price2, order=c(2,0,1)) #ARMA(2,1)
#AR(3)
summary(fit2)
```

```
##
## Call:
## arima(x = oil.price2, order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2     ar3  intercept
##       1.3667  -0.6061  0.0967    19.1219
## s.e.  0.0822   0.1306  0.0846     0.8059
##
## sigma^2 estimated as 2.187:  log likelihood = -290.7,  aic = 589.41
##
## Training set error measures:
```

```
## Warning in trainingaccuracy(object, test, d, D): test elements must be within
## sample
```

```
##                    ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN
```

```
coeftest(fit2)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1        1.366725   0.082171 16.6326 < 2.2e-16 ***
## ar2       -0.606097   0.130607 -4.6406 3.474e-06 ***
## ar3        0.096749   0.084588  1.1438    0.2527
## intercept 19.121889   0.805924 23.7267 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#ARMA(2,1)
summary(fit3)
```

```
##
## Call:
## arima(x = oil.price2, order = c(2, 0, 1))
##
## Coefficients:
##          ar1      ar2     ma1  intercept
##       1.1107  -0.2899  0.2686    19.1195
## s.e.  0.1974   0.1838  0.2042     0.8122
##
## sigma^2 estimated as 2.181:  log likelihood = -290.49,  aic = 588.98
##
## Training set error measures:
```

```
## Warning in trainingaccuracy(object, test, d, D): test elements must be within
## sample
```

```
##                ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN
```

```
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1         1.11072    0.19741  5.6266 1.838e-08 ***
## ar2        -0.28991    0.18378 -1.5775    0.1147
## ma1         0.26864    0.20418  1.3157    0.1883
## intercept 19.11950    0.81221 23.5401 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- AR(3), ARMA(2,1) .
- AIC AR(2) .
- AR(2) !!

6. 2

-> `auto.arima()` ->
`auto.arima()` AIC, BIC .

```
auto.arima(oil.price2)
```

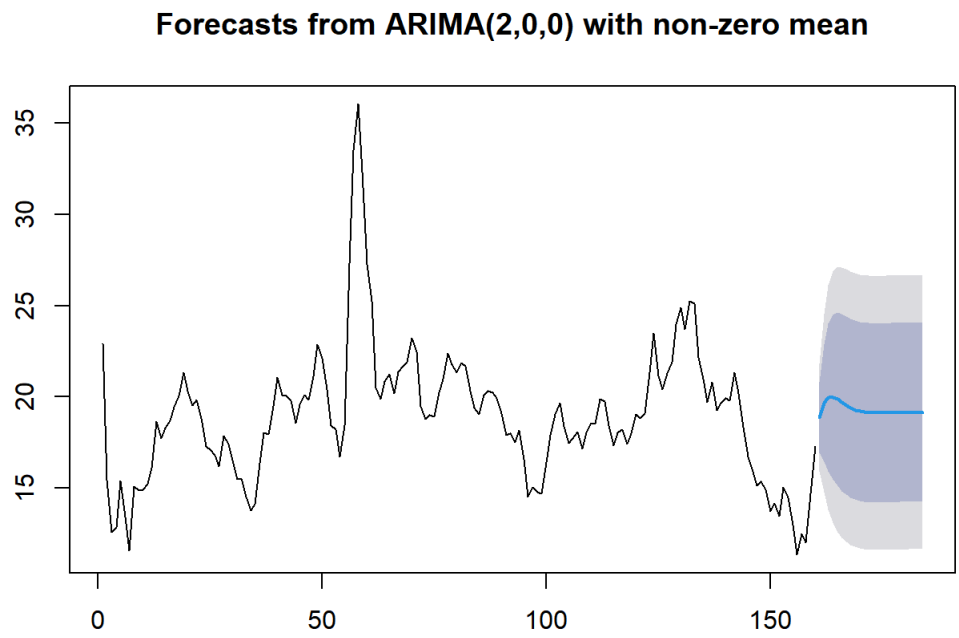
```
## Series: oil.price2
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2     mean
##       1.3267  -0.4835  19.1463
## s.e.  0.0746   0.0750   0.7397
##
## sigma^2 estimated as 2.247:  log likelihood=-291.35
## AIC=590.71   AICc=590.97   BIC=603.01
```

