HuYuDataInsight LLC

Zhaowei Cai

2024-05-09

(a)

```
library(urca)
```

```
## Warning: package 'urca' was built under R version 4.2.3
```

```
library(vars)
```

```
## Warning: package 'vars' was built under R version 4.2.3

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.2.3

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.2.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.3

##
## Attaching package: 'zoo'

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

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.2.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.2.3
```

```r
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 4.2.3

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.2.3

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 4.2.3

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

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```

```r
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

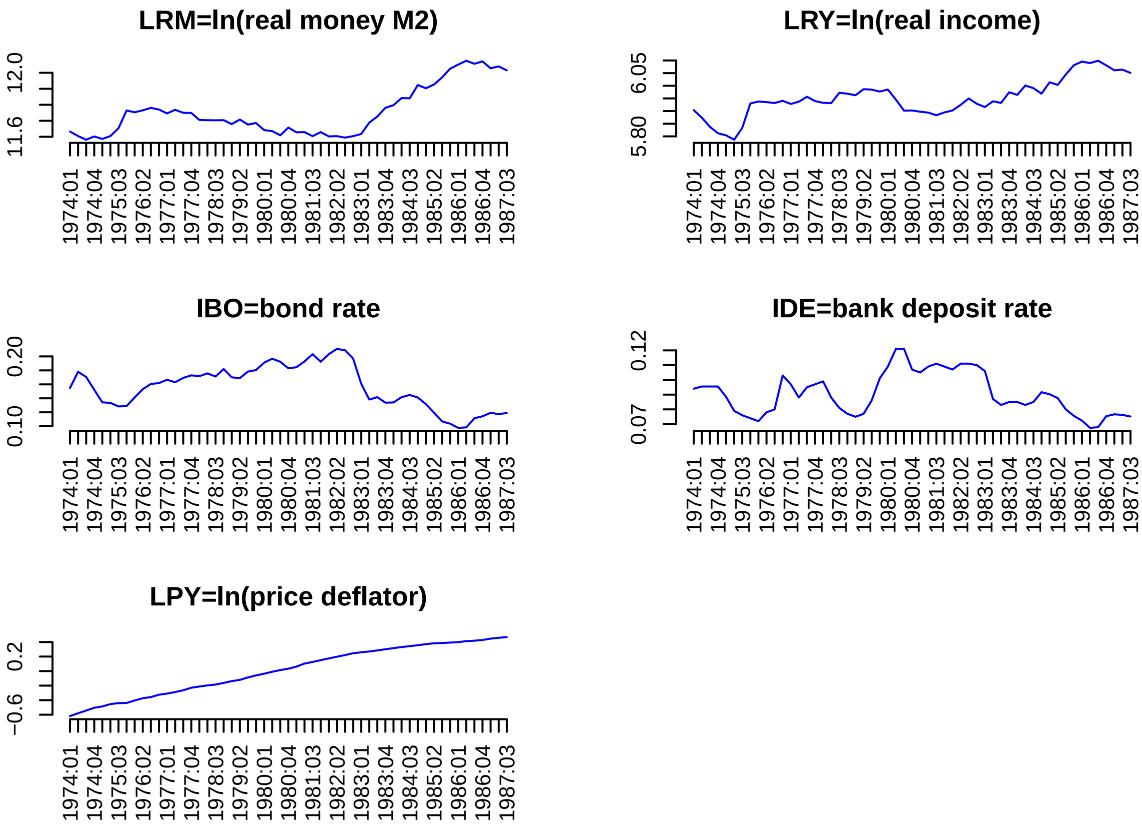
```r
library(fGarch)
```

```
## Warning: package 'fGarch' was built under R version 4.2.3

## NOTE: Packages 'fBasics', 'timeDate', and 'timeSeries' are no longer
## attached to the search() path when 'fGarch' is attached.
##
## If needed attach them yourself in your R script by e.g.,
##         require("timeSeries")

##
## Attaching package: 'fGarch'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```r
library(stargazer)
```

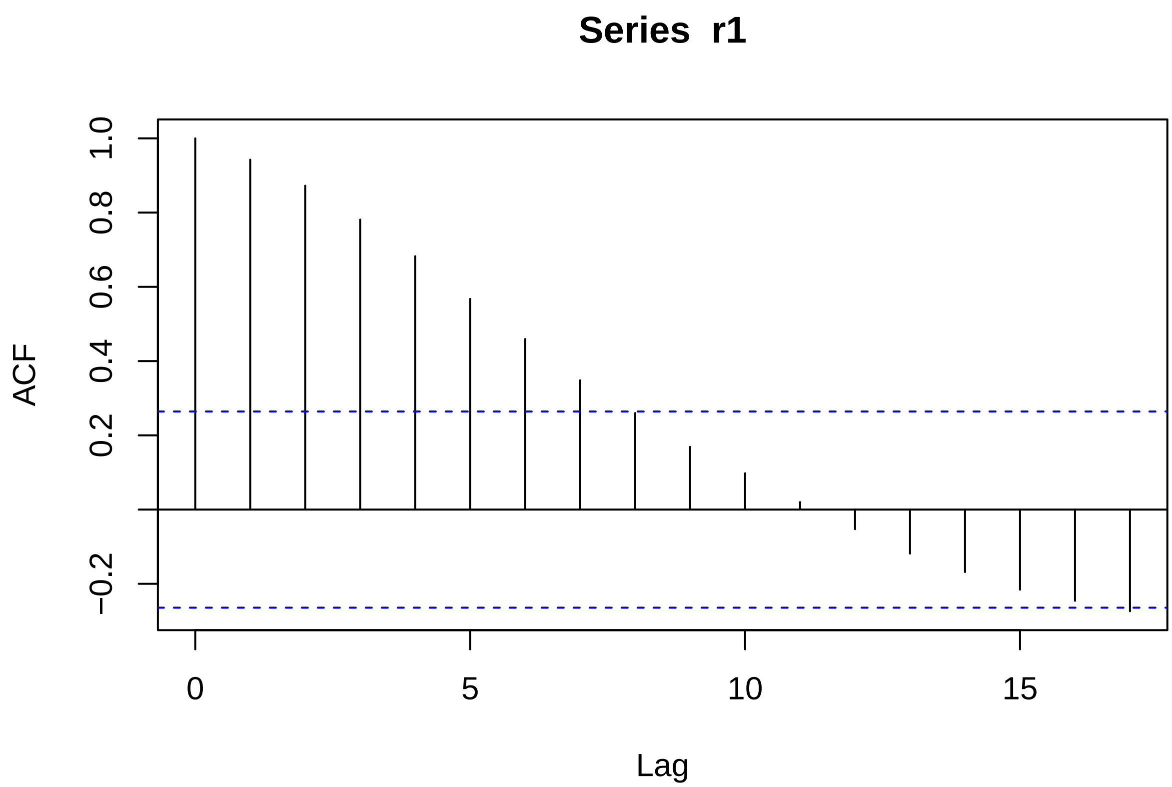
```
##
## Please cite as:

##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
data(denmark)
df.lev = denmark[,c('LRM','LRY','IBO','IDE','LPY')]
m.lev  = as.matrix(df.lev)
nr_lev = nrow(df.lev)

# quarterly centered dummy variables
dum_season = data.frame(yyyymm = denmark$ENTRY)
substr.q = as.numeric(substring(denmark$ENTRY, 6,7))
dum_season$Q2 = (substr.q==2)-1/4
dum_season$Q3 = (substr.q==3)-1/4
dum_season$Q4 = (substr.q==4)-1/4
dum_season = dum_season[,-1]

# visualization
str.main = c(
    'LRM=ln(real money M2)', 'LRY=ln(real income)',
    'IBO=bond rate', 'IDE=bank deposit rate', 'LPY=ln(price deflator)')
par(mfrow=c(3,2), mar=c(5,3,3,3))
for(i in 1:5) {
    matplot(m.lev[,i], axes=FALSE,
        type=c('l'), col = c('blue'),
        main = str.main[i]) # plot the columns of one matrix m.lev
    axis(2) # y axis
    axis(1, at=seq_along(1:nrow(df.lev)),
         labels=denmark$ENTRY, las=2)
}
```

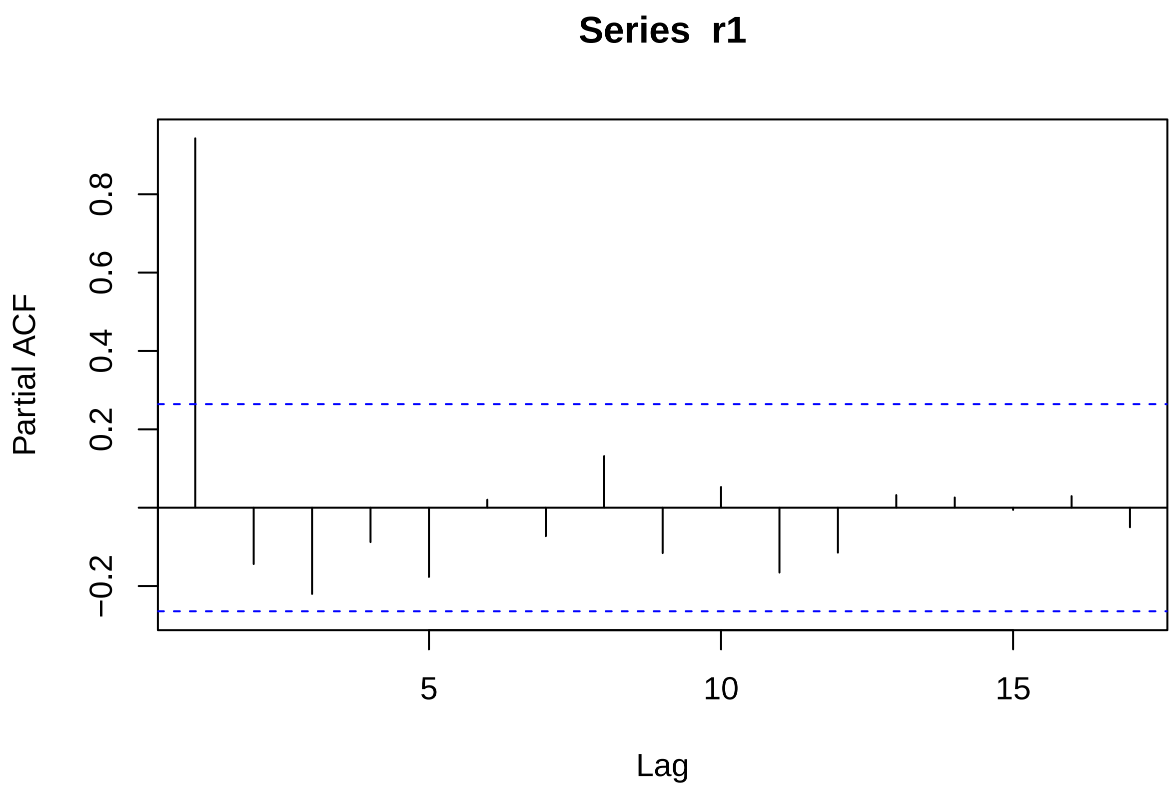


(b)

```r
##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
r1 = df.lev$LRM
acf(r1)
```




```
pacf(r1)
```




```
adf.test(r1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  r1
## Dickey-Fuller = -1.4882, Lag order = 3, p-value = 0.7808
## alternative hypothesis: stationary
```

```
summary(ur.df(r1, type='trend', lags=20, selectlags="BIC"))
```

