FTAP Homework 8

Zachary Fogelson

July 22, 2015

Problem 1

$$E[AR(1)] = \frac{\beta_0}{1 - \beta_1}$$

a i. $\mu_1 = 0$

ii. $\mu_2 = 0$

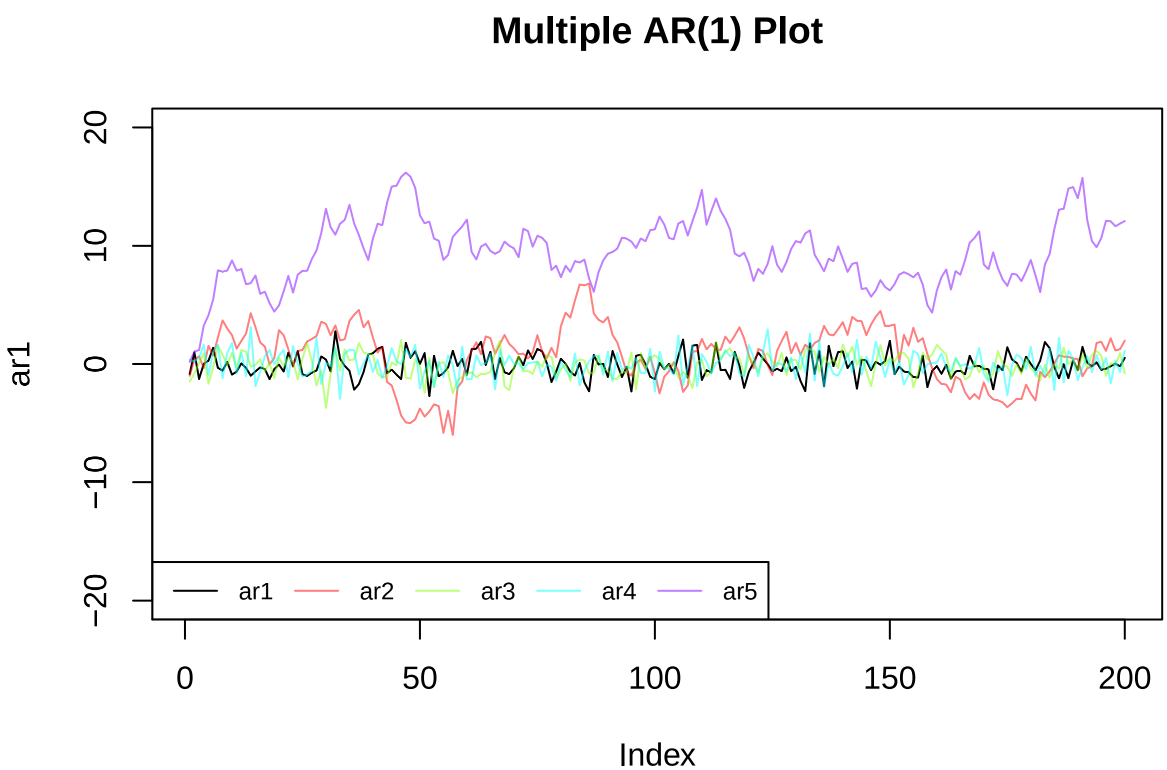
iii. $\mu_3 = 0$

iv. $\mu_4 = 0$

v. $\mu_5 = 10$

b

```
ar1 <- genAR_1(0,0,200,1)
ar2 <- genAR_1(0,.9,200,1)
ar3 <- genAR_1(0,.1,200,1)
ar4 <- genAR_1(0,-.5,200,1)
ar5 <- genAR_1(1,.9,200,1)
plot(ar1, type = "l", ylim = c(-20,20), main="Multiple AR(1) Plot")
lines(ar2, col = rainbow(4, alpha = .5)[1])
lines(ar3, col = rainbow(4, alpha  = .5)[2])
lines(ar4, col = rainbow(4, alpha = .5)[3])
lines(ar5, col = rainbow(4, alpha = .5)[4])
legend("bottomleft", c("ar1","ar2", "ar3", "ar4", "ar5"), col=c('black', rainbow(4, alpha = .5)[1], rai
```



Yes, the plots look like I would expect, all plots are centered around their respective means and "swing" as a function of their β_1

c

Yes, all of the series appear to be mean reverting, series 1-4 revert to 0, series 5 reverts to 10.

d

```
pr <- function(p){
  print(p$pred)
}

p1 <- predict(arima(ar1, order = c(1,0,0)), n.ahead = 1, Trace = F)
p2 <- predict(arima(ar2, order = c(1,0,0)), n.ahead = 1, Trace = F)
p3 <- predict(arima(ar3, order = c(1,0,0)), n.ahead = 1, Trace = F)
p4 <- predict(arima(ar4, order = c(1,0,0)), n.ahead = 1, Trace = F)
p5 <- predict(arima(ar5, order = c(1,0,0)), n.ahead = 1, Trace = F)
```

```
mapply(pr, list(p1, p2, p3, p4, p5))
```

```
## Time Series:
## Start = 201
## End = 201
## Frequency = 1
## [1] -0.1213941
## Time Series:
## Start = 201
## End = 201
## Frequency = 1
```

```
## [1] 1.851293
## Time Series:
## Start = 201
## End = 201
## Frequency = 1
## [1] -0.194344
## Time Series:
## Start = 201
## End = 201
## Frequency = 1
## [1] -0.4515747
## Time Series:
## Start = 201
## End = 201
## Frequency = 1
## [1] 11.88553
```

```
## [1] -0.1213941  1.8512925 -0.1943440 -0.4515747 11.8855308
```

e

```
oneStep <- function(beta0, beta1, yt){
  beta0 + beta1*yt
}

arimas <- data.frame(beta0 = c(0,0,0,0,1), beta1 = c(0,.9, .1, -.5, .9), yt = c(ar1[200], ar2[200], ar3

mapply(oneStep, arimas$beta0, arimas$beta1, arimas$yt)
```

```
## [1]  0.00000000  1.78388603 -0.07847846 -0.55789119 11.87452939
```

f

Conditional Variance:

$$Var(Y_t|Y_{t-1}) = Var(\beta_0 + \beta_1 Y_{t-1} + \epsilon_t)$$

+ Because Y_{t-1} is given all terms excpet for ϵ_t are constants therefore there is the covariance term goes to 0 and we are left with:

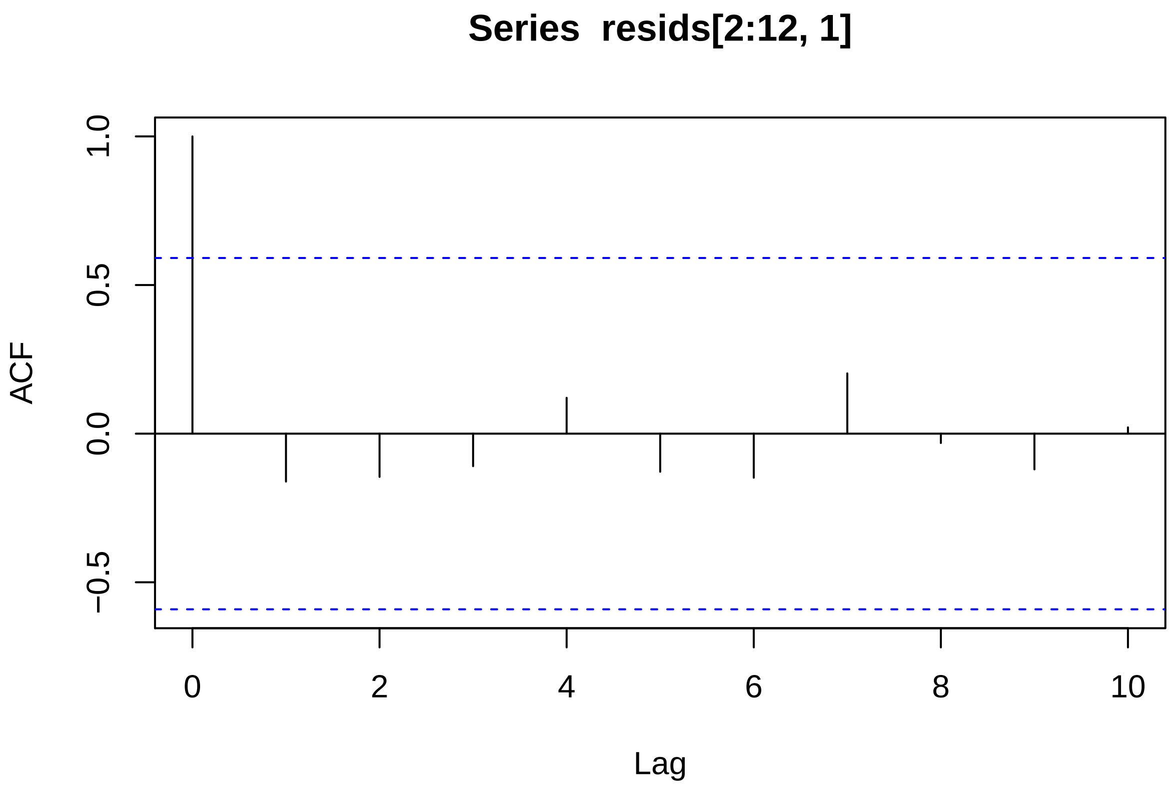
$$Var(Y_t|Y_{t-1}) = Var(\beta_0) + Var(\beta_1 Y_{t-1}) + Var(\epsilon_t)$$
$$Var(Y_t|Y_{t-1}) = Var(\epsilon_t)$$
$$Var(Y_t|Y_{t-1}) = \sigma^2$$

```
myCI <- function (x){
  list(Upper = x+2, Lower = x-2)
}
myPredicts <- mapply(oneStep, arimas$beta0, arimas$beta1, arimas$yt)
mapply(myCI, myPredicts)
```