- AR(2) .

```
fit <- Arima(oil.price2, c(2,0,0)) ##
fit
```

```
## Series: oil.price2
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2     mean
##       1.3267  -0.4835  19.1463
## s.e.  0.0746   0.0750   0.7397
##
## sigma^2 estimated as 2.247:  log likelihood=-291.35
## AIC=590.71   AICc=590.97   BIC=603.01
```

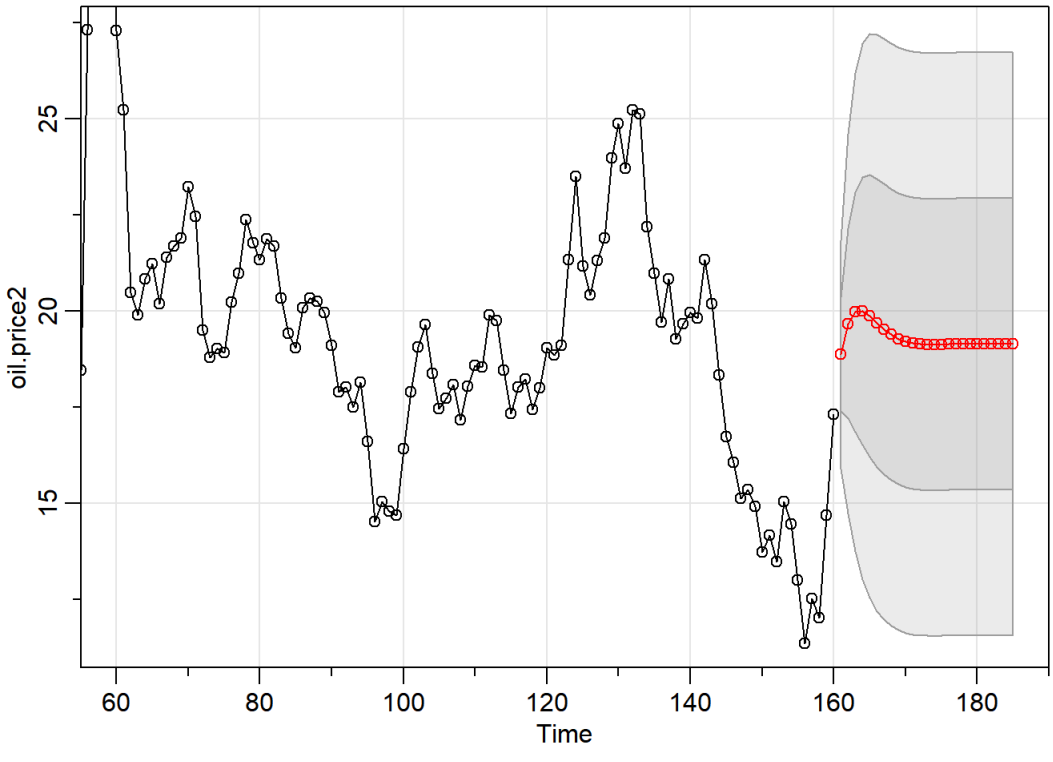
```
plot(forecast(fit,h=24))
```



sarima.for(data n.ahead, p, d, q) !