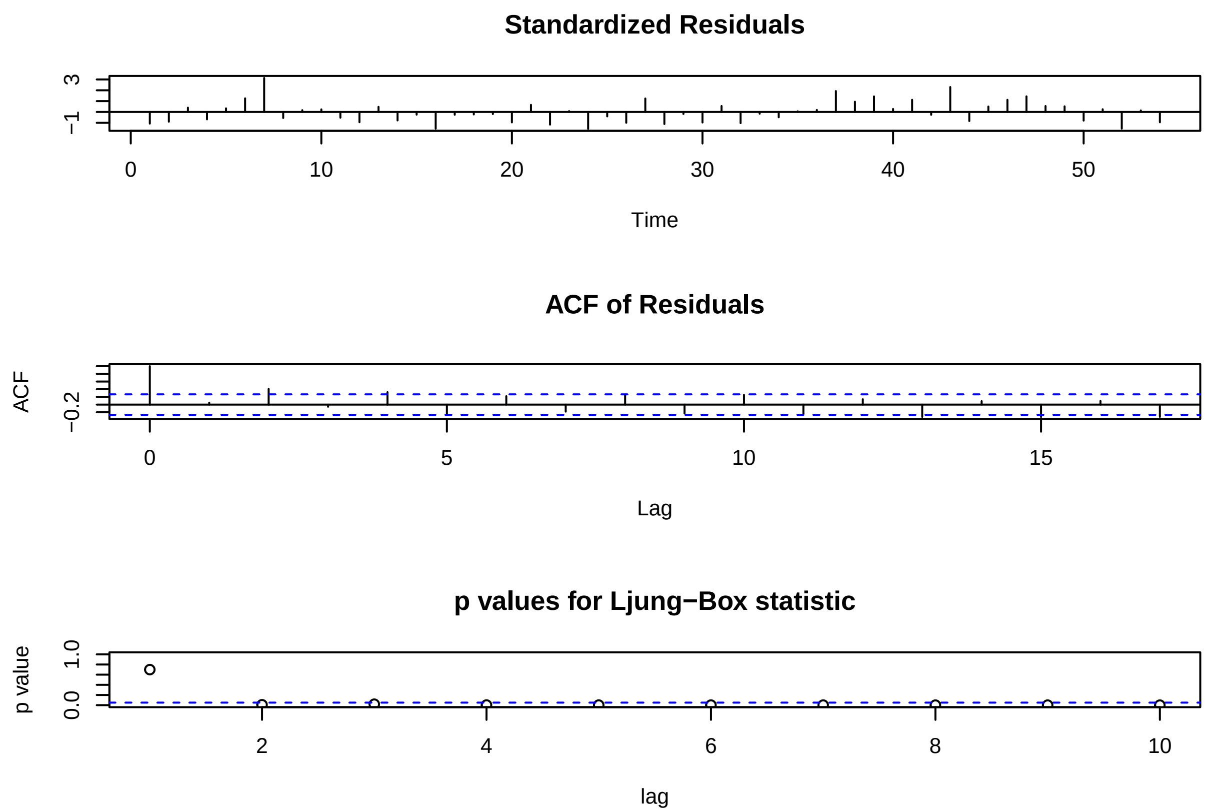
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.034213 -0.016241 -0.002672  0.005154  0.053331
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.2594478  0.5962276   2.112 0.043386 *
## z.lag.1     -0.1121226  0.0531447  -2.110 0.043627 *
## tt           0.0017624  0.0009559   1.844 0.075477 .
## z.diff.lag1 -0.0225423  0.1478966  -0.152 0.879912
## z.diff.lag2  0.5741102  0.1482740   3.872 0.000566 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02656 on 29 degrees of freedom
## Multiple R-squared:  0.4315, Adjusted R-squared:  0.3531
## F-statistic: 5.504 on 4 and 29 DF,  p-value: 0.002013
##
##
## Value of test-statistic is: -2.1098 1.7633 2.2334
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

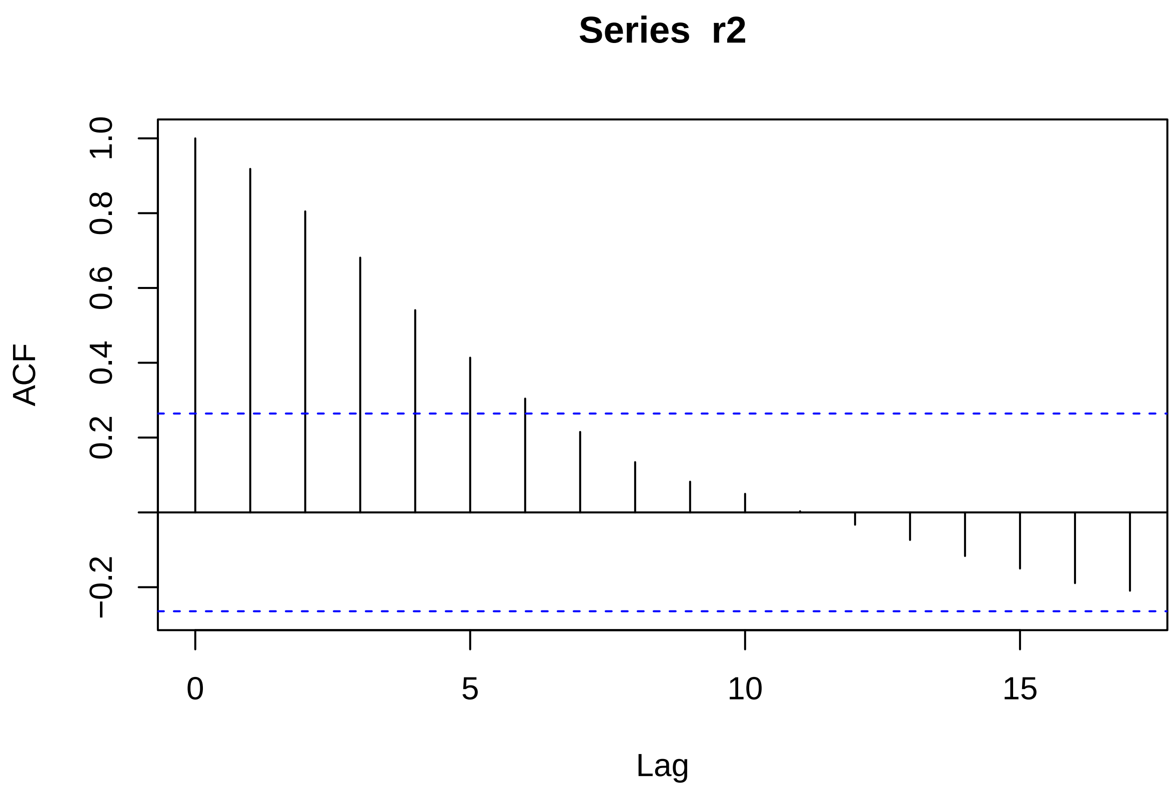
```
# drift+difference
diffr1 = na.omit(diff(r1))
fit1 = auto.arima(diffr1)
summary(fit1)
```

```
## Series: diffr1
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.0071
## s.e.  0.0045
##
## sigma^2 = 0.001098:  log likelihood = 107.86
## AIC=-211.72   AICc=-211.49   BIC=-207.75
##
## Training set error measures:
##                        ME       RMSE        MAE     MPE     MAPE      MASE
## Training set 1.301043e-18 0.03283106 0.02618595 117.875 186.2099 0.6931985
##                    ACF1
## Training set 0.05113235
```

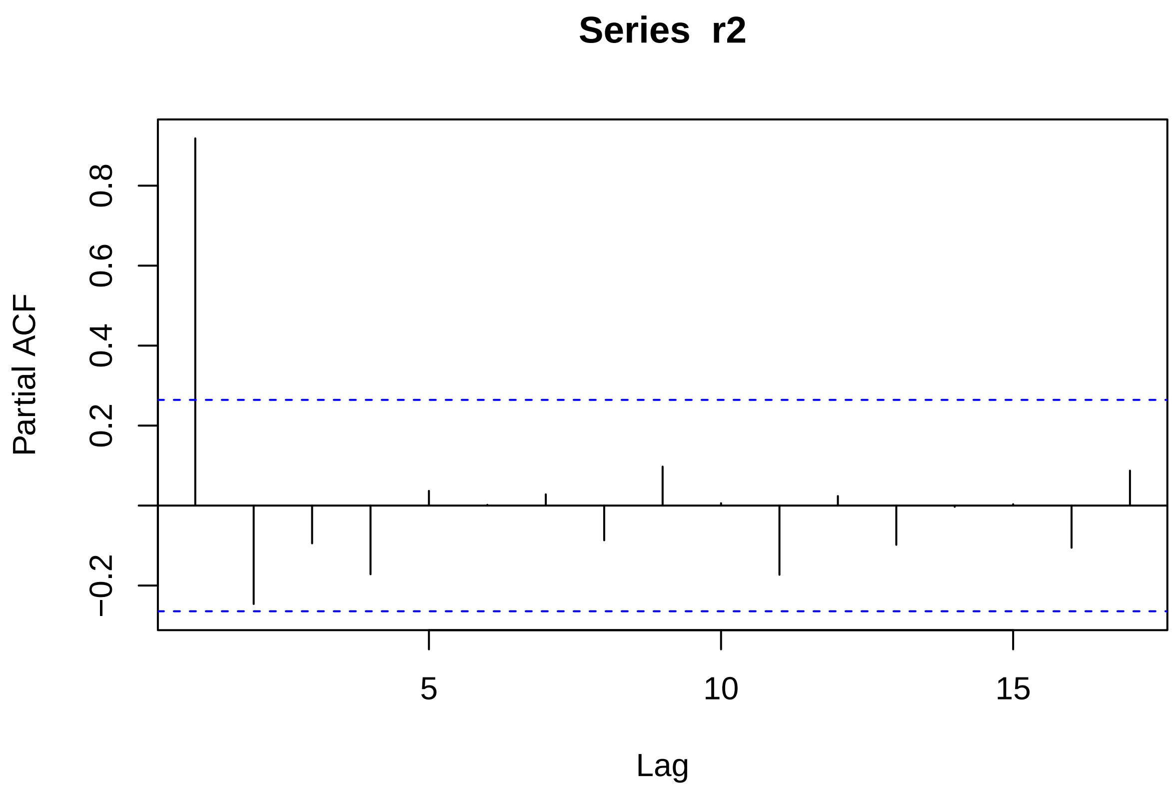
```
tsdiag(fit1)
```




```
##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
r2 = df.lev$LRY
acf(r2)
```




```
pacf(r2)
```




```
adf.test(r2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  r2
## Dickey-Fuller = -2.6695, Lag order = 3, p-value = 0.3048
## alternative hypothesis: stationary
```

```
summary(ur.df(r2, type='trend', lags=20, selectlags="BIC"))
```