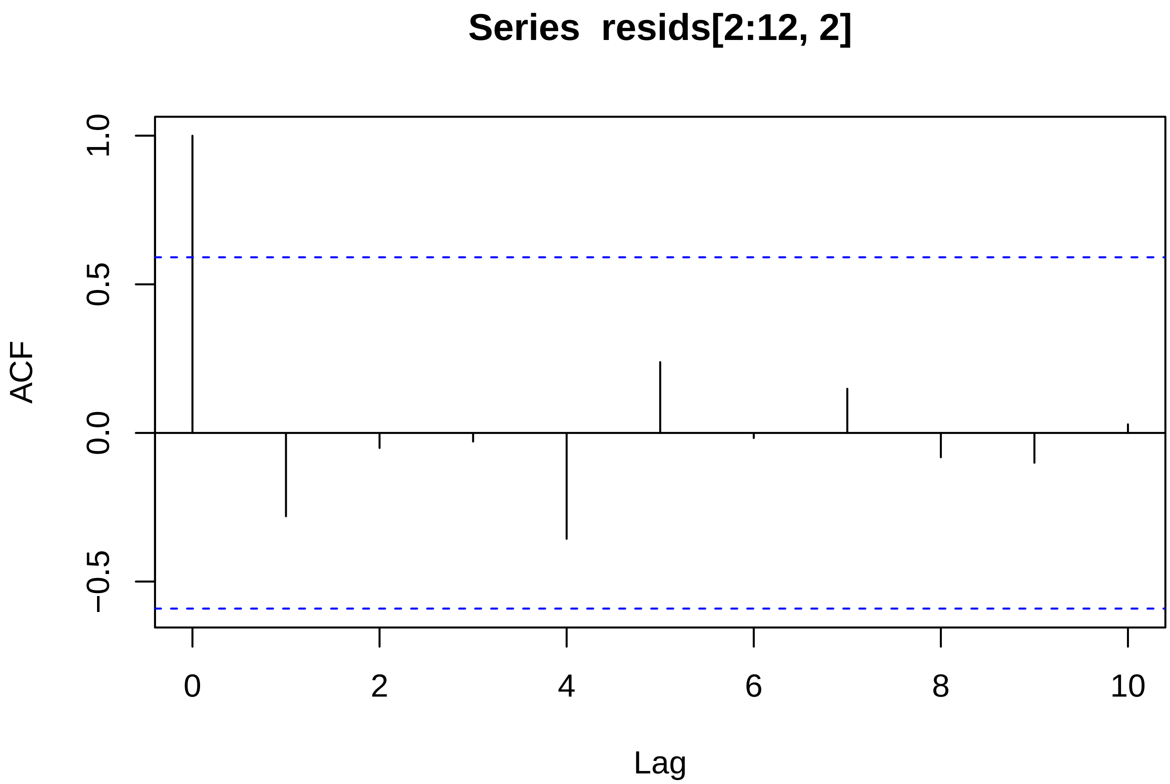
```
##       [,1] [,2]      [,3]      [,4]      [,5]
## Upper 2    3.783886  1.921522  1.442109  13.87453
## Lower -2   -0.216114 -2.078478 -2.557891 9.874529
```

g

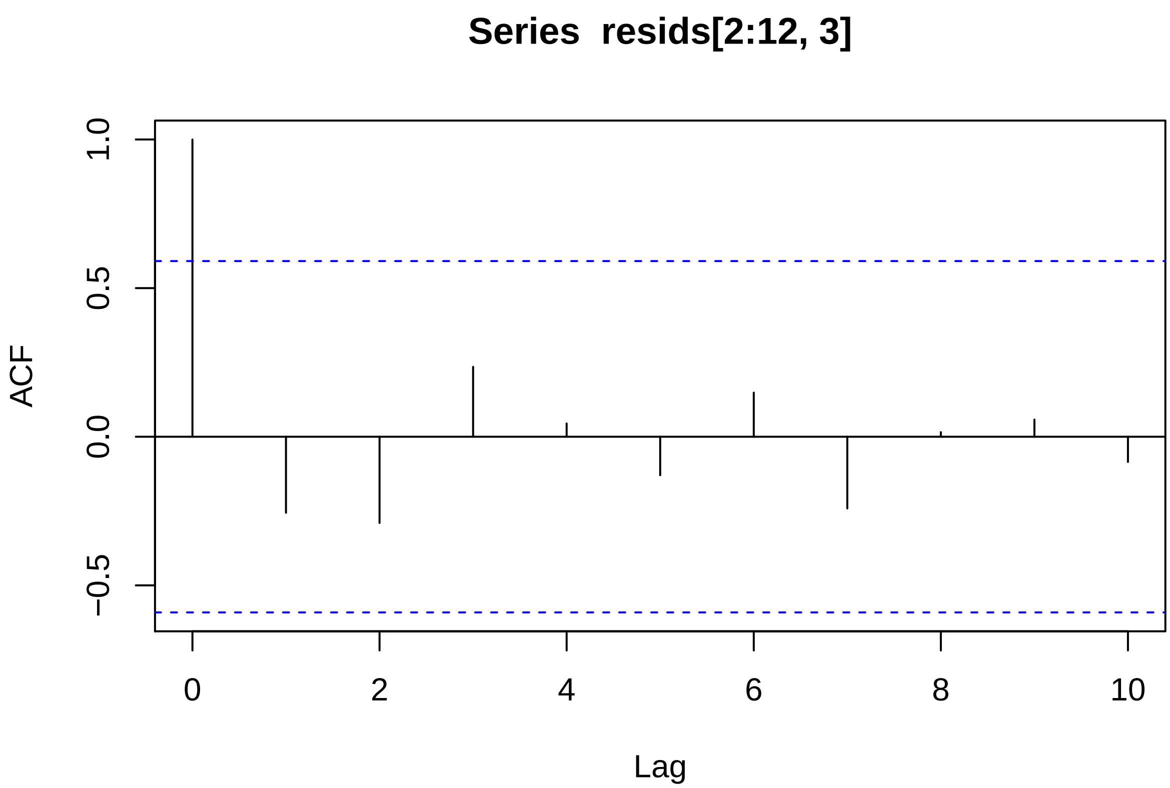
```r
resAR <- function(ar){
  resid(arima(ar, order = c(1,0,0)))
}
resids <- mapply(resAR, list(ar1, ar2, ar3, ar4, ar5))
acf(resids[2:12,1])
```