n.ahead .

```
sarima.for(oil.price2, 25, 2,0,0)
```



```
## $pred
## Time Series:
## Start = 161
## End = 185
## Frequency = 1
##  [1] 18.86962 19.66712 19.97106 19.98869 19.86512 19.69266 19.52359 19.38269
##  [9] 19.27749 19.20605 19.16214 19.13843 19.12820 19.12610 19.12825 19.13212
## [17] 19.13622 19.13978 19.14253 19.14445 19.14567 19.14636 19.14669 19.14679
## [25] 19.14676
##
## $se
## Time Series:
## Start = 161
## End = 185
## Frequency = 1
##  [1] 1.484881 2.466943 3.111163 3.481463 3.668470 3.750192 3.780017 3.788425
##  [9] 3.789894 3.789937 3.790034 3.790298 3.790579 3.790786 3.790907 3.790965
## [17] 3.790988 3.790995 3.790997 3.790997 3.790997 3.790997 3.790997 3.790997
## [25] 3.790998
```

7. 3

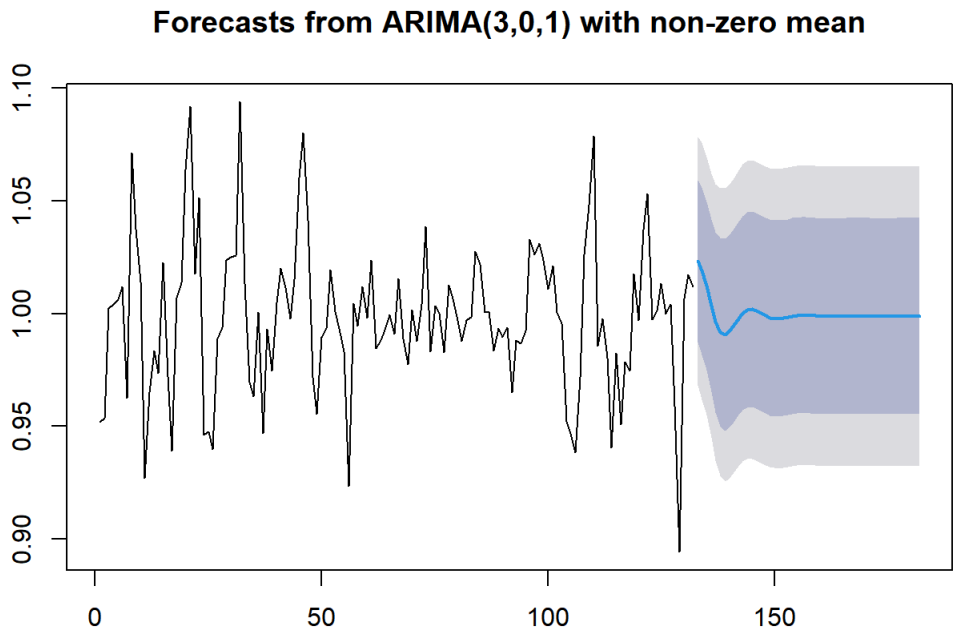
1 AirPassengers

```
data <- AirPassengers
dc.m <- decompose(data, type="multiplicative")
dat <- dc.m$random[complete.cases(dc.m$random)]
auto.arima(dat)
```