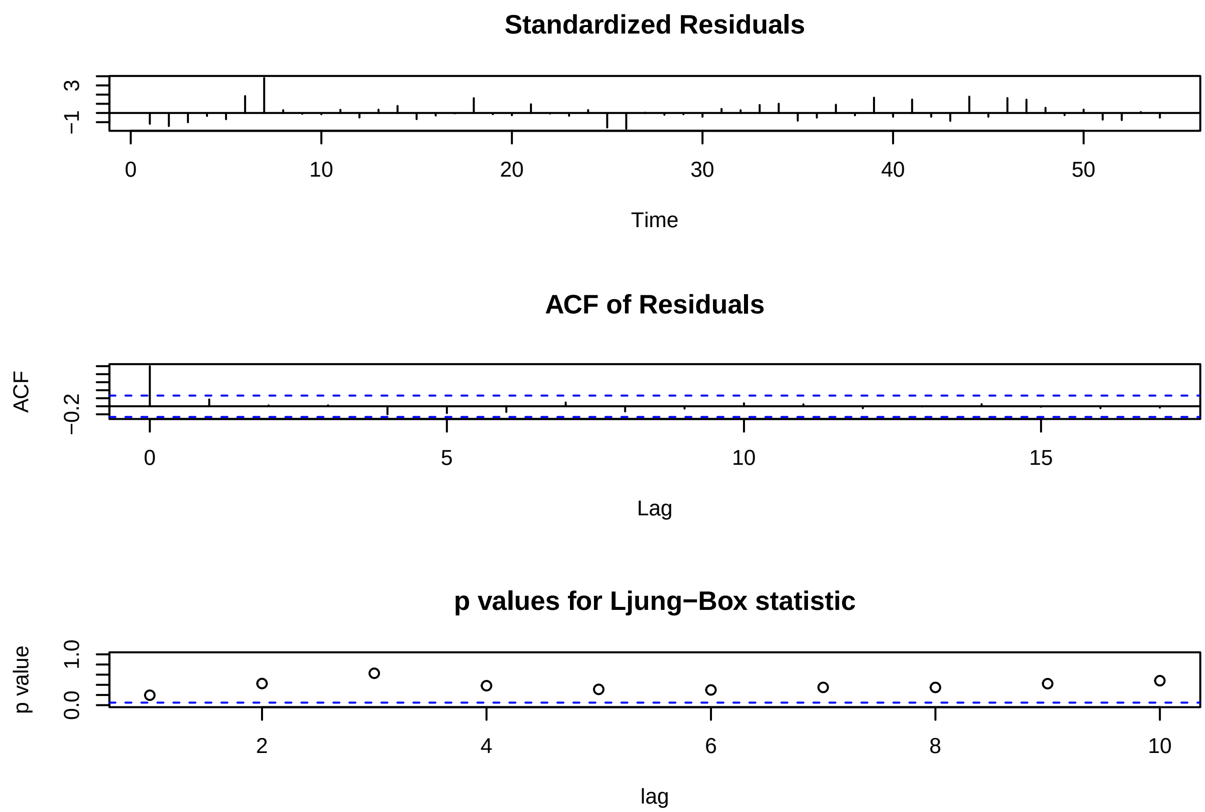
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.038854 -0.016514 -0.002254  0.015310  0.035822
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8636001  0.4964316   1.740   0.0922 .
## z.lag.1     -0.1503337  0.0855157  -1.758   0.0890 .
## tt           0.0010023  0.0005623   1.783   0.0848 .
## z.diff.lag   0.0154460  0.1786783   0.086   0.9317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02238 on 30 degrees of freedom
## Multiple R-squared:  0.1092, Adjusted R-squared:  0.02015
## F-statistic: 1.226 on 3 and 30 DF,  p-value: 0.3173
##
##
## Value of test-statistic is: -1.758 1.379 1.8368
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

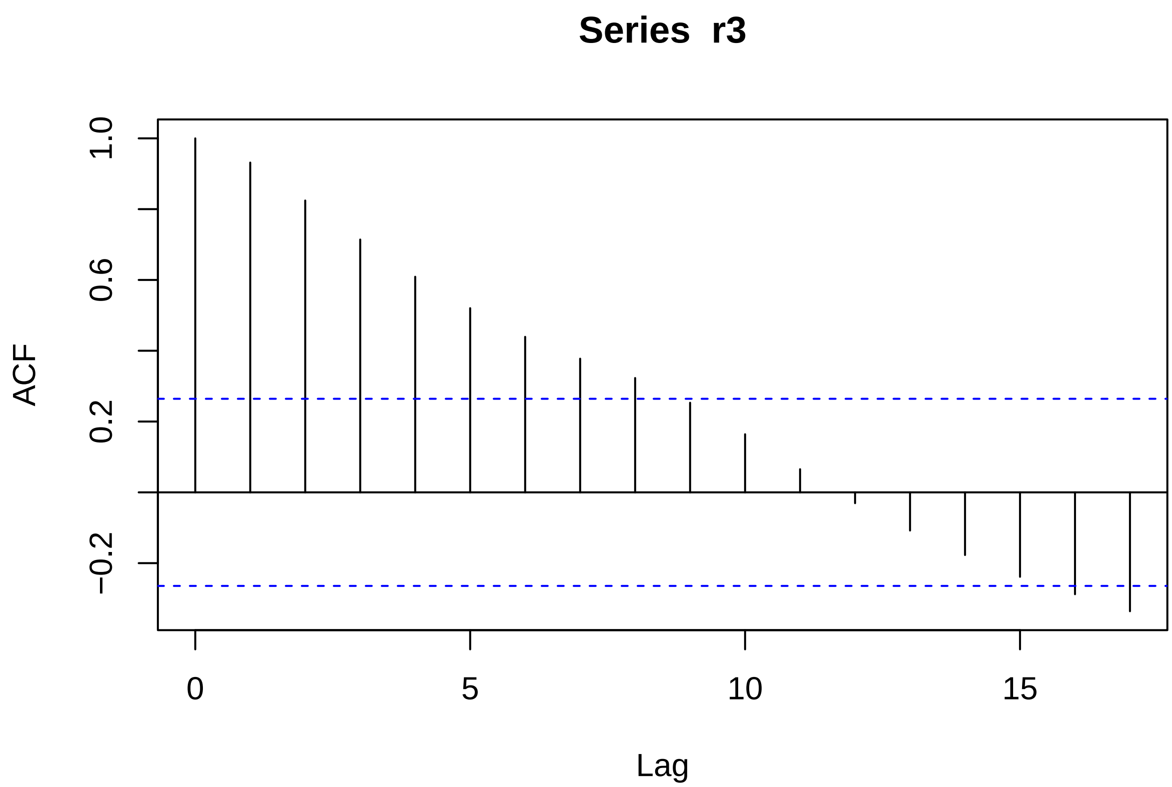
```
# difference
diffr2 = na.omit(diff(r2))
fit2 = auto.arima(diffr2)
summary(fit2)
```

```
## Series: diffr2
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 0.0006409:  log likelihood = 121.9
## AIC=-241.8   AICc=-241.72   BIC=-239.81
##
## Training set error measures:
##                       ME       RMSE        MAE MPE MAPE      MASE      ACF1
## Training set 0.002725399 0.02531514 0.01874242 100  100 0.7498318 0.1714808
```

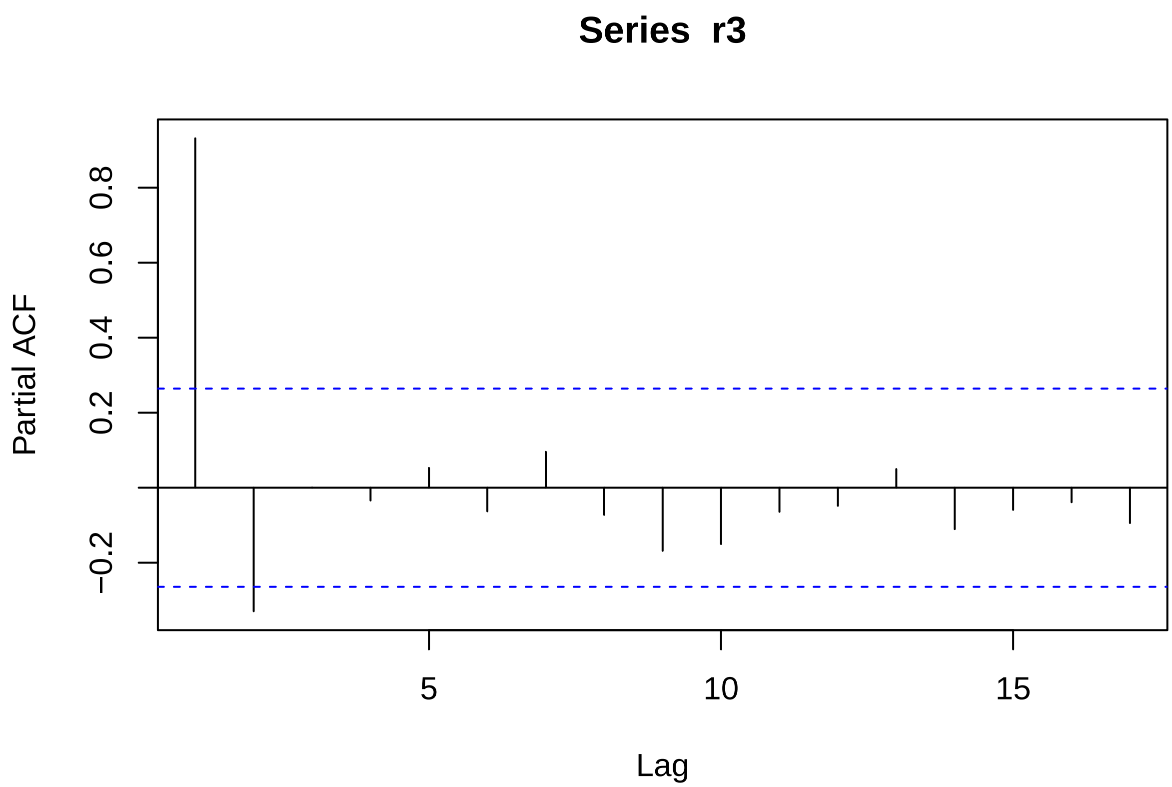
```
tsdiag(fit2)
```




```
##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
r3 = df.lev$IBO
acf(r3)
```




```
pacf(r3)
```




```
adf.test(r3)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  r3
## Dickey-Fuller = -1.7639, Lag order = 3, p-value = 0.6697
## alternative hypothesis: stationary
```

```
summary(ur.df(r3, type='trend', lags=20, selectlags="BIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0214655 -0.0061794 -0.0003114  0.0056962  0.0209332
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0621476  0.0207509   2.995  0.00546 **
## z.lag.1     -0.2216932  0.0738704  -3.001  0.00537 **
## tt          -0.0007479  0.0002678  -2.793  0.00901 **
## z.diff.lag   0.5113810  0.1459334   3.504  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008531 on 30 degrees of freedom
## Multiple R-squared:  0.3773, Adjusted R-squared:  0.315
## F-statistic:  6.06 on 3 and 30 DF,  p-value: 0.002368
##
##
## Value of test-statistic is: -3.0011 3.1736 4.6509
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
# drift+tt+difference
diffr3 = na.omit(diff(r3))
summary(ur.df(diffr3, type='trend', lags=20, selectlags="BIC"))
```