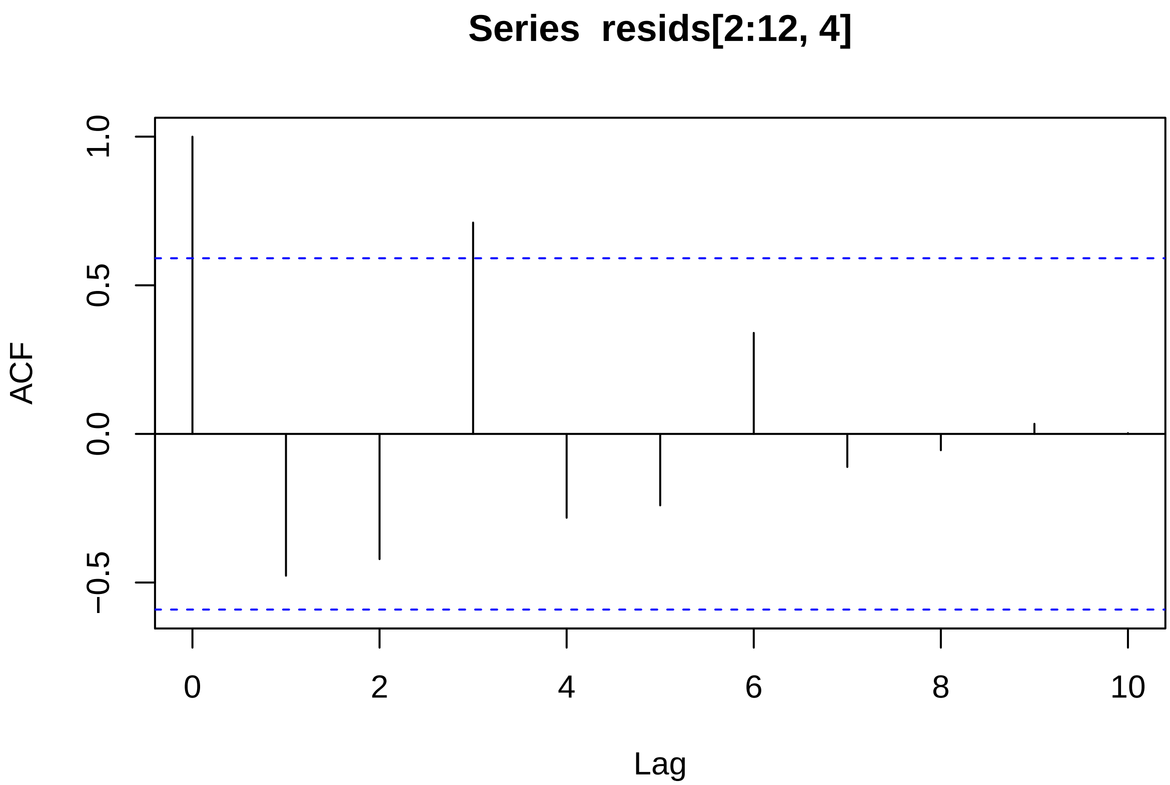

```r
acf(resids[2:12,2])
```



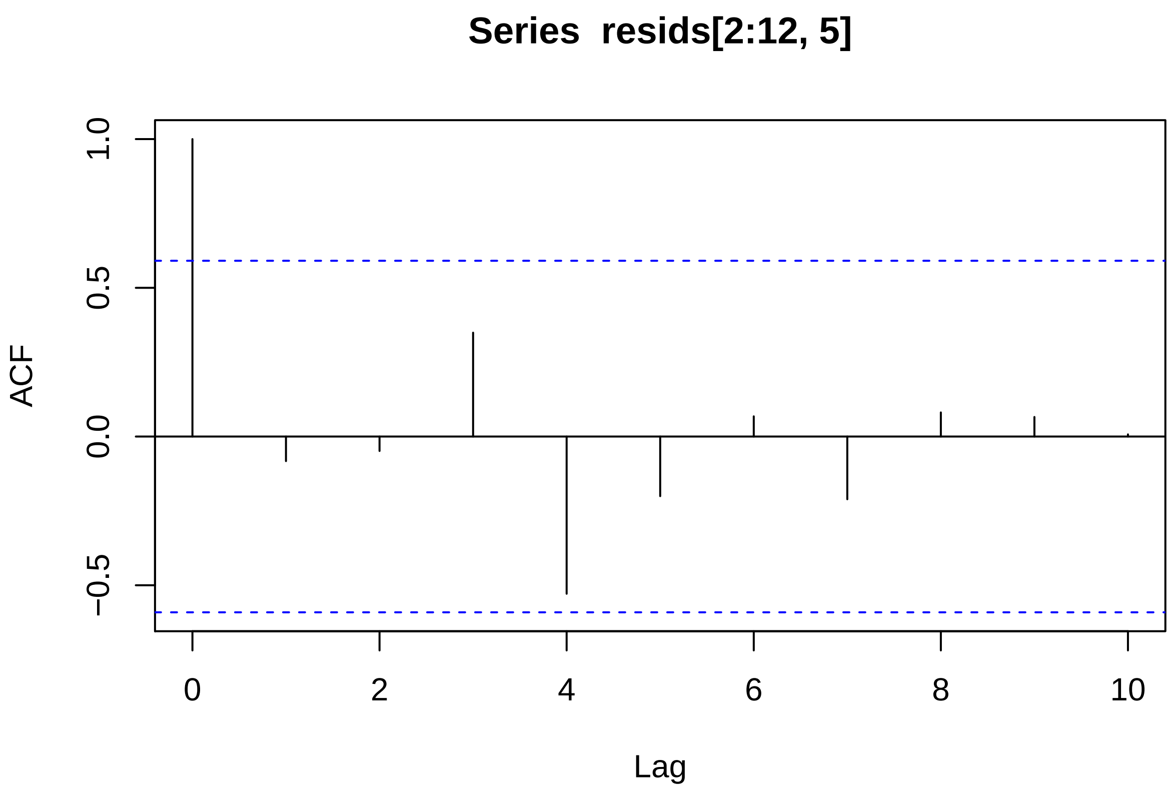

```
acf(resids[2:12,3])
```



```r
acf(resids[2:12,4])
```




```r
acf(resids[2:12,5])
```



Problem 2

a

```
ma1<-arima.sim(list(ma=c(.8)),n=200) + 1
ma2<-arima.sim(list(ma=c(-.8)),n=200)
```

```
# I assume that the first a) and b) on the homework are replicated in later steps, I'll leave the code 
# they aren't
plot(ma1, type = "l", ylim = c(-5,5), main = "Multiples Ma(1) Plot")
lines(ma2, col = "red")
acf(ma1, lag.max=10)
pacf(ma1, lag.max=10)
acf(ma2, lag.max=10)
pacf(ma2, lag.max=10)
```

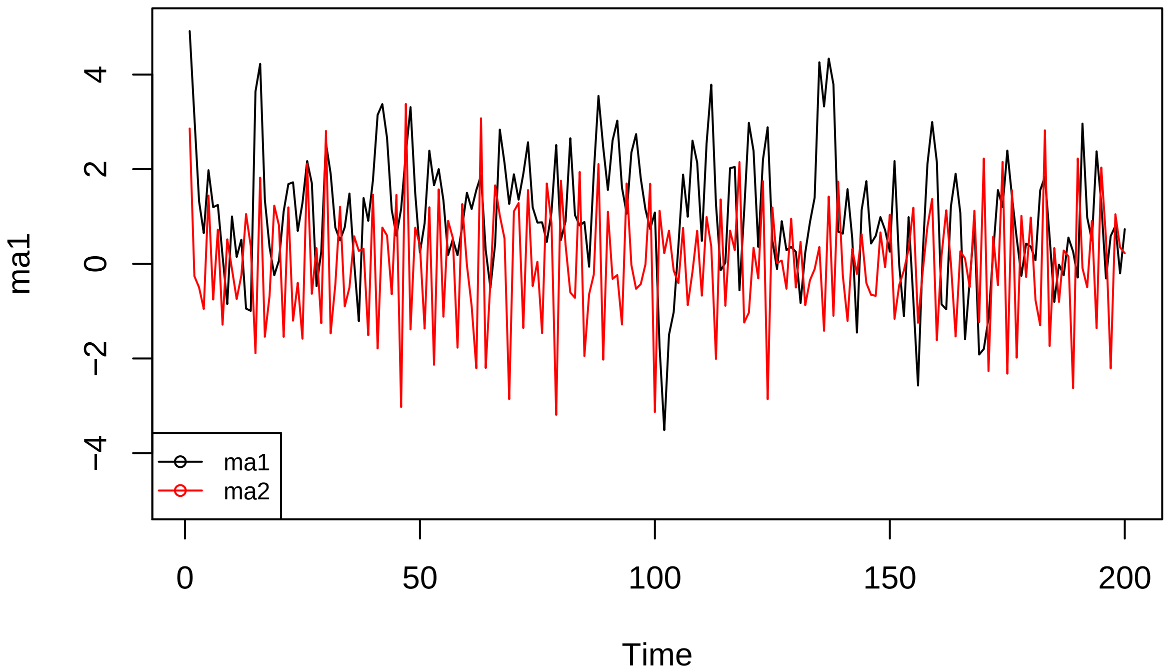
a

$$
\begin{aligned}
\mathrm{E}[y_t] &= \mathrm{E}[\phi_0 + \theta_1\epsilon_{t-1} + \epsilon_t] \\
&= \phi_0 + \theta_1\mathrm{E}[\epsilon_{t-1}] + \mathrm{E}[\epsilon_t] \\
&= \phi_0 + \theta_1 0 + 0 \\
&= \phi_0.
\end{aligned}
$$

Therefore, the unconditional mean for ma1 = 1 and the unconditional mean for ma2 = 0

b

```
plot(ma1, type = "l", ylim = c(-5,5))
lines(ma2, col = "red")
legend("bottomleft", c("ma1","ma2"), pch=1, col=c('black', 'red'), lty=1, cex=.75)
```



c

Yes, each of the plots look the way that I expect because their means are centered around their expected value and their swing is proportional to their θ value.

d

```
predict(ma1)$mean[1]
```

```
## [1] 0.4884332
```

```
predict(ma2)$mean[1]
```

```
## [1] -0.01359228
```

e

```
maOneStep <- function(mean, theta, prev){
  mean + theta*prev
}

maOneStep(1, .8, ma1[200])
```

```
## [1] 1.585862
```

```
maOneStep(0, -.8, ma2[200])
```

```
## [1] -0.1784449
```

f

```r
maCI <- function(mean, theta, prev, p){
  cur <- maOneStep(mean, theta, prev)
  upper<-p$upper[1,2]
  lower<-p$lower[1,2]
  return(data.frame(forecast = cur, upr = upper, lwr = lower))
}

maCI(1, .8, ma1[200], predict(ma1))
```

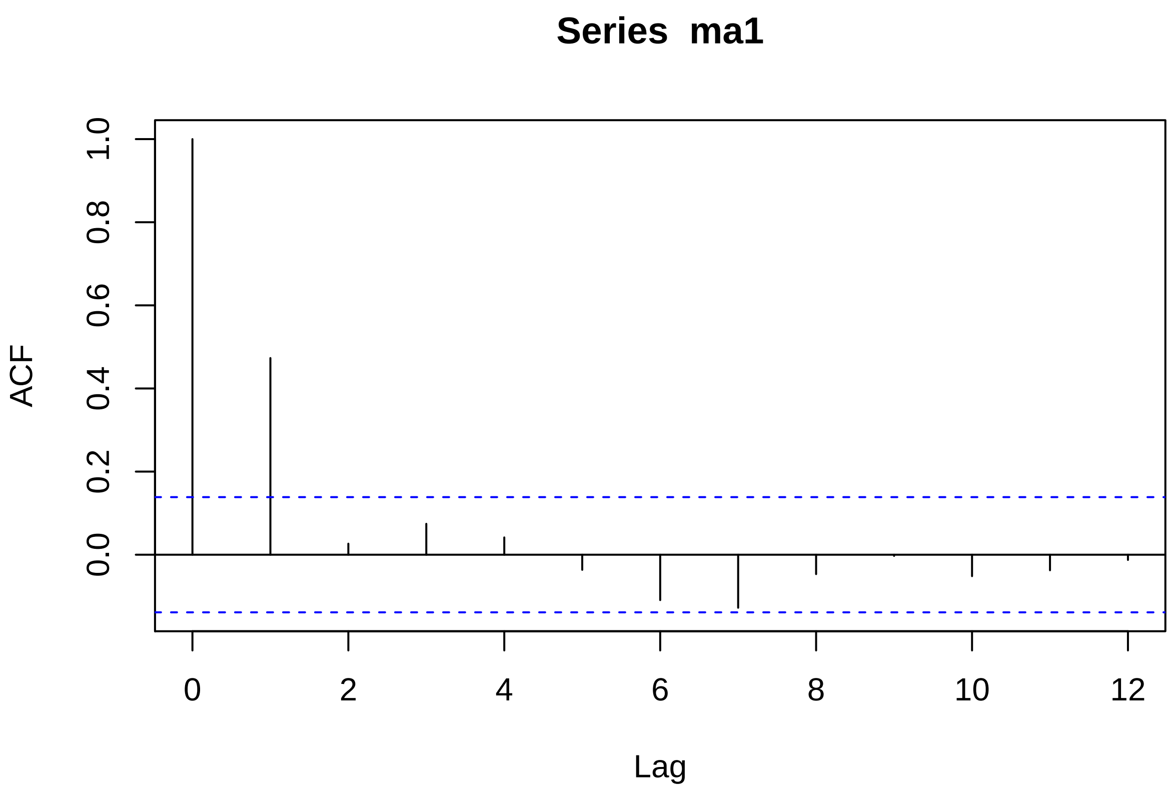
```
##   forecast      upr       lwr
## 1 1.585862 3.063392 -2.086526
```

```r
maCI(0, -.8, ma2[200], predict(ma2))
```