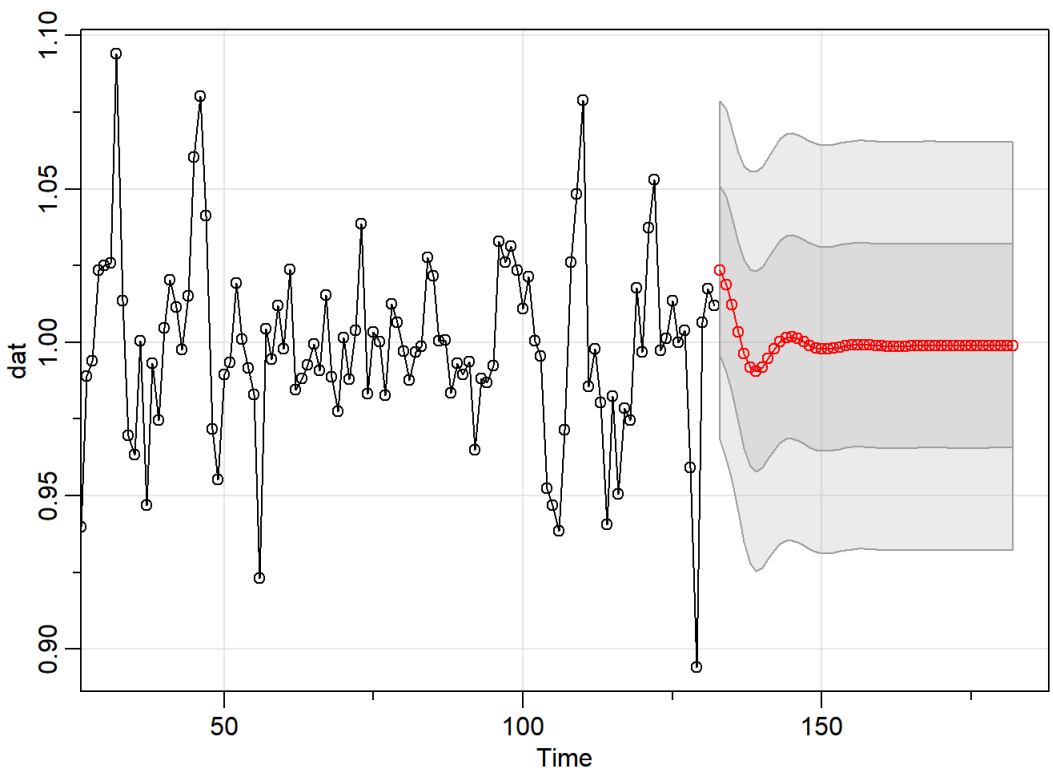
```
## Series: dat
## ARIMA(3,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3      ma1     mean
##       1.1189  -0.2467  -0.2238  -0.8513  0.9989
## s.e.  0.1012   0.1322   0.0915   0.0668  0.0011
##
## sigma^2 estimated as 0.0007906:  log likelihood=286.1
## AIC=-560.2   AICc=-559.53   BIC=-542.91
```

```
fit <- Arima(dat,order=c(3,0,1))
```

```
plot(forecast(fit, h=50))
```




```
sarima.for(dat, 50, 3,0,1)
```




```
## $pred
## Time Series:
## Start = 133
## End = 182
## Frequency = 1
##  [1] 1.0233897 1.0189050 1.0122793 1.0034404 0.9961890 0.9917391 0.9905276
##  [8] 0.9918928 0.9947152 0.9978076 1.0002657 1.0016215 1.0018398 1.0011995
## [15] 1.0001257 0.9990333 0.9982193 0.9978183 0.9978151 0.9980925 0.9984935
## [22] 0.9988744 0.9991396 0.9992526 0.9992283 0.9991140 0.9989667 0.9988356
## [29] 0.9987508 0.9987212 0.9987384 0.9987839 0.9988372 0.9988818 0.9989083
## [36] 0.9989150 0.9989061 0.9988884 0.9988694 0.9988545 0.9988464 0.9988453
## [43] 0.9988494 0.9988561 0.9988628 0.9988678 0.9988701 0.9988701 0.9988683
## [50] 0.9988658
##
## $se
## Time Series:
## Start = 133
## End = 182
## Frequency = 1
##  [1] 0.02757917 0.02854917 0.02858611 0.02928679 0.03067836 0.03196428
##  [7] 0.03259151 0.03271684 0.03271902 0.03281844 0.03299443 0.03313914
## [13] 0.03320252 0.03321143 0.03321315 0.03322900 0.03325203 0.03326879
## [19] 0.03327506 0.03327558 0.03327611 0.03327852 0.03328148 0.03328338
## [25] 0.03328397 0.03328399 0.03328411 0.03328446 0.03328483 0.03328504
## [31] 0.03328509 0.03328509 0.03328511 0.03328516 0.03328521 0.03328523
## [37] 0.03328523 0.03328523 0.03328524 0.03328525 0.03328525 0.03328525
## [43] 0.03328525 0.03328525 0.03328525 0.03328525 0.03328526 0.03328526
## [49] 0.03328526 0.03328526
```