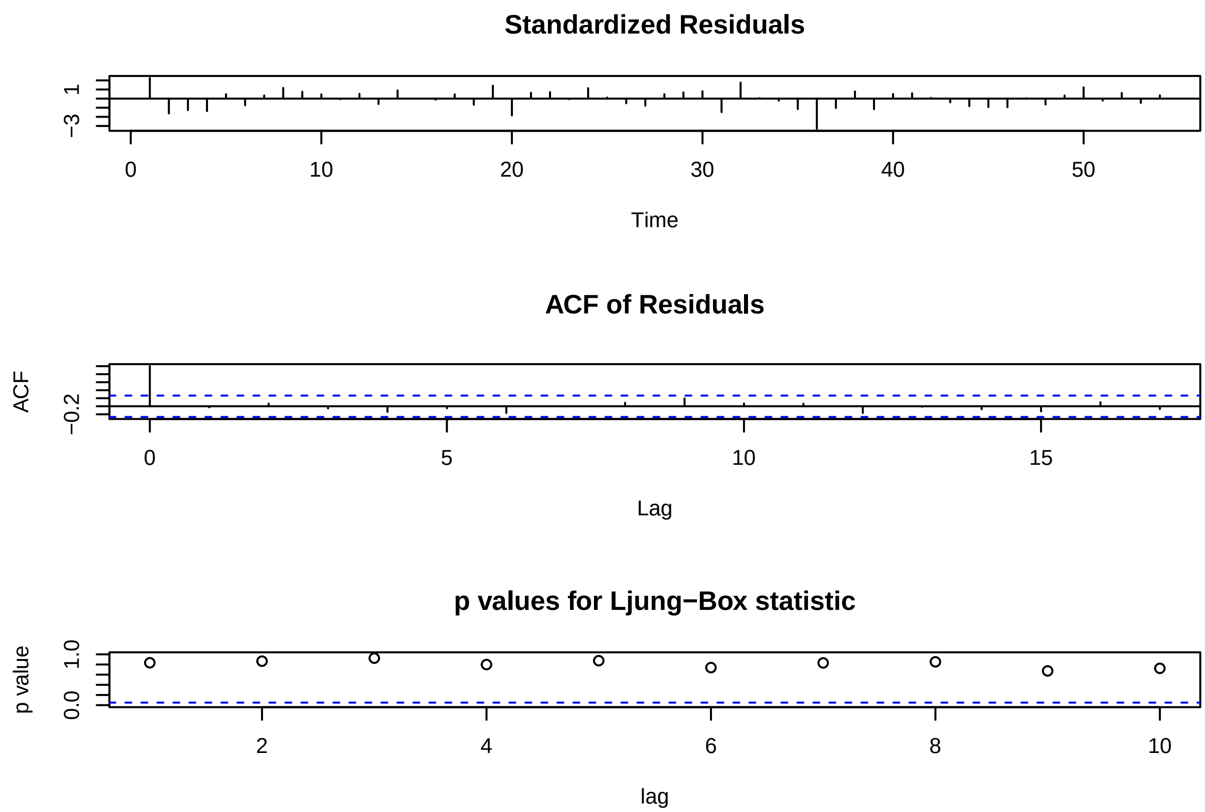
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0293587 -0.0056207  0.0005643  0.0062299  0.0191437
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.363e-03  6.793e-03   0.201  0.84239
## z.lag.1     -6.774e-01  1.938e-01  -3.495  0.00155 **
## tt          -6.538e-05  1.787e-04  -0.366  0.71713
## z.diff.lag   1.895e-01  1.789e-01   1.060  0.29806
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009687 on 29 degrees of freedom
## Multiple R-squared:  0.3101, Adjusted R-squared:  0.2387
## F-statistic: 4.345 on 3 and 29 DF,  p-value: 0.01204
```

```
##
##
## Value of test-statistic is: -3.4947 4.0743 6.1087
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

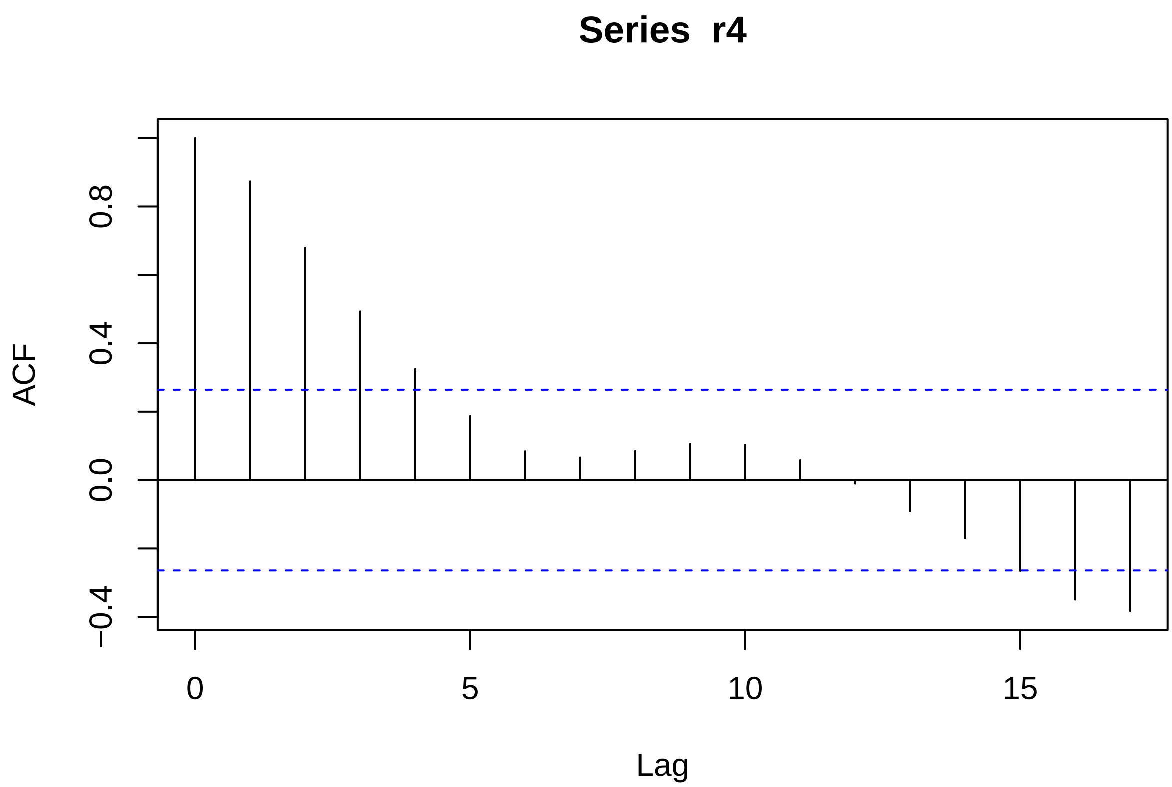
```
fit3 = auto.arima(diffr3)
summary(fit3)
```

```
## Series: diffr3
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##          ma1
##       0.4133
## s.e.  0.1326
##
## sigma^2 = 9.035e-05:  log likelihood = 175.21
## AIC=-346.41   AICc=-346.18   BIC=-342.44
##
## Training set error measures:
##                         ME        RMSE       MAE      MPE     MAPE      MASE
## Training set -0.0004624448 0.009416927 0.0074372 117.2988 166.8015 0.8205272
##                     ACF1
## Training set -0.02777735
```

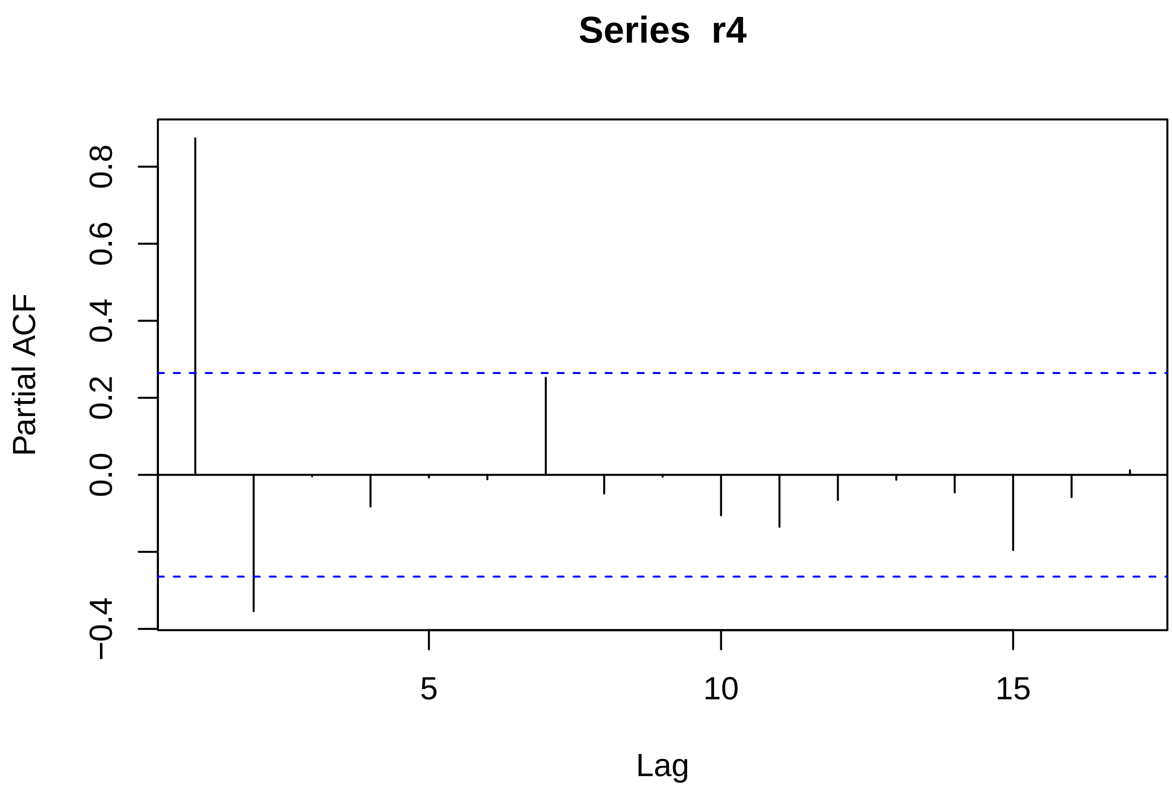
```
tsdiag(fit3)
```




```
##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
r4 = df.lev$IDE
acf(r4)
```




```
pacf(r4)
```




```
adf.test(r4)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  r4
## Dickey-Fuller = -2.1932, Lag order = 3, p-value = 0.4967
## alternative hypothesis: stationary
```

```
summary(ur.df(r4, type='trend', lags=20, selectlags="BIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0139191 -0.0019000  0.0005107  0.0026697  0.0081482
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0378140  0.0093466   4.046 0.000337 ***
## z.lag.1     -0.2599794  0.0674633  -3.854 0.000570 ***
## tt          -0.0003667  0.0001099  -3.337 0.002271 **
## z.diff.lag   0.4427329  0.1352629   3.273 0.002680 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004877 on 30 degrees of freedom
## Multiple R-squared:  0.4874, Adjusted R-squared:  0.4361
## F-statistic: 9.509 on 3 and 30 DF,  p-value: 0.000143
##
##
## Value of test-statistic is: -3.8536 5.5047 8.2569
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
# drift+tt+difference
diffr4 = na.omit(diff(r4))
summary(ur.df(diffr4, type='trend', lags=20, selectlags="BIC"))
```