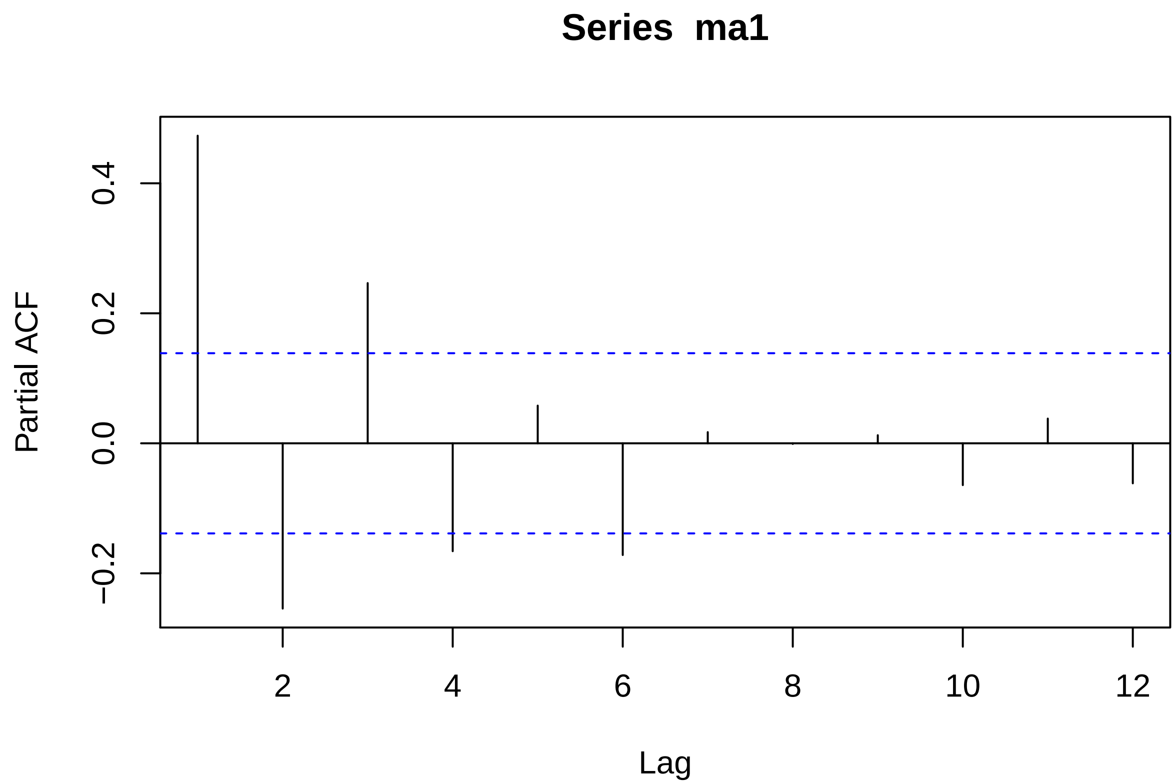
```
##     forecast     upr       lwr
## 1 -0.1784449 2.52917 -2.556355
```

g

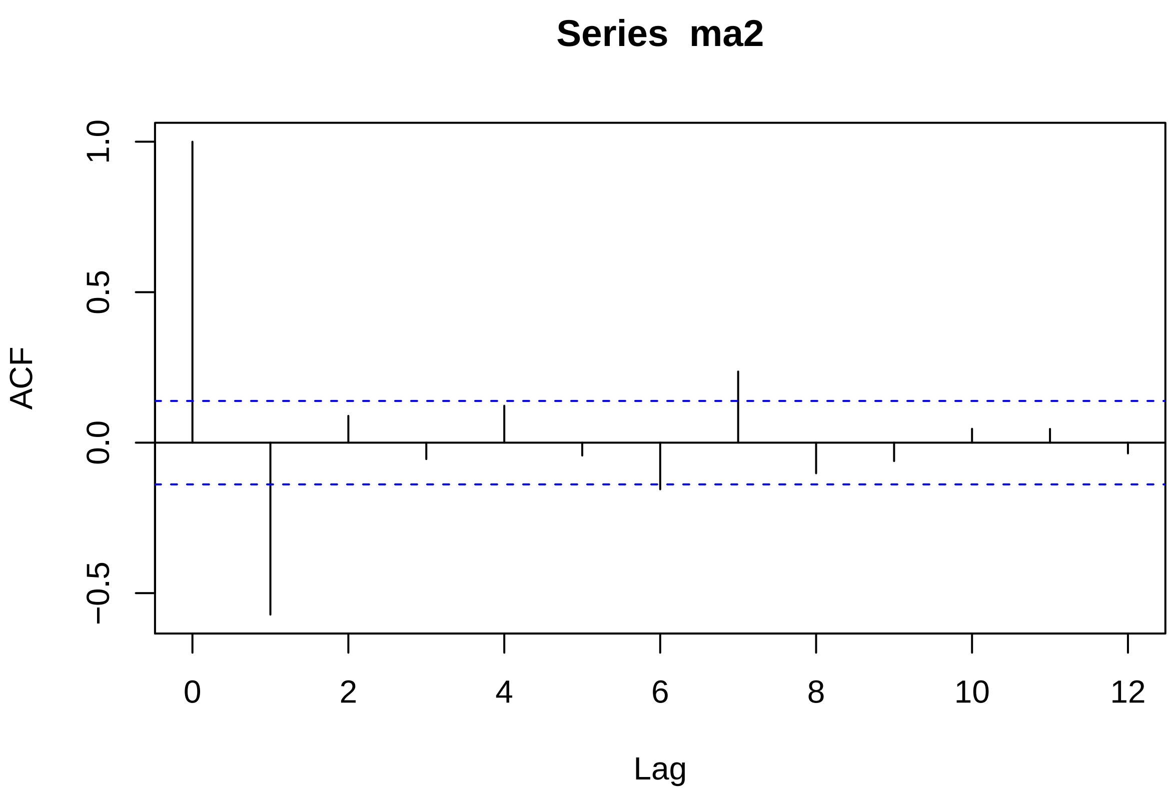
```r
acf(ma1, lag.max=12)
```



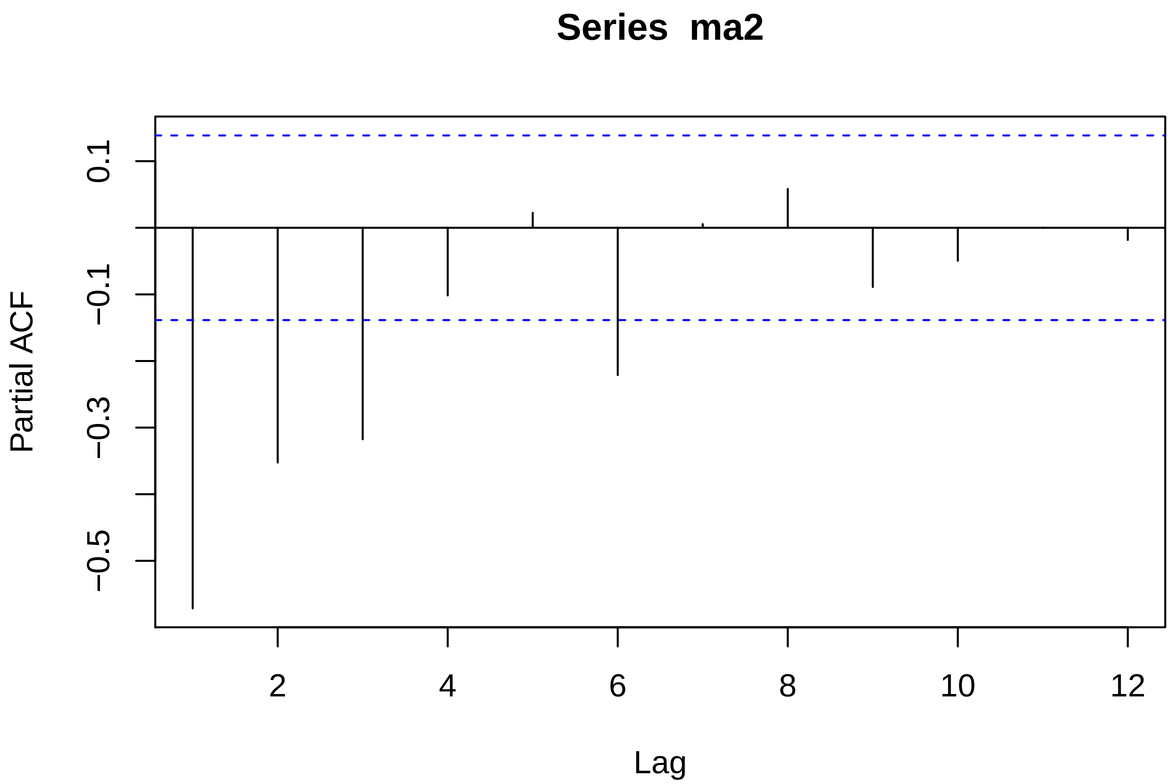

```r
pacf(ma1, lag.max=12)
```




```r
acf(ma2, lag.max=12)
```

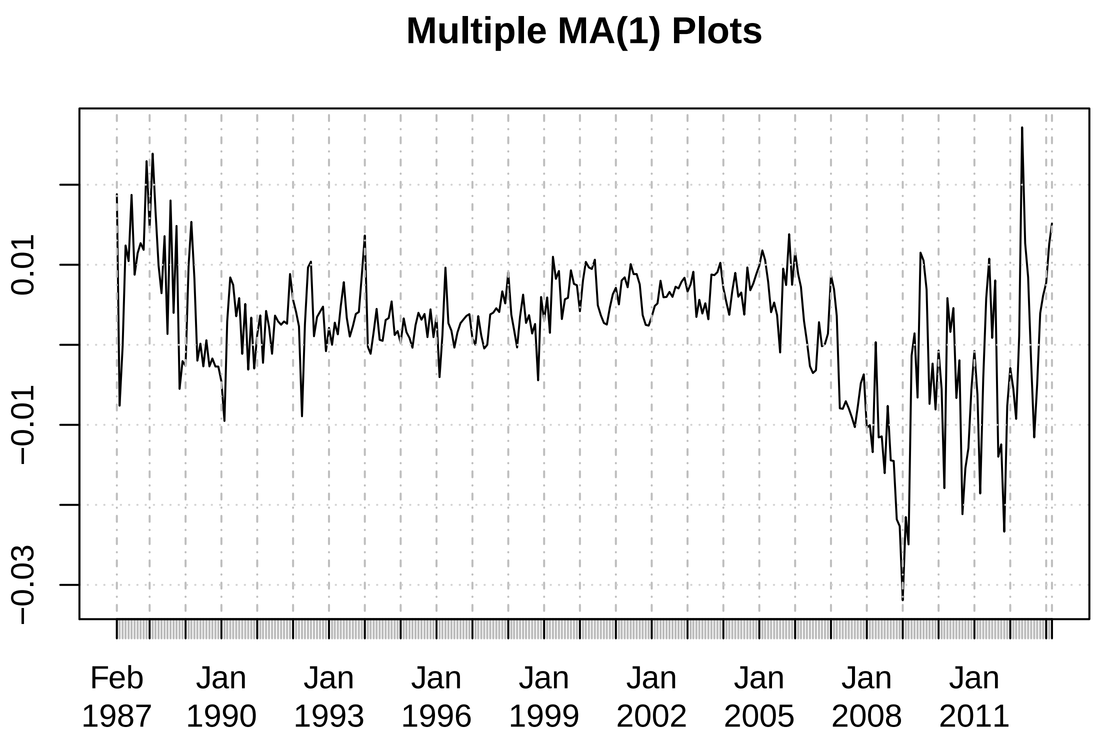


```r
pacf(ma2, lag.max=12)
```



Problem 3

```r
hpc <- read.xls("hpchicago.xls", skip = 1)
hpc <- subset(hpc, select = c(YEAR, CHXR, ret_raw, CHXR.SA, ret_sa))
hpc <- hpc[complete.cases(hpc),]
hpcXts <- xts(subset(hpc, select = c(ret_sa)), order.by = as.Date(as.yearmon(hpc$YEAR)))
plot(as.xts(hpcXts), main = "Multiple MA(1) Plots")
```



