Timeseries HW6

Ze YANG

10/15/2018

Ruppert Page 397 Problem 9-11

Problem 9

```
library(car)
```

```
## Loading required package: carData
```

```
library(AER)
```

```
## Loading required package: lmtest
```

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

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

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

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```
library(forecast)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'default/
## America/New_York'
```

```
library(arfima)
```

```
## Loading required package: ltsa
```

```
## Note that the arfima package has new defaults starting with
## 1.4-0: type arfimachanges() for a list, as well as some other notes.
## NOTE: some of these are quite important!
```

```
##
## Attaching package: 'arfima'
```

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

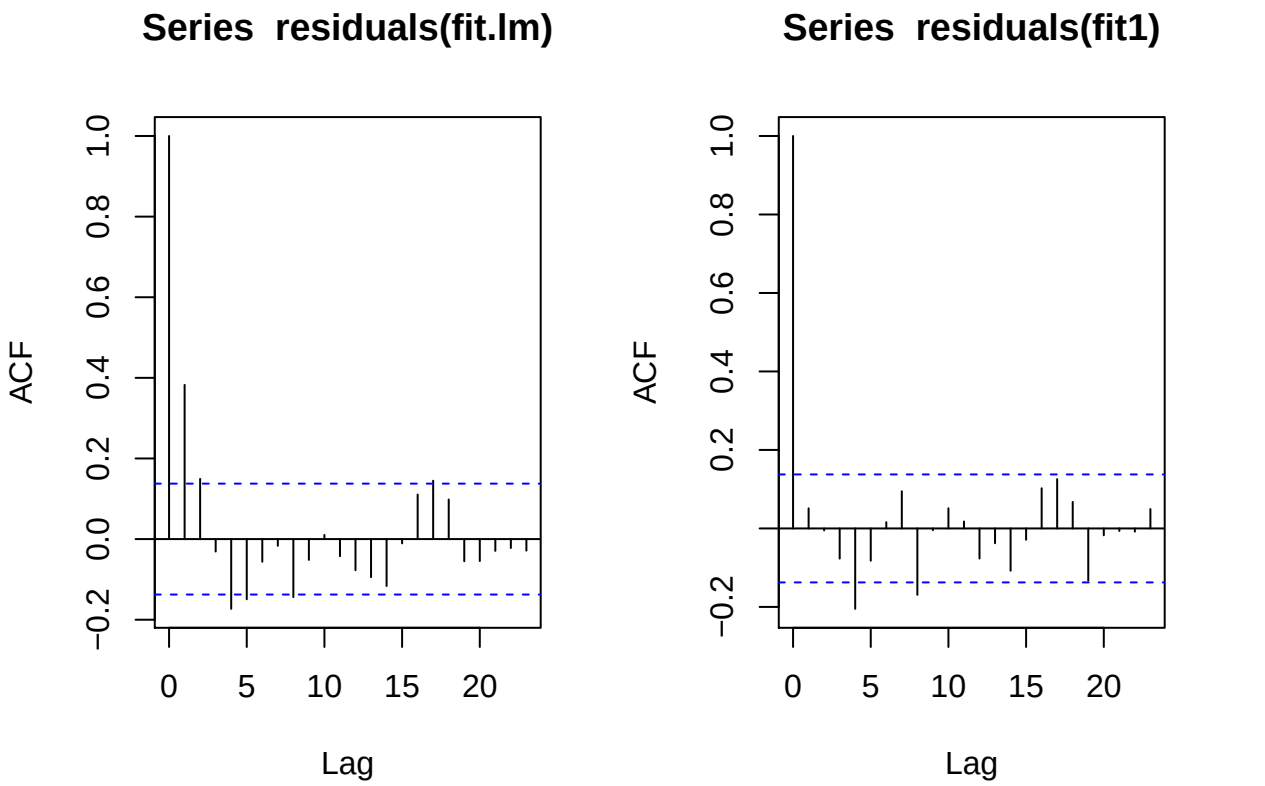
```
## The following object is masked from 'package:stats':
##
##     BIC
```

```
data("USMacroG")
MacroDiff = as.data.frame(apply(USMacroG, 2, diff))
attach(MacroDiff)
fit1 = arima(unemp, order=c(1,0,0), xreg=cbind(invest, government))
```

```
fit.lm = lm(unemp ~ invest + government)
c(AIC(fit1), AIC(fit.lm))
```

```
## [1]  86.85233 138.88245
```

```
par(mfrow=c(1,2))
acf(residuals(fit.lm))
acf(residuals(fit1))
```




```
durbinWatsonTest(fit.lm, 5)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.38257735      1.212435   0.000
##    2       0.14938261      1.635514   0.008
##    3      -0.03084882      1.980947   0.996
##    4      -0.17313955      2.238821   0.038
##    5      -0.14934313      2.180022   0.120
##  Alternative hypothesis: rho[lag] != 0
```

Comments:

- The AR(1) model has smaller AIC (86.85233 V.S. 138.88245).
- The ACF plot shows that the residuals of AR(1) fit has no significant serial correlation. Hence AR(1) model is a better fit.
- The Durbin-Watson test on the simple linear regression model suggests we reject the null hypothesis for lag = 1 and 2, that is, the residuals exhibits serial correlation for lag = 1,2.

Problem 11

```
fit2 = arima(unemp, order=c(2,0,0), xreg=cbind(invest, government))
fit11 = arima(unemp, order=c(1,0,1), xreg=cbind(invest, government))
c(AIC(fit2), AIC(fit11), AIC(fit1), AIC(fit.lm))
```