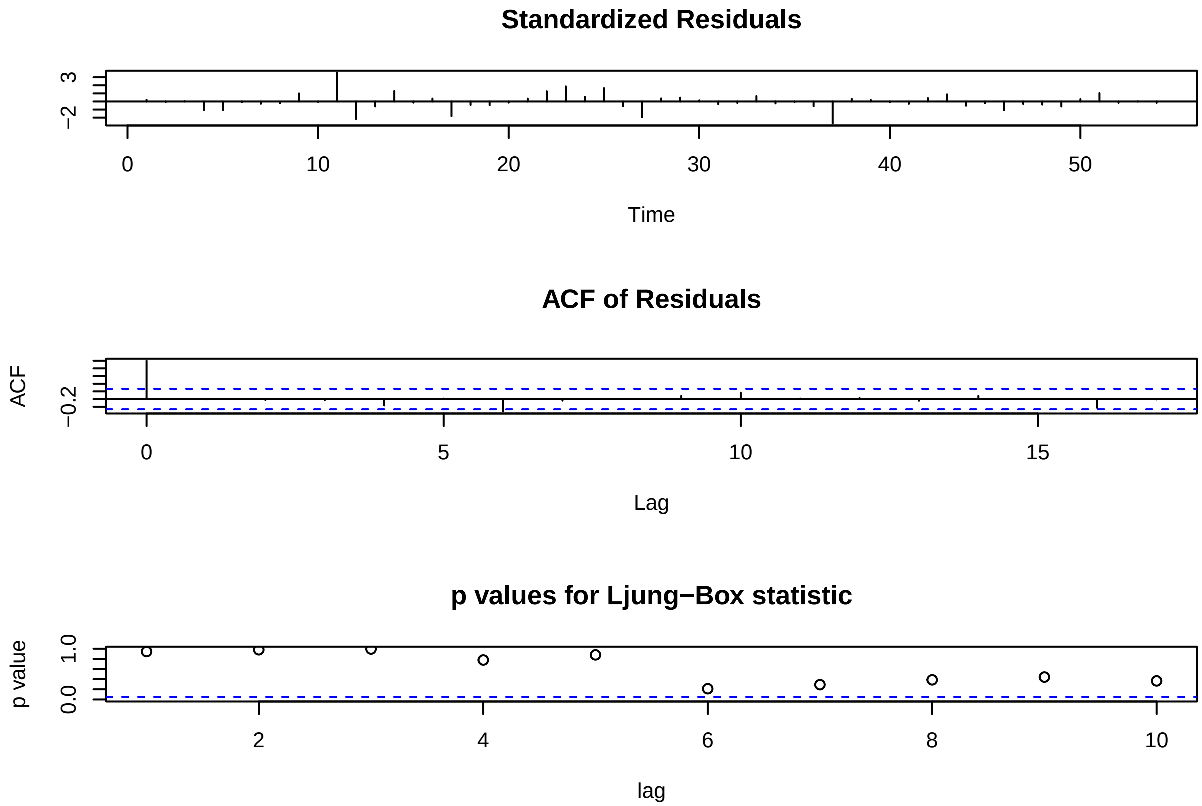
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0172159 -0.0027660 -0.0002231  0.0038719  0.0104575
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0049887  0.0042521   1.173 0.250245
## z.lag.1     -0.7612201  0.1919341  -3.966 0.000439 ***
## tt          -0.0001370  0.0001116  -1.227 0.229662
## z.diff.lag   0.2955934  0.1759922   1.680 0.103782
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005787 on 29 degrees of freedom
## Multiple R-squared:  0.3559, Adjusted R-squared:  0.2893
## F-statistic: 5.342 on 3 and 29 DF,  p-value: 0.004709
```

```
##
##
## Value of test-statistic is: -3.966 5.2463 7.8655
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

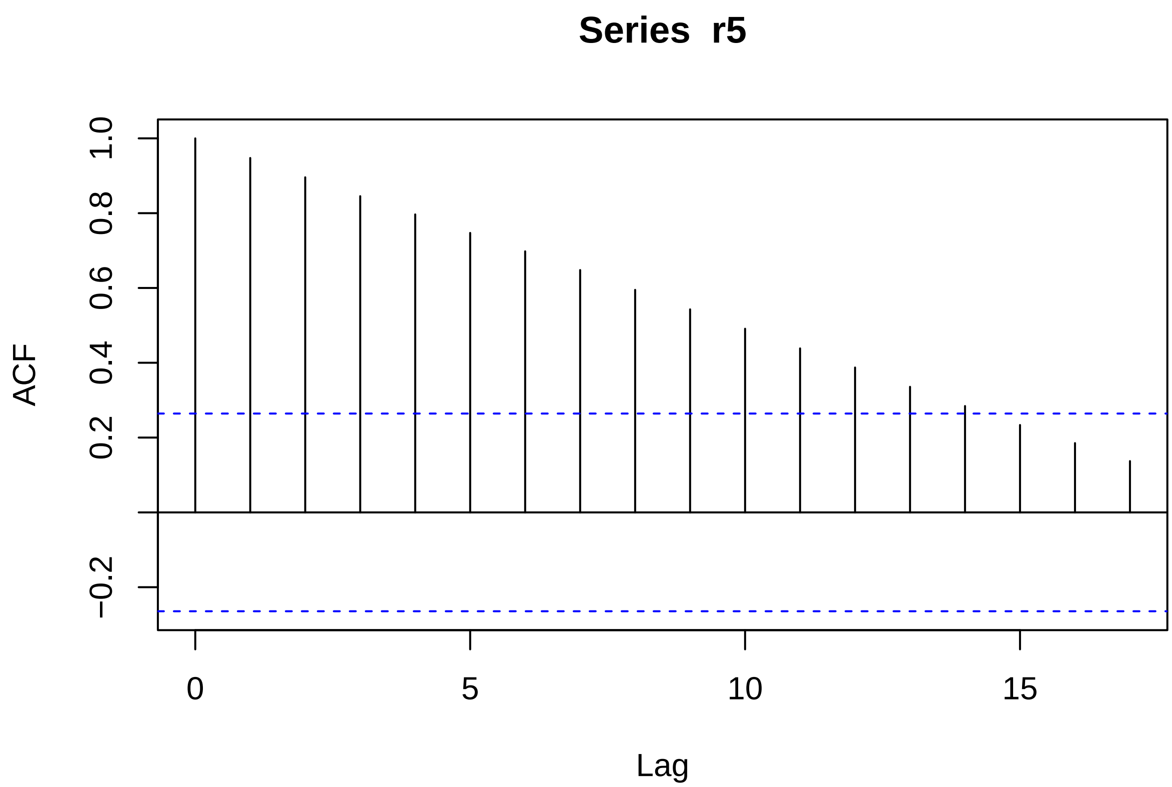
```
fit4 = auto.arima(diffr4)
summary(fit4)
```

```
## Series: diffr4
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##          ma1
##       0.3525
## s.e.  0.1382
##
## sigma^2 = 4.187e-05:  log likelihood = 196
## AIC=-388   AICc=-387.76   BIC=-384.02
##
## Training set error measures:
##                        ME        RMSE         MAE MPE MAPE     MASE
## Training set -0.000263954 0.006410705 0.004351406 NaN  Inf 0.747766
##                      ACF1
## Training set -0.009387154
```

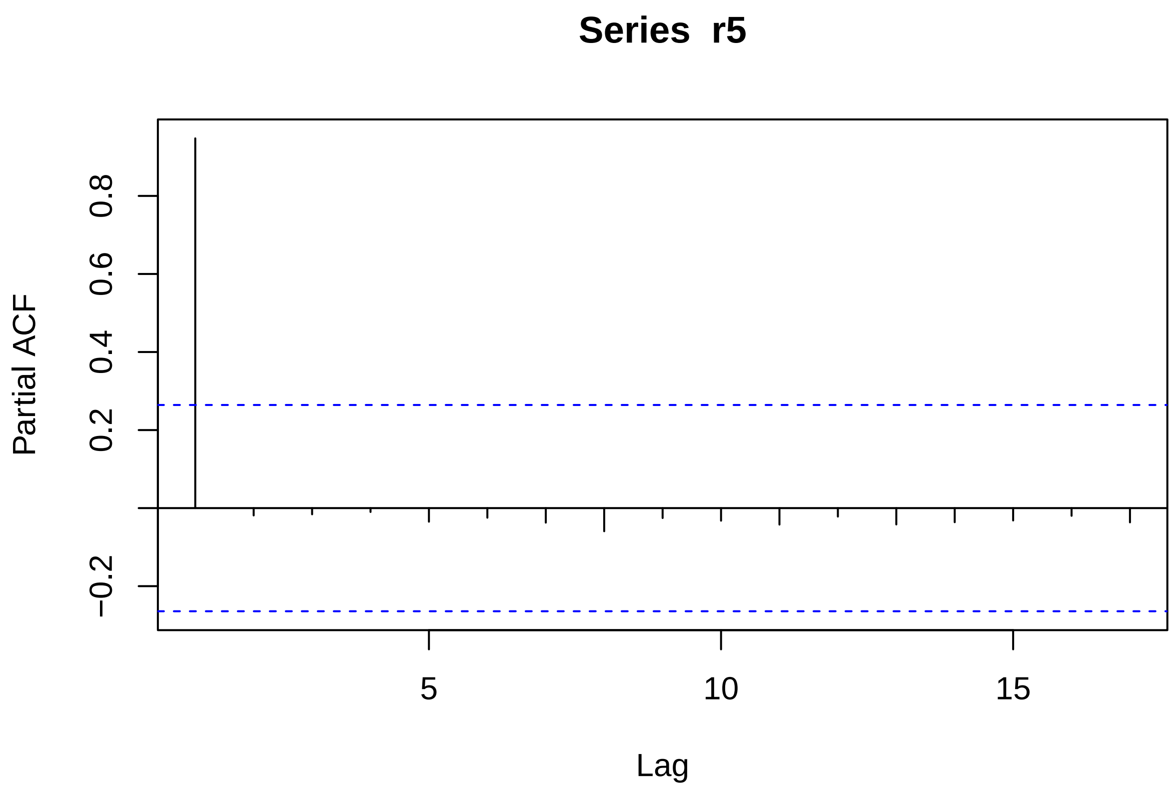
```
tsdiag(fit4)
```




```r
##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
r5 = df.lev$LPY
acf(r5)
```




```
pacf(r5)
```




```r
adf.test(r5)
```

```
## Warning in adf.test(r5): p-value greater than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  r5
## Dickey-Fuller = 0.52014, Lag order = 3, p-value = 0.99
## alternative hypothesis: stationary
```

```r
summary(ur.df(r5, type='trend', lags=20, selectlags="BIC"))
```