a

```r
ar1 <- arima(hpc$ret_sa, order = c(1,0,0))
summary(ar1)
```

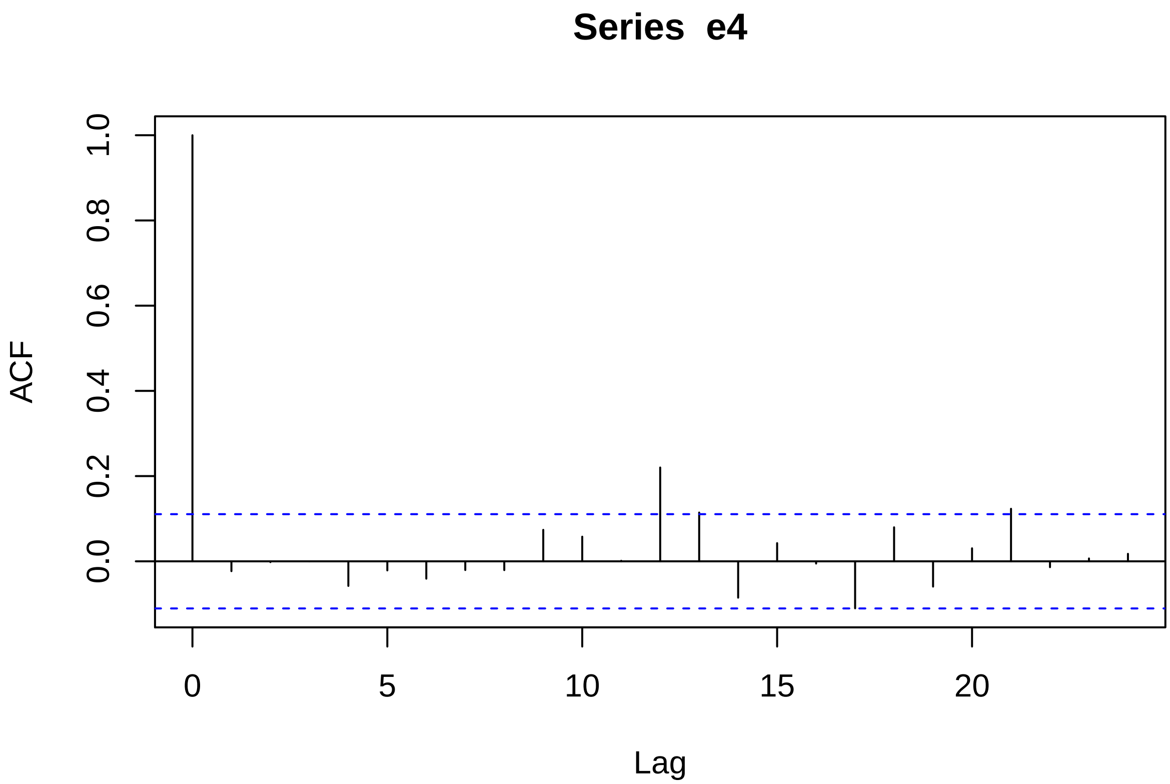
```
##
## Call:
## arima(x = hpc$ret_sa, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.6797     0.0027
## s.e.  0.0419     0.0010
##
## sigma^2 estimated as 3.404e-05:  log likelihood = 1169.35,  aic = -2332.69
##
## Training set error measures:
##                         ME       RMSE         MAE  MPE MAPE      MASE
## Training set -4.631477e-05 0.00583453 0.004084536 -Inf  Inf 0.9346362
##                    ACF1
## Training set -0.1418769
```

b

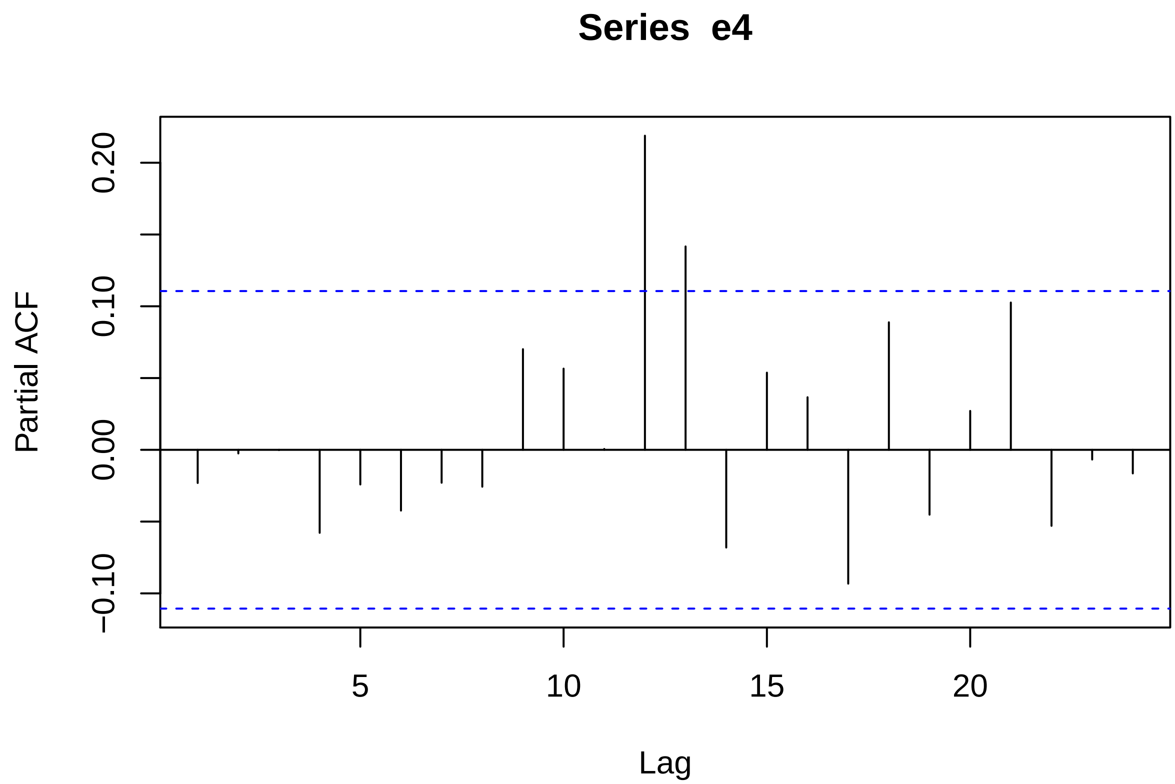
```r
ar4 <- arima(hpc$ret_sa, order = c(4,0,0))
```

```r
# Used for testing but not printed here because their are so many plots on this assignment
acf(hpc$ret_sa)
pacf(hpc$ret_sa)
```