```
## [1]  86.27964  87.19768  86.85233 138.88245
```

The AR(2) model has a smaller AIC than AR(1) and ARMA(1,1).

Ford Data

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: TTR
```

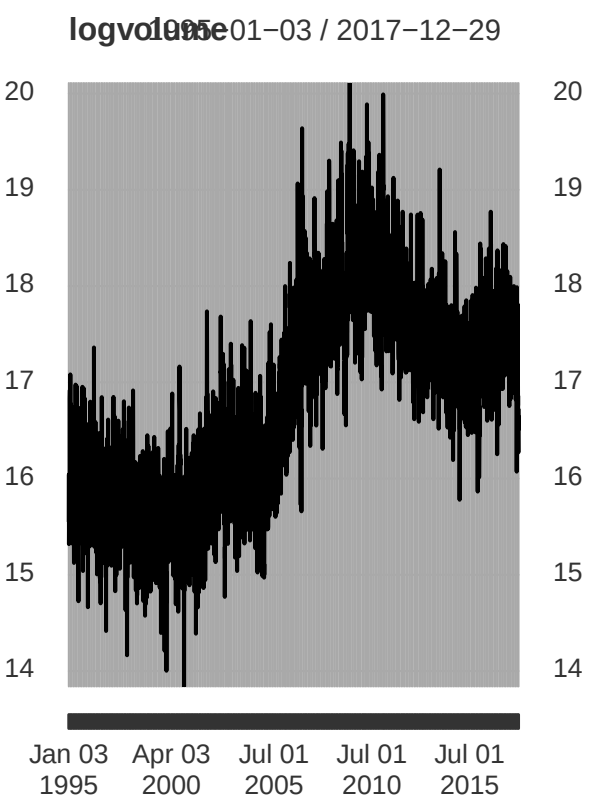
```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

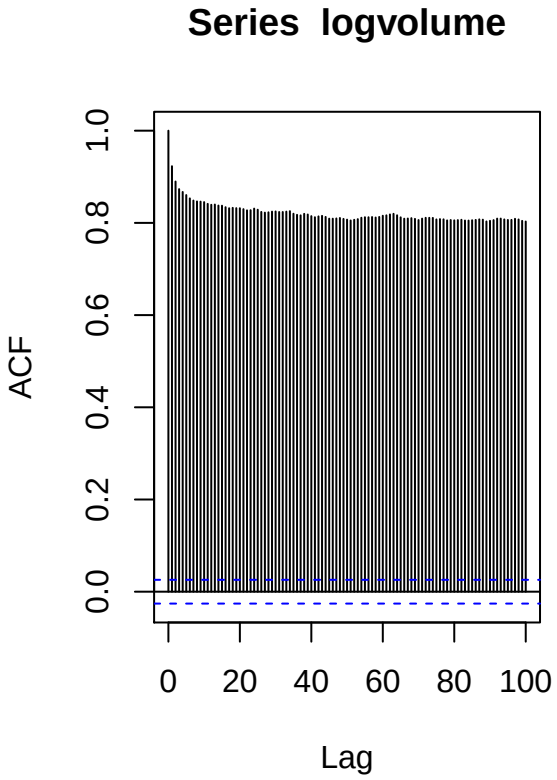
```
FORD = getSymbols('F', from='1995-1-1', to='2017-12-31', auto.assign = F)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```
volume = FORD$F.Volume
logvolume = log(volume)
par(mfrow=c(1,2))
plot(logvolume)
acf(logvolume, 100)
```





- Yes, it is probably a good idea to fit ARFIMA model on the log volume data, because it as an ACF plot that decays very slowly, which matches the long-memory characteristic that ARFIMA models.

```
# Takes very long time to run
maxp = 5
maxq = 5
holdaic = matrix(0, nrow=(maxp+1), ncol=(maxq+1))
for (p in 0:maxp) {
  for (q in 0:maxq) {
    holdaic[(p+1), (q+1)] = summary(
      arfima(ts(logvolume), order=c(p,0,q), quiet=TRUE))$aic
    print(paste('Fitted model', p, q))
  }
}
```

```
# I saved the results in a file
holdaic = as.data.frame(read.csv('holdaic.csv'))
holdaic
```

```
##   X        V1        V2        V3        V4        V5        V6
## 1 1 -11983.27 -11998.30 -12001.49 -12018.40 -12017.36 -12014.29
## 2 2 -11996.42 -11997.98 -12019.80 -12018.97 -12019.67 -12018.21
## 3 3 -12003.90 -12020.32 -12018.33 -12018.45 -11994.33 -11919.82
## 4 4 -12016.92 -12006.60 -12010.73 -12014.88 -11943.61 -11928.33
## 5 5 -12015.09 -12019.12 -12017.78 -12016.95 -11910.66 -12012.81
## 6 6 -12010.39 -12018.89 -12016.10 -12020.64 -11998.40      0.00
```

```
fit53 = arfima(ts(logvolume), order=c(5,0,3), quiet=TRUE)
```

```
coef(fit53)
```

```
##            phi(1)      phi(2)    phi(3)      phi(4)      phi(5)   theta(1)
## Mode 1 -0.4700237 -0.09331057 0.6614612 -0.07099347 0.003657788 -0.4968304
##           theta(2)  theta(3)       d.f Fitted mean
## Mode 1 -0.05025472 0.7387022 0.4966248    16.66867
```

```
AIC(fit53)
```

```
## [1] -12020.64
```

The best model based on AIC is ARFIMA(5, 0.4966, 3), which has an AIC of -12020.64.