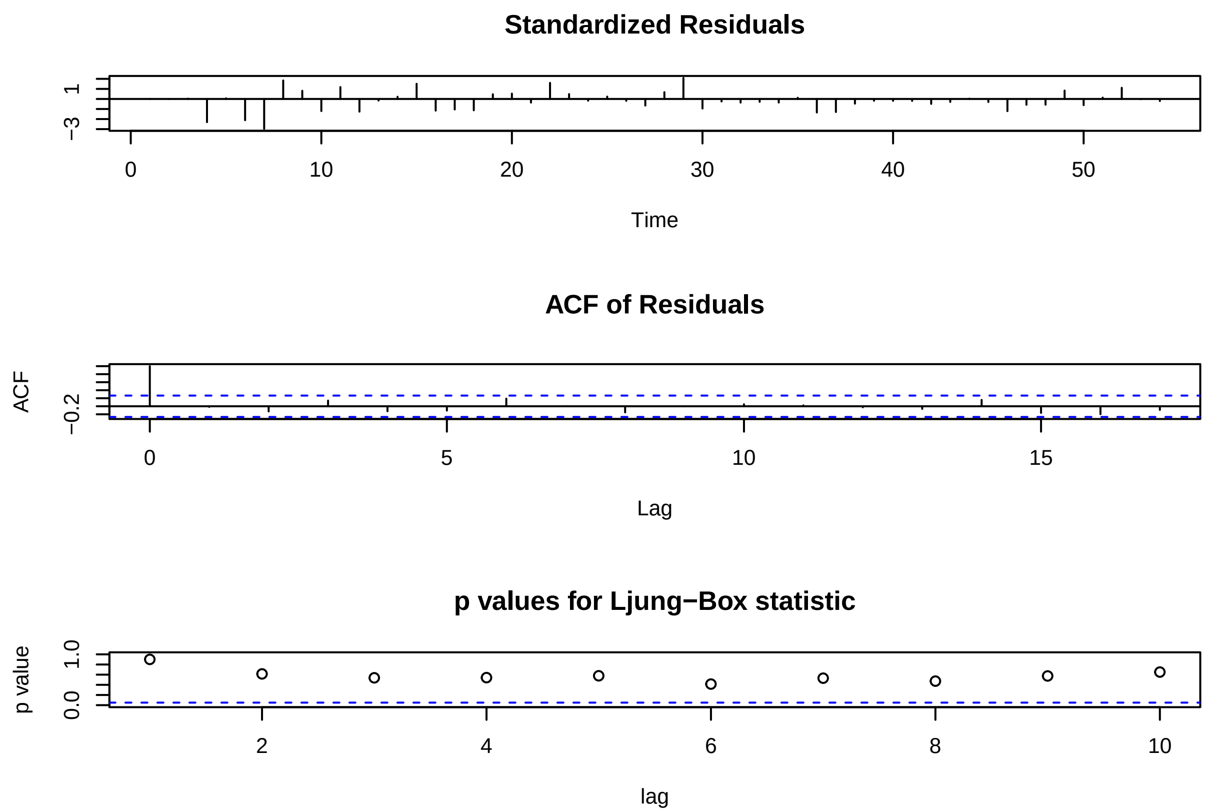
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
```

```
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0110527 -0.0040516  0.0003389  0.0024347  0.0175157
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0233066  0.0153437   1.519    0.139
## z.lag.1     -0.0229432  0.0267707  -0.857    0.398
## tt          -0.0001095  0.0005138  -0.213    0.833
## z.diff.lag   0.2007069  0.1765189   1.137    0.265
##
## Residual standard error: 0.005688 on 30 degrees of freedom
## Multiple R-squared:  0.5984, Adjusted R-squared:  0.5582
## F-statistic:  14.9 on 3 and 30 DF,  p-value: 4.031e-06
##
##
## Value of test-statistic is: -0.857 6.5383 6.6028
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
diffr5 = na.omit(diff(r5))
fit5 = auto.arima(diffr5)
summary(fit5)
```

```
## Series: diffr5
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.7244
## s.e.   0.1039
##
## sigma^2 = 6.896e-05:  log likelihood = 178.85
## AIC=-353.7   AICc=-353.46   BIC=-349.76
##
## Training set error measures:
##                        ME        RMSE         MAE       MPE     MAPE      MASE
## Training set -0.001683098 0.008149259 0.005979213 -47.52873 62.74448 0.8222477
##                     ACF1
## Training set -0.01653031
```

```
tsdiag(fit5)
```

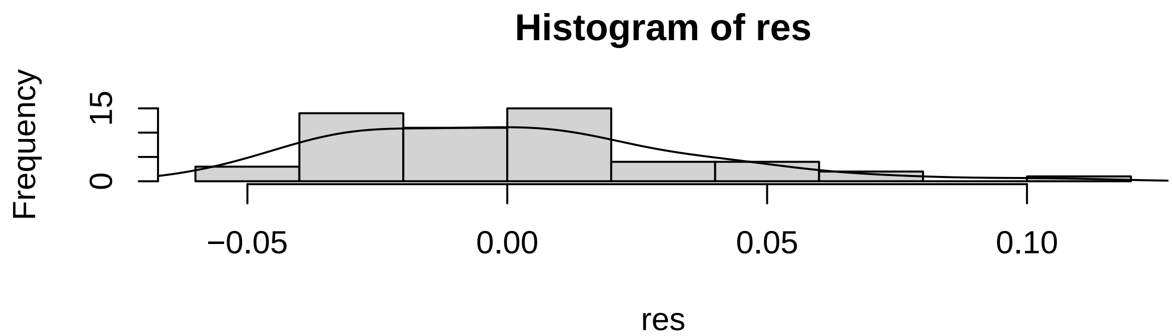


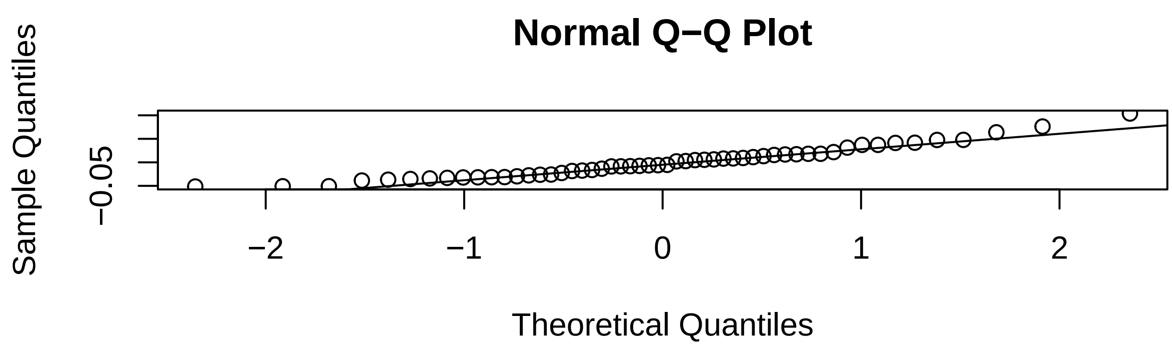
(c)

```
res=residuals(fit1)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.95285, p-value = 0.03316
```

```
par(mfrow=c(2,1))
hist(res)
lines(density(res))
qqnorm(res)
qqline(res)
```

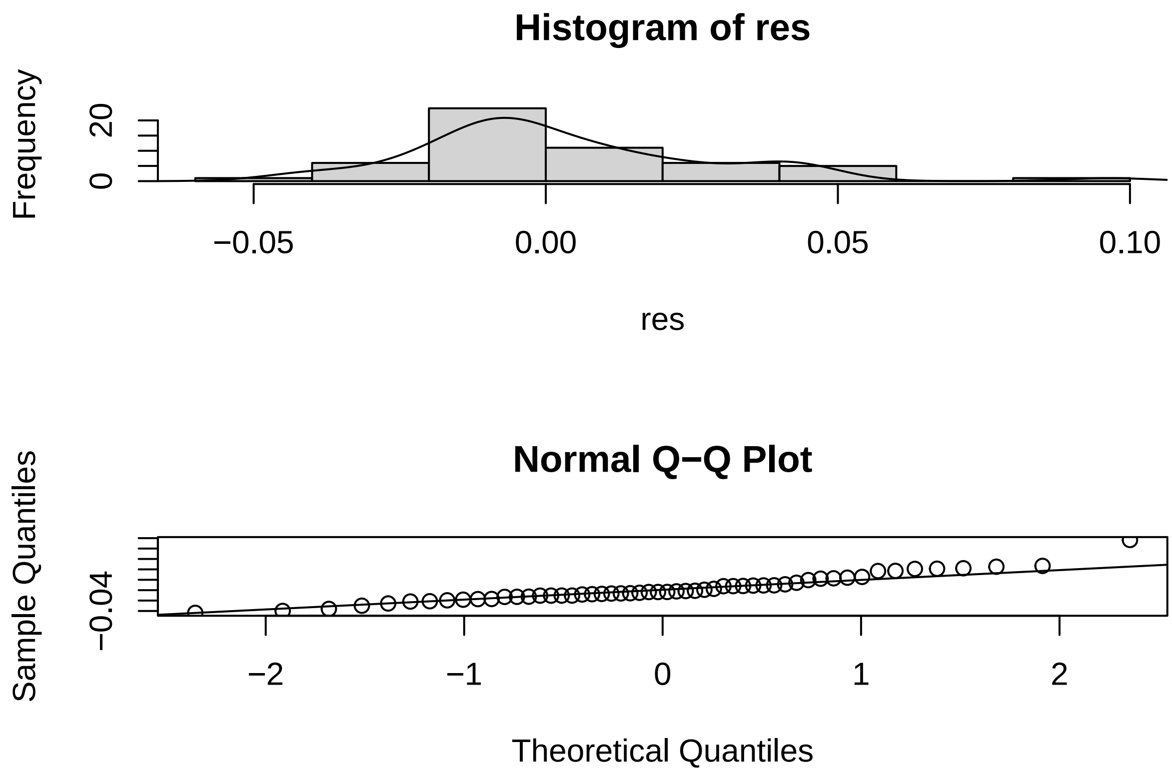