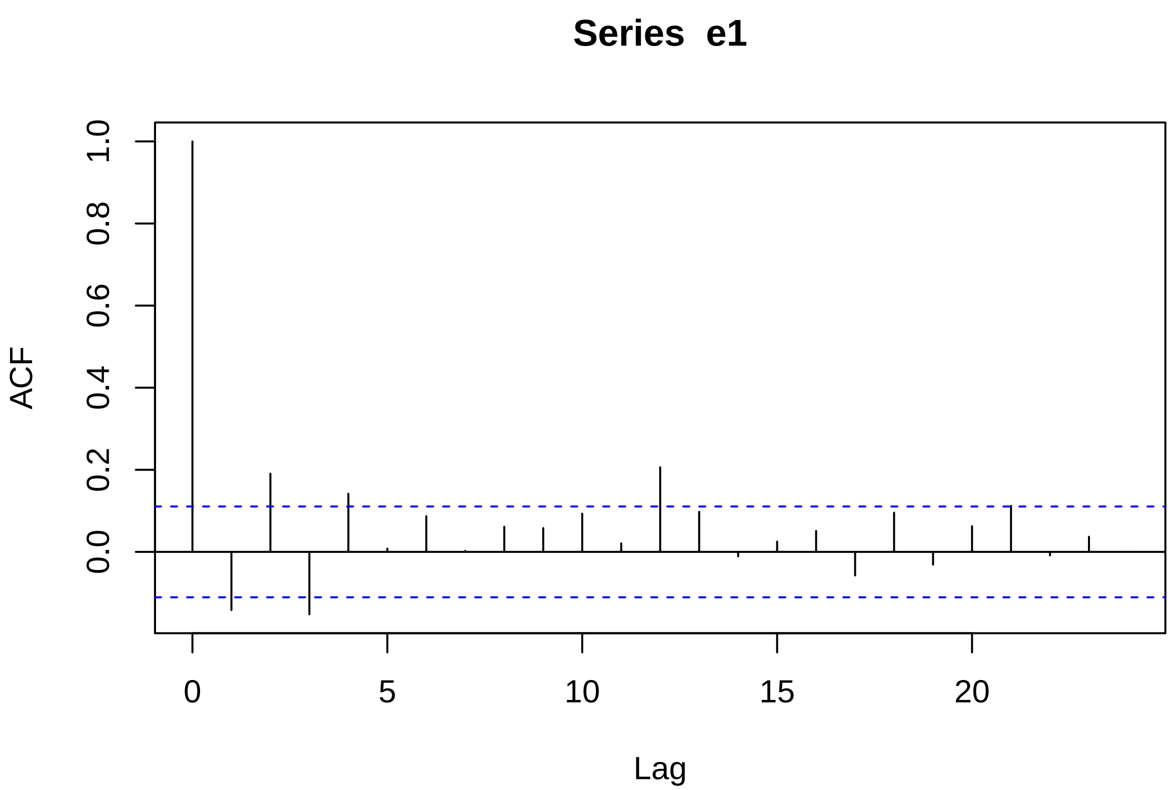
```r
e4 <- resid(ar4)
acf(e4)
```



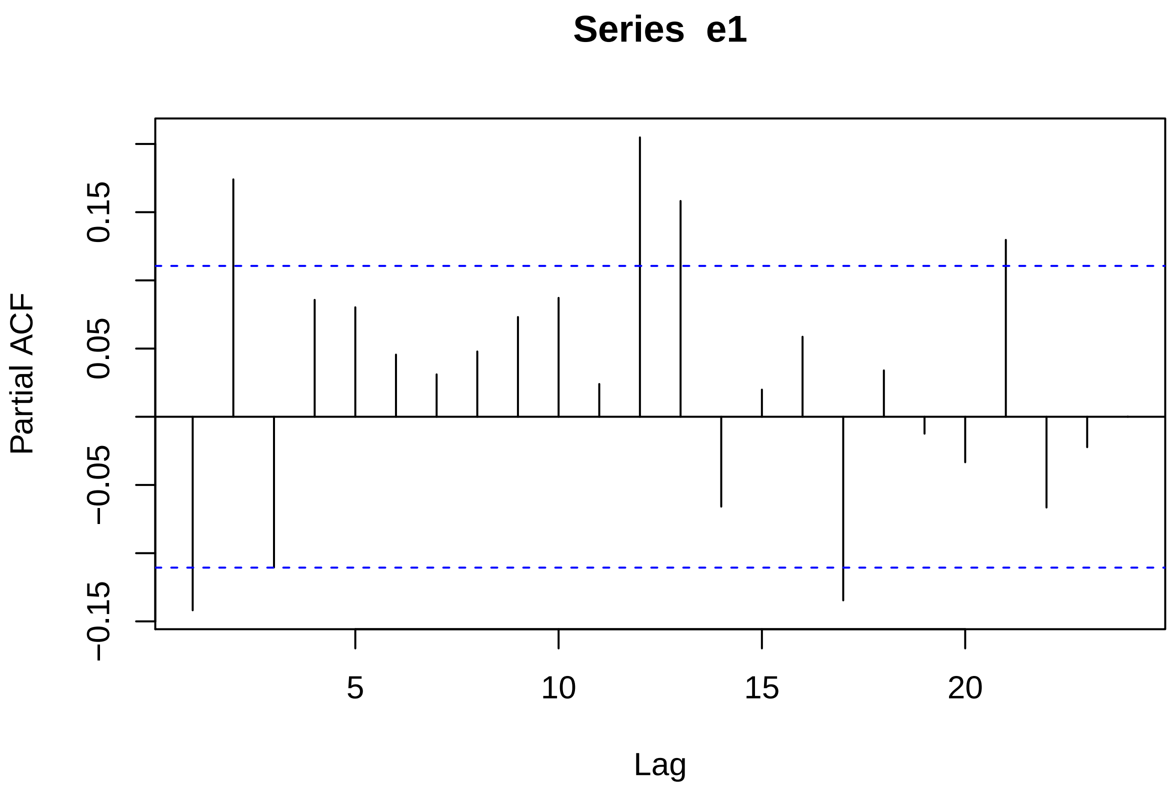

```r
pacf(e4)
```




```r
e1 <- resid(ar1)
acf(e1)
```



```r
pacf(e1)
```



For this problem, I looked at the acf and pacf plots and compared the auto-correlation and partial auto-correlation of the residulas. Because we do not know any good test statistics for time series, I tried to visualally minimize the correlations in the residuals and ar4 was clearly the best.

c

```r
1 - var(resid(ar4)) / var(fitted(ar4) +resid(ar4)) * ((length(fitted(ar4)) - 1) /  (length(fitted(ar4))
```

```
## [1] 0.4972696
```

```r
1 - var(resid(ar1)) / var(fitted(ar1) + resid(ar1)) * ((length(fitted(ar4)) - 1) /  (length(fitted(ar4)
```

```
## [1] 0.4530995
```

The AR(4) model out performs the AR(1) as measured by R^2

d

```r
par4 <- predict(ar4)
par1 <- predict(ar1)
par4$pred
```

```
## Time Series:
## Start = 315
## End = 315
## Frequency = 1
## [1] 0.01165995
```

```
par1$pred
```

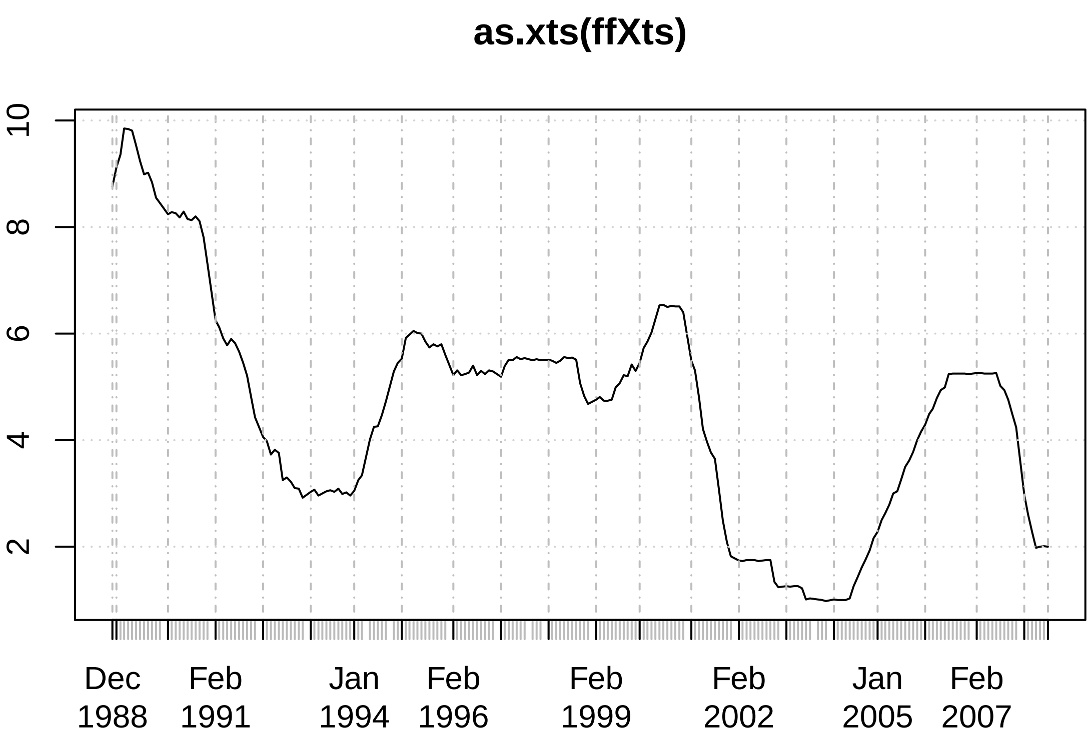
```
## Time Series:
## Start = 315
## End = 315
## Frequency = 1
## [1] 0.01115224
```

Problem 4

```
ff <- read.xls("FedFunds.xls")
ff <- rename(x = ff, c("DATE......"="date", "X.FFO"="ffo"))
```

a

```
ffXts = xts(x=ff$ffo, order.by=as.Date(ff$date, "%Y-%m-%d"))
plot(as.xts(ffXts))
```




```
cat("Sample Mean: ", mean(ff$ffo), "Standard Error: ", sd(ff$ffo)/sqrt(length(ff$ffo)))
```

```
## Sample Mean:  4.606986 Standard Error:  0.1430624
```

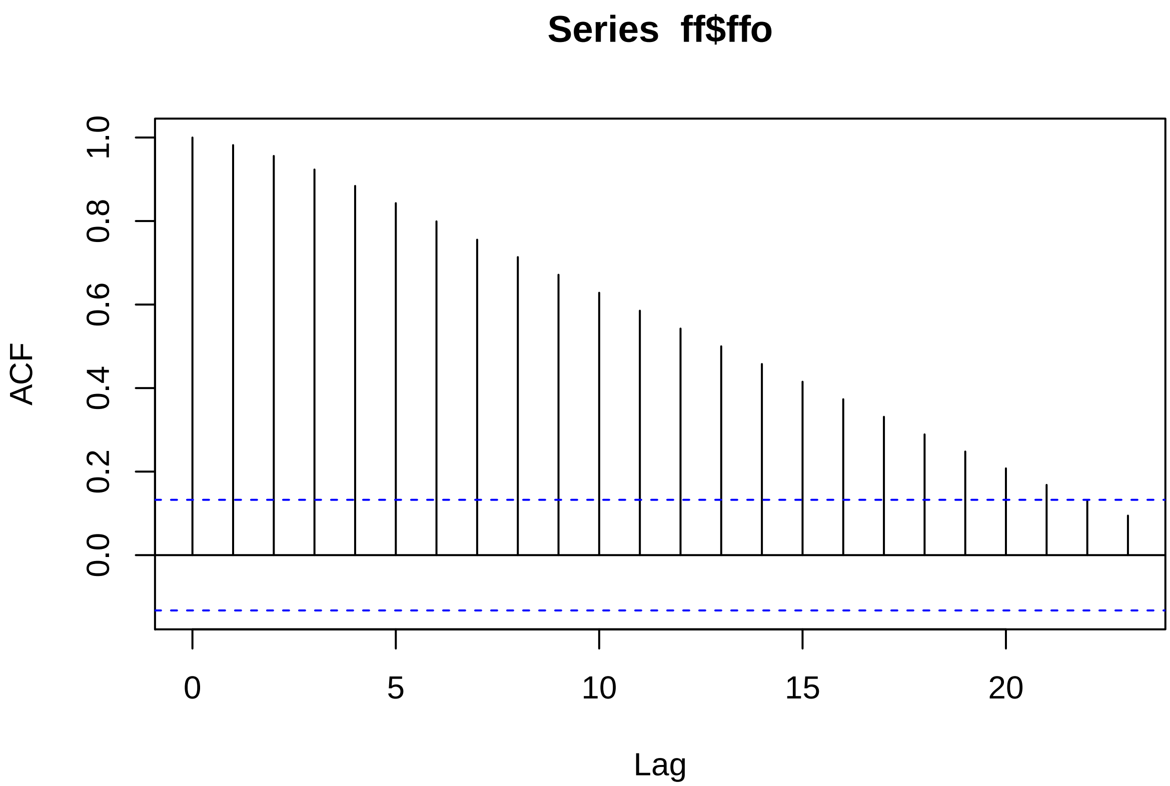
b

```r
cat("Sample Variance: ", var(ff$ffo))
```