```
res=residuals(fit2)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.93223, p-value = 0.004479
```

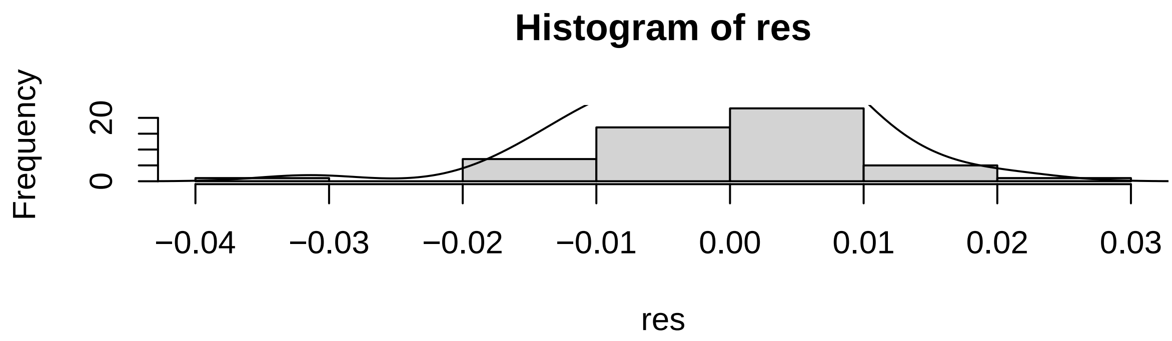
```
par(mfrow=c(2,1))
hist(res)
lines(density(res))
qqnorm(res)
qqline(res)
```

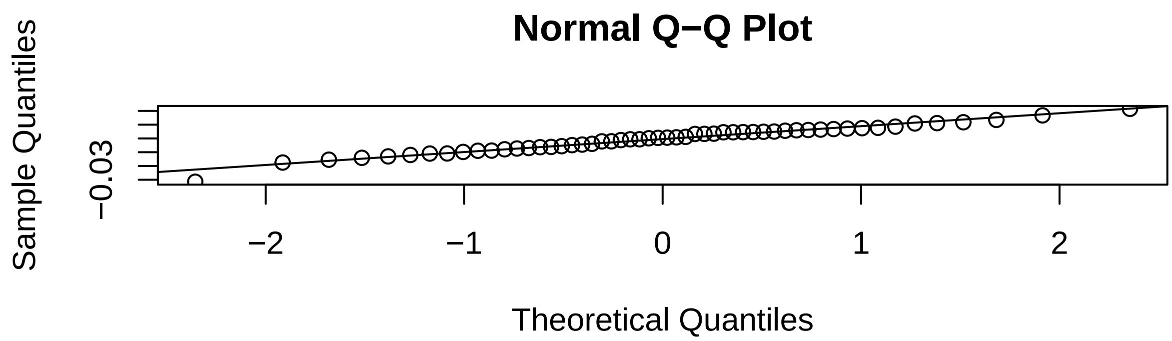



```
res=residuals(fit3)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97651, p-value = 0.3657
```

```
par(mfrow=c(2,1))
hist(res)
lines(density(res))
qqnorm(res)
qqline(res)
```

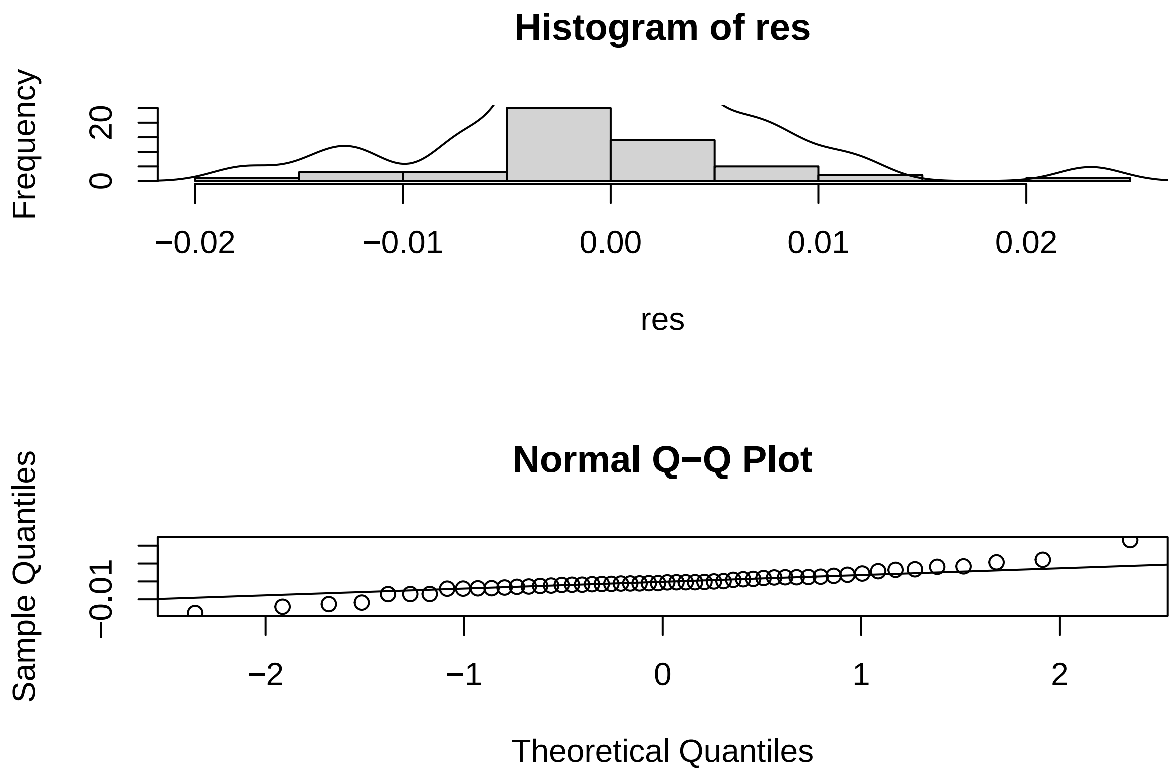





```r
res=residuals(fit4)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.92677, p-value = 0.002722
```

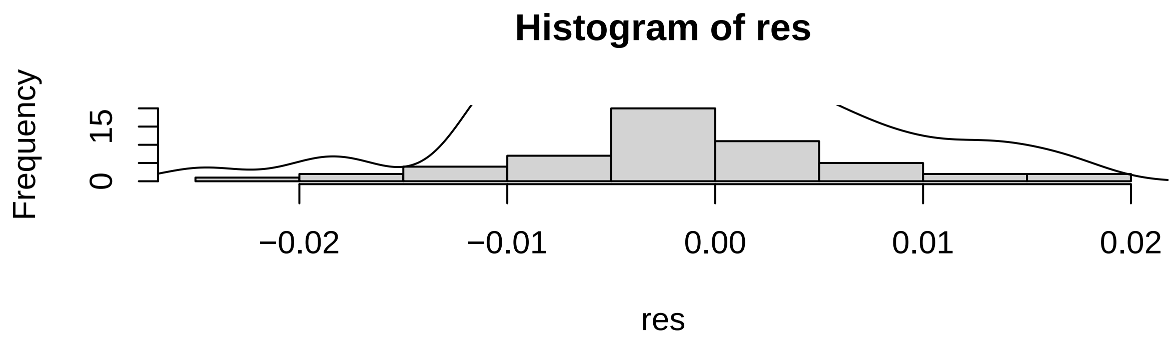
```r
par(mfrow=c(2,1))
hist(res)
lines(density(res))
qqnorm(res)
qqline(res)
```

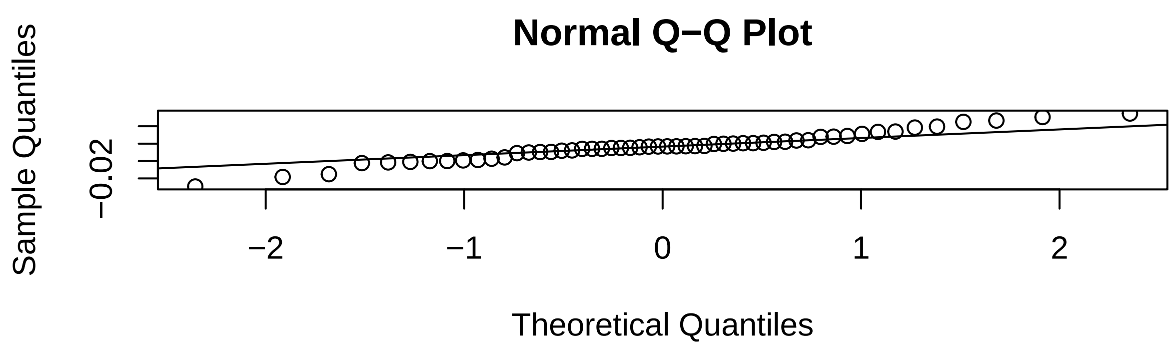



```
res=residuals(fit5)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.96857, p-value = 0.1667
```

```
par(mfrow=c(2,1))
hist(res)
lines(density(res))
qqnorm(res)
qqline(res)
```





(d)

```
# LRM
p1 <- forecast(fit1, h=3, level=0.95)
last1 <- r1[length(r1)]
(LRM1 <- last1+cumsum(p1$mean))
```

```
## [1] 12.02238 12.02947 12.03656
```

```
# 95% CI:
last1+cumsum(p1$lower) # lb
```

```
## [1] 11.95743 11.89957 11.84170
```

```
last1+cumsum(p1$upper) # ub
```

```
## [1] 12.08733 12.15937 12.23141
```

```
# LRY
p2 <- forecast(fit2, h=3, level=0.95)
last2 <- r2[length(r2)]
(LRY1 <- last2+cumsum(p2$mean))
```

```
## [1] 6.05083 6.05083 6.05083
```

```r
# 95% CI:
last2+cumsum(p2$lower) # lb
```

```
## [1] 6.001213 5.951596 5.901980
```

```r
last2+cumsum(p2$upper) # ub
```

```
## [1] 6.100447 6.150064 6.199680
```

```r
# IBO
p3 <- forecast(fit3, h=3, level=0.95)
last3 <- r3[length(r3)]
(IBO1 <- last3+cumsum(p3$mean))
```

```
## [1] 0.1204021 0.1204021 0.1204021
```

```r
# 95% CI:
last3+cumsum(p3$lower) # lb
```

```
## [1] 0.10177198 0.08161339 0.06145480
```

```r
last3+cumsum(p3$upper) # ub
```

```
## [1] 0.1390323 0.1591909 0.1793494
```

```r
# IDE
p4 <- forecast(fit4, h=3, level=0.95)
last4 <- r4[length(r4)]
(IDE1 <- last4+cumsum(p4$mean))
```

```
## [1] 0.07477594 0.07477594 0.07477594
```

```r
# 95% CI:
last4+cumsum(p4$lower) # lb
```

```
## [1] 0.06209321 0.04864540 0.03519760
```

```r
last4+cumsum(p4$upper) # ub
```

```
## [1] 0.08745867 0.10090648 0.11435428
```

```r
# LPY
p5 <- forecast(fit5, h=3, level=0.95)
last5 <- r5[length(r5)]
(LPY1 <- last5+cumsum(p5$mean))
```

```
## [1] 0.4795805 0.4908265 0.5020725
```

```r
# 95% CI:
last5+cumsum(p5$lower) # lb
```

```
## [1] 0.4633040 0.4576667 0.4514436
```

```r
last5+cumsum(p5$upper) # ub
```

```
## [1] 0.4958570 0.5239864 0.5527014
```

2. Please (a) build the best VAR models using all five variables together. (b) Please be sure to check the integrated order, and make considerations for trend and seasonality. (c) Please check the residuals to ensure your model is a good fit. (d). Please use your best VAR model to make forecast for the next three quarters.