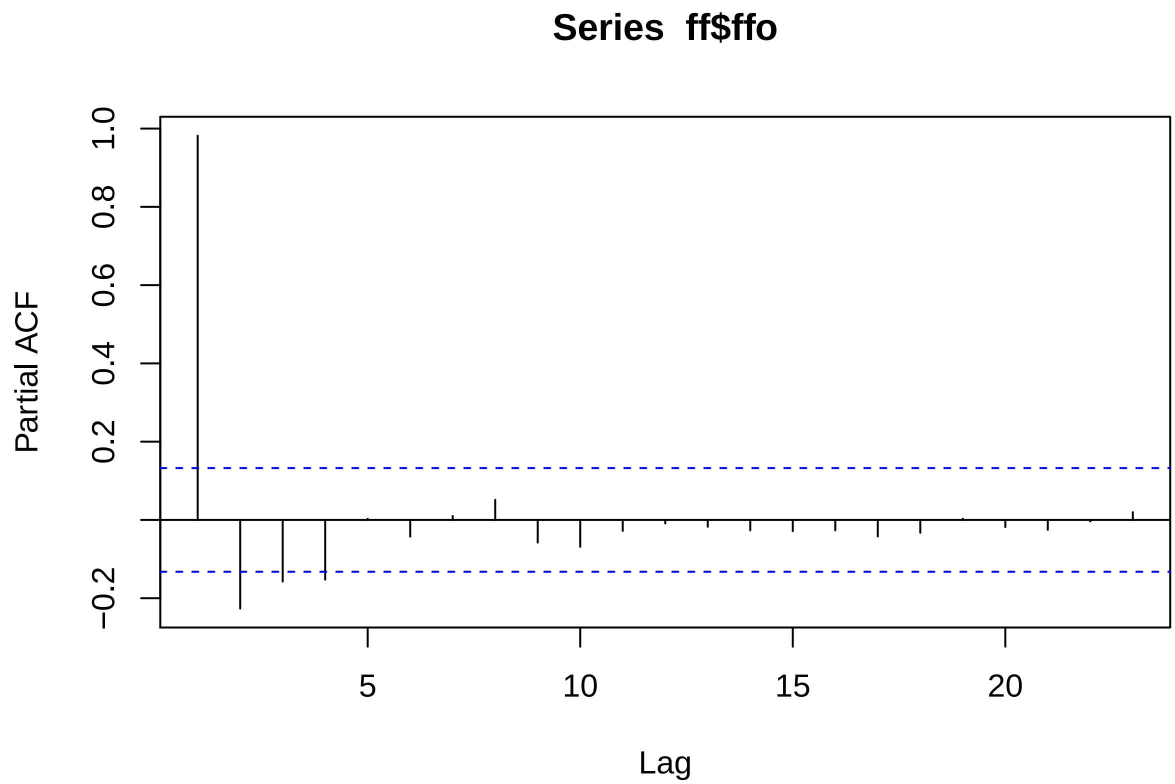
```
## Sample Variance:  4.482243
```

c

```r
acf(ff$ffo)
```




```r
pacf(ff$ffo)
```




```r
ar1 <- arima(ff$ffo, order = c(4,0,0))
arMa1 <- arima(ff$ffo, order = c(2,0,0))
```

```r
summary(ar1)
```

```
##
## Call:
## arima(x = ff$ffo, order = c(4, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3      ar4  intercept
##       1.3600  -0.1496  -0.0807  -0.1431     4.5878
## s.e.  0.0666   0.1137   0.1140   0.0684     0.8454
##
## sigma^2 estimated as 0.03322:  log likelihood = 59.29,  aic = -106.58
##
## Training set error measures:
##                         ME      RMSE       MAE        MPE     MAPE
## Training set -0.007134788 0.1822499 0.1180297 -0.4360891 3.114063
##                   MASE        ACF1
## Training set 0.8040775 0.004547313
```

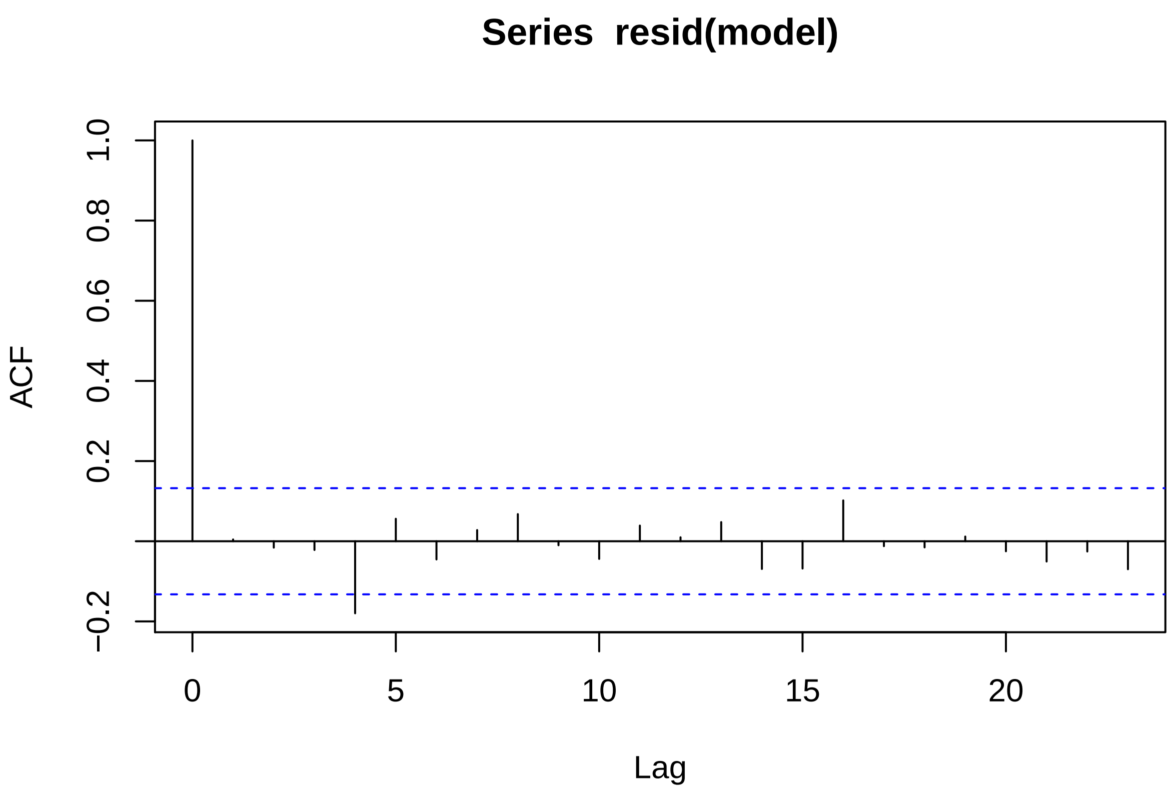
```r
summary(arMa1)
```

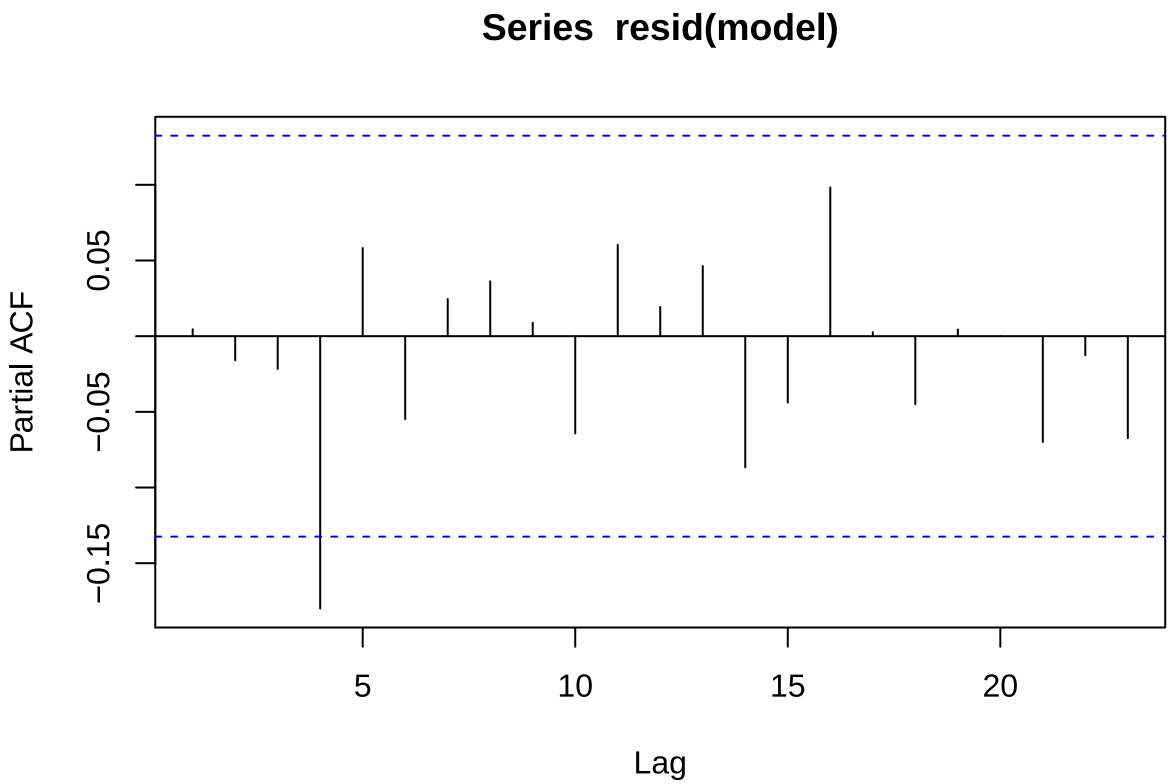
```
##
## Call:
## arima(x = ff$ffo, order = c(2, 0, 0))
```

```
##
## Coefficients:
##          ar1      ar2  intercept
##       1.5456  -0.5528     4.7800
## s.e.  0.0567   0.0573     1.4468
##
## sigma^2 estimated as 0.03662:  log likelihood = 48.67,  aic = -89.34
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.01385966 0.1913725 0.1262976 -0.5529476 3.291726 0.8604022
##                    ACF1
## Training set -0.1553034
```