```r
df.diff = diff(as.matrix(df.lev), lag = 1)
colnames(df.diff) = c('dLRM', 'dLRY', 'dIBO', 'dIDE', 'dLPY')
m.diff = as.matrix(df.diff)
# lag length
VARselect(df.diff, lag.max = 4, type = 'const', season = 4)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      4      1      1      1
##
## $criteria
##                     1             2             3             4
## AIC(n) -4.473719e+01 -4.452122e+01 -4.428230e+01 -4.496819e+01
## HQ(n)  -4.408189e+01 -4.350187e+01 -4.289889e+01 -4.322073e+01
## SC(n)  -4.301637e+01 -4.184439e+01 -4.064946e+01 -4.037934e+01
## FPE(n)  3.797496e-20  4.989343e-20  7.170655e-20  4.541927e-20
```

```r
# estimation
vare_diff = VAR(df.diff, p = 1, type = 'const', season = 4)
summary(vare_diff)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: dLRM, dLRY, dIBO, dIDE, dLPY
## Deterministic variables: const
## Sample size: 53
## Log Likelihood: 854.848
## Roots of the characteristic polynomial:
## 0.6244 0.6244 0.4702 0.2424 0.1406
## Call:
## VAR(y = df.diff, p = 1, type = "const", season = 4L)
##
##
## Estimation results for equation dLRM:
## =====================================
```

```
## dLRM = dLRM.l1 + dLRY.l1 + dIBO.l1 + dIDE.l1 + dLPY.l1 + const + sd1 + sd2 + sd3
##
##          Estimate Std. Error t value Pr(>|t|)
## dLRM.l1  0.401690   0.184714   2.175 0.035074 *
## dLRY.l1 -0.134204   0.161998  -0.828 0.411900
## dIBO.l1 -0.752716   0.387901  -1.940 0.058745 .
## dIDE.l1 -0.321308   0.587225  -0.547 0.587031
## dLPY.l1  0.030032   0.459989   0.065 0.948240
## const    0.004066   0.010592   0.384 0.702908
## sd1      0.055745   0.013504   4.128 0.000161 ***
## sd2      0.023774   0.010130   2.347 0.023493 *
## sd3      0.066215   0.011465   5.775 7.21e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02352 on 44 degrees of freedom
## Multiple R-Squared: 0.5723,  Adjusted R-squared: 0.4945
## F-statistic: 7.359 on 8 and 44 DF,  p-value: 3.656e-06
##
##
## Estimation results for equation dLRY:
## =====================================
## dLRY = dLRM.l1 + dLRY.l1 + dIBO.l1 + dIDE.l1 + dLPY.l1 + const + sd1 + sd2 + sd3
##
##          Estimate Std. Error t value Pr(>|t|)
## dLRM.l1  0.521260   0.182756   2.852  0.00659 **
## dLRY.l1 -0.131002   0.160281  -0.817  0.41815
## dIBO.l1  0.084805   0.383789   0.221  0.82614
## dIDE.l1 -0.428987   0.581000  -0.738  0.46422
## dLPY.l1  0.135191   0.455113   0.297  0.76783
## const   -0.003003   0.010479  -0.287  0.77581
## sd1      0.026721   0.013361   2.000  0.05171 .
## sd2      0.009415   0.010022   0.939  0.35265
## sd3      0.022310   0.011343   1.967  0.05554 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02327 on 44 degrees of freedom
## Multiple R-Squared: 0.2805,  Adjusted R-squared: 0.1497
## F-statistic: 2.144 on 8 and 44 DF,  p-value: 0.05135
##
##
## Estimation results for equation dIBO:
## =====================================
## dIBO = dLRM.l1 + dLRY.l1 + dIBO.l1 + dIDE.l1 + dLPY.l1 + const + sd1 + sd2 + sd3
##
##           Estimate Std. Error t value Pr(>|t|)
## dLRM.l1  0.0703650  0.0660249   1.066  0.29236
## dLRY.l1  0.1471801  0.0579054   2.542  0.01463 *
## dIBO.l1  0.3908819  0.1386530   2.819  0.00719 **
## dIDE.l1  0.1701401  0.2099001   0.811  0.42197
## dLPY.l1  0.0784661  0.1644203   0.477  0.63556
```

```
## const   -0.0033868  0.0037859  -0.895  0.37588
## sd1      0.0080372  0.0048271   1.665  0.10301
## sd2      0.0047316  0.0036208   1.307  0.19808
## sd3      0.0006048  0.0040981   0.148  0.88334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.008408 on 44 degrees of freedom
## Multiple R-Squared: 0.3774,  Adjusted R-squared: 0.2642
## F-statistic: 3.334 on 8 and 44 DF,  p-value: 0.004575
##
##
## Estimation results for equation dIDE:
## =====================================
## dIDE = dLRM.l1 + dLRY.l1 + dIBO.l1 + dIDE.l1 + dLPY.l1 + const + sd1 + sd2 + sd3
##
##          Estimate Std. Error t value Pr(>|t|)
## dLRM.l1  0.064389   0.045613   1.412 0.165092
## dLRY.l1  0.028880   0.040004   0.722 0.474158
## dIBO.l1  0.360907   0.095788   3.768 0.000486 ***
## dIDE.l1  0.138121   0.145009   0.952 0.346048
## dLPY.l1  0.147664   0.113590   1.300 0.200376
## const   -0.003645   0.002616  -1.394 0.170450
## sd1      0.003010   0.003335   0.903 0.371610
## sd2      0.001116   0.002501   0.446 0.657706
## sd3      0.004734   0.002831   1.672 0.101622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005809 on 44 degrees of freedom
## Multiple R-Squared: 0.3997,  Adjusted R-squared: 0.2906
## F-statistic: 3.662 on 8 and 44 DF,  p-value: 0.002388
##
##
## Estimation results for equation dLPY:
## =====================================
## dLPY = dLRM.l1 + dLRY.l1 + dIBO.l1 + dIDE.l1 + dLPY.l1 + const + sd1 + sd2 + sd3
##
##          Estimate Std. Error t value Pr(>|t|)
## dLRM.l1  0.029470   0.066831   0.441  0.66140
## dLRY.l1  0.054297   0.058613   0.926  0.35931
## dIBO.l1  0.241136   0.140346   1.718  0.09280 .
## dIDE.l1 -0.168564   0.212464  -0.793  0.43182
## dLPY.l1  0.562287   0.166429   3.379  0.00153 **
## const    0.008168   0.003832   2.131  0.03867 *
## sd1      0.003643   0.004886   0.746  0.45989
## sd2     -0.003423   0.003665  -0.934  0.35537
## sd3      0.002570   0.004148   0.619  0.53880
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.008511 on 44 degrees of freedom
## Multiple R-Squared: 0.3162,  Adjusted R-squared: 0.1918
## F-statistic: 2.543 on 8 and 44 DF,  p-value: 0.02273
##
##
##
## Covariance matrix of residuals:
##            dLRM       dLRY       dIBO       dIDE       dLPY
## dLRM  5.533e-04  2.111e-04 -8.202e-05 -4.874e-05 -1.213e-04
## dLRY  2.111e-04  5.417e-04 -8.713e-06 -2.572e-05 -9.491e-05
## dIBO -8.202e-05 -8.713e-06  7.070e-05  1.290e-05  1.268e-05
## dIDE -4.874e-05 -2.572e-05  1.290e-05  3.374e-05  1.173e-05
## dLPY -1.213e-04 -9.491e-05  1.268e-05  1.173e-05  7.244e-05
##
## Correlation matrix of residuals:
##         dLRM     dLRY     dIBO    dIDE    dLPY
## dLRM  1.0000  0.38562 -0.41469 -0.3567 -0.6058
## dLRY  0.3856  1.00000 -0.04453 -0.1903 -0.4791
## dIBO -0.4147 -0.04453  1.00000  0.2641  0.1772
## dIDE -0.3567 -0.19028  0.26412  1.0000  0.2373
## dLPY -0.6058 -0.47913  0.17716  0.2373  1.0000
```

```r
# residuals test
serial.test(vare_diff)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object vare_diff
## Chi-squared = 342.24, df = 375, p-value = 0.8865
```

```r
# forecast of differenced data
varf_diff = predict(vare_diff, n.ahead = 3, ci = 0.95)
# predictions & 95%CI
(LRM2 <- last1+cumsum(varf_diff$fcst$dLRM[,1]))
```

```
## [1] 12.04046 12.02601 12.05205
```

```r
last1+cumsum(varf_diff$fcst$dLRM[,2]) # lb
```

```
## [1] 11.99435 11.92678 11.89777
```

```r
last1+cumsum(varf_diff$fcst$dLRM[,3]) # ub
```

```
## [1] 12.08656 12.12525 12.20633
```

```r
(LRY2 <- last2+cumsum(varf_diff$fcst$dLRY[,1]))
```

```
## [1] 6.046409 6.044270 6.049727
```

```r
last2+cumsum(varf_diff$fcst$dLRY[,2]) # lb
```

```
## [1] 6.000793 5.947671 5.900720
```

```r
last2+cumsum(varf_diff$fcst$dLRY[,3]) # ub
```

```
## [1] 6.092025 6.140869 6.198735
```

```r
(IBO2 <- last3+cumsum(varf_diff$fcst$dIBO[,1]))
```

```
## [1] 0.11045617 0.10248479 0.09955815
```

```r
last3+cumsum(varf_diff$fcst$dIBO[,2]) # lb
```

```
## [1] 0.09397633 0.06709616 0.04433615
```

```r
last3+cumsum(varf_diff$fcst$dIBO[,3]) # ub
```

```
## [1] 0.1269360 0.1378734 0.1547801
```

```r
(IDE2 <- last4+cumsum(varf_diff$fcst$dIDE[,1]))
```

```
## [1] 0.07400246 0.06859206 0.06326033
```

```r
last4+cumsum(varf_diff$fcst$dIDE[,2]) # lb
```

```
## [1] 0.06261739 0.04425943 0.02533070
```

```r
last4+cumsum(varf_diff$fcst$dIDE[,3]) # ub
```

```
## [1] 0.08538753 0.09292470 0.10118995
```

```r
(LPY2 <- last5+cumsum(varf_diff$fcst$dLPY[,1]))
```

```
## [1] 0.4831566 0.4976067 0.5152937
```

```r
last5+cumsum(varf_diff$fcst$dLPY[,2]) # lb
```

```
## [1] 0.4664755 0.4620170 0.4602349
```

```r
last5+cumsum(varf_diff$fcst$dLPY[,3]) # ub
```

```
## [1] 0.4998377 0.5331964 0.5703526
```

3. we have already got forecast for the next 3 quarters based on two methods. If we know the true values of the following 3 quarters, we can compute the MSE to compare which one is better.

“‘