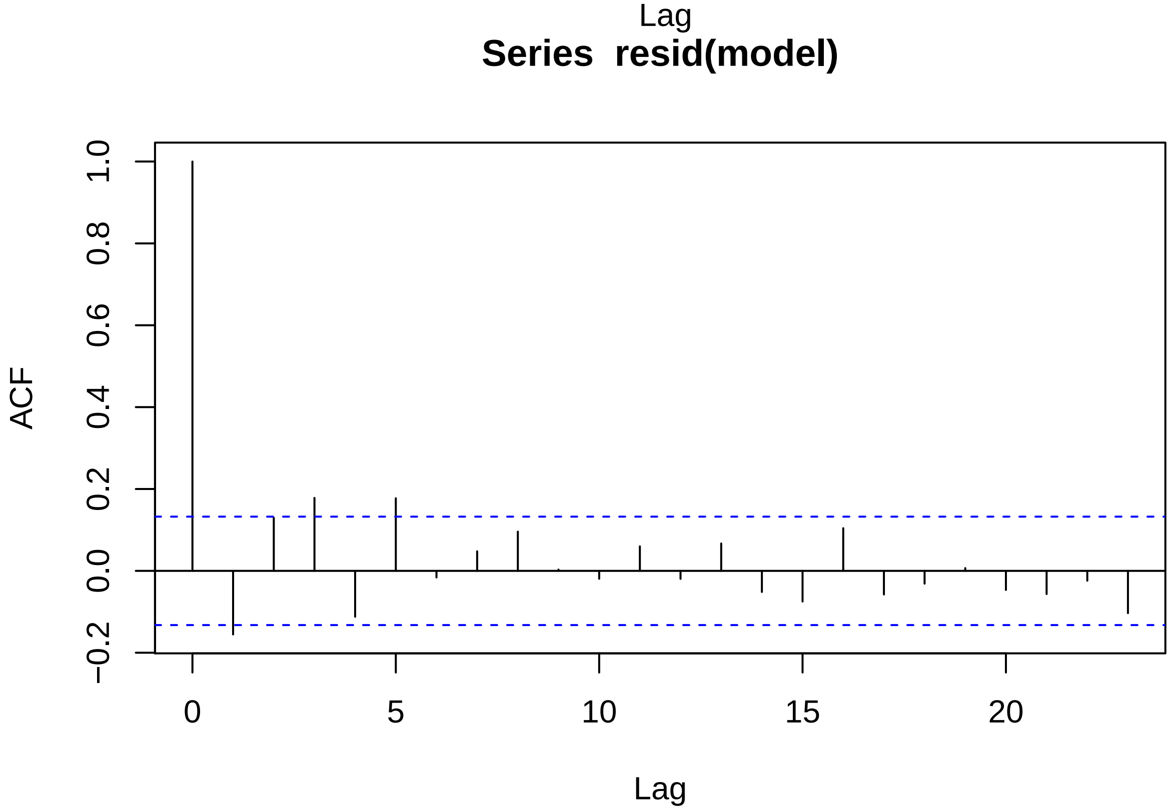
d

```
acfPacfResid <- function (model){
  acf(resid(model))
  pacf(resid(model))
}
mapply(acfPacfResid, list(ar1,arMa1))
```

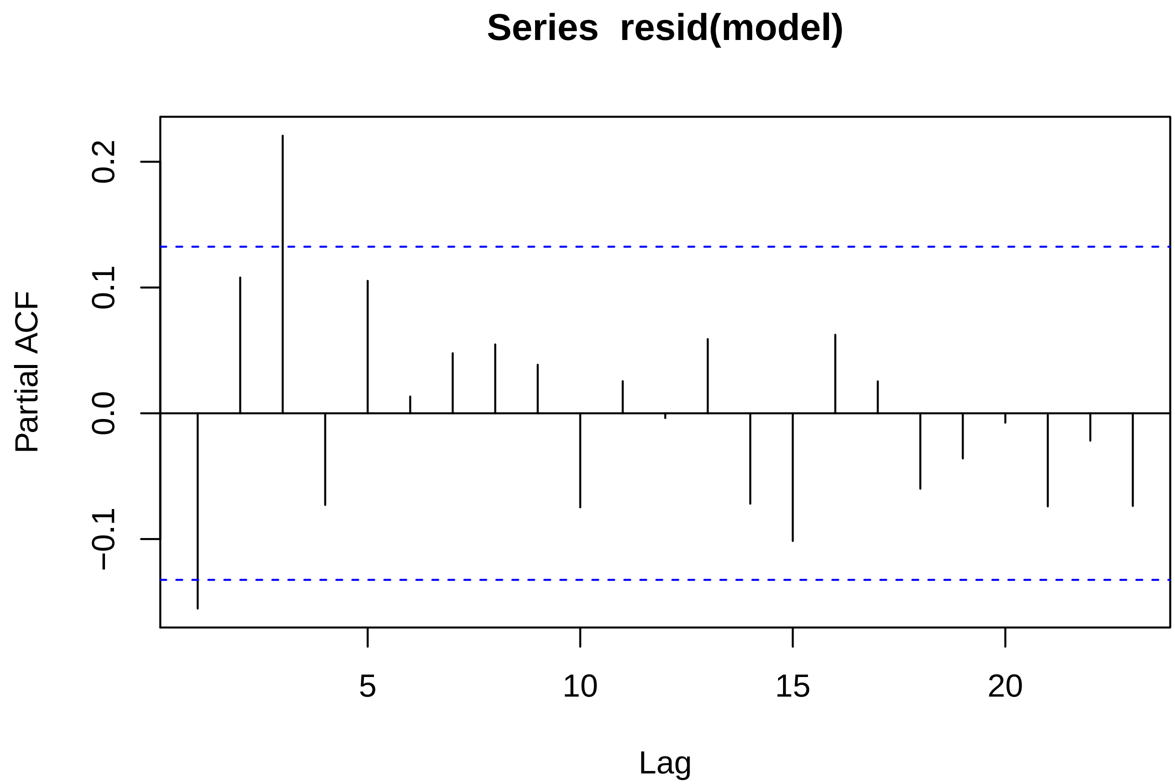




Series resid(model)
Partial ACF
0.05
−0.05
−0.15
5
10
15
20
Lag



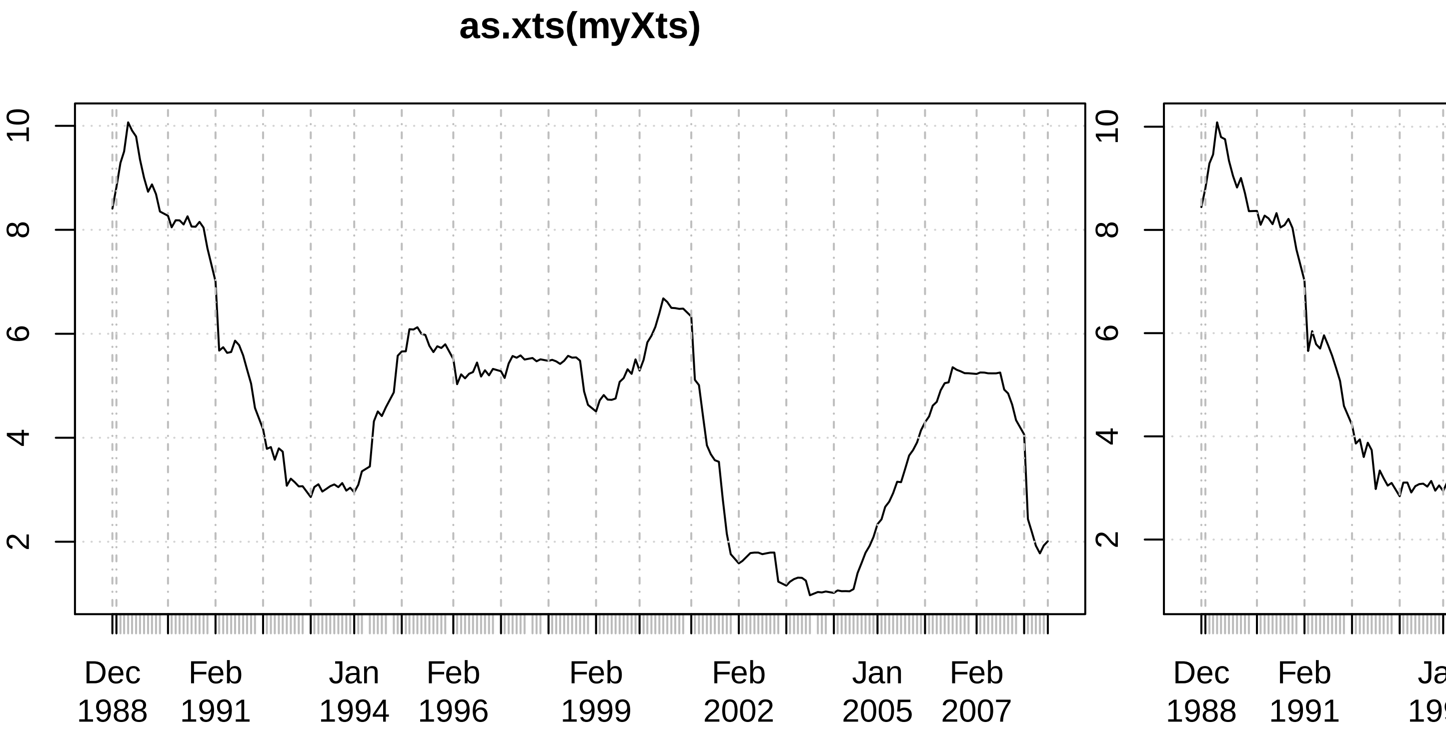
Series resid(model)
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
Lag




```
##        [,1]            [,2]
## acf    Numeric,23      Numeric,23
## type   "partial"       "partial"
## n.used 219             219
## lag    Numeric,23      Numeric,23
## series "resid(model)" "resid(model)"
## snames NULL            NULL
```

e

```
xtsMe <- function(me){
  myXts = xts(x=(ff$ffo - me$residuals), order.by=as.Date(ff$date, "%Y-%m-%d"))
  plot(as.xts(myXts))
}
mapply(xtsMe, list(ar1, arMa1))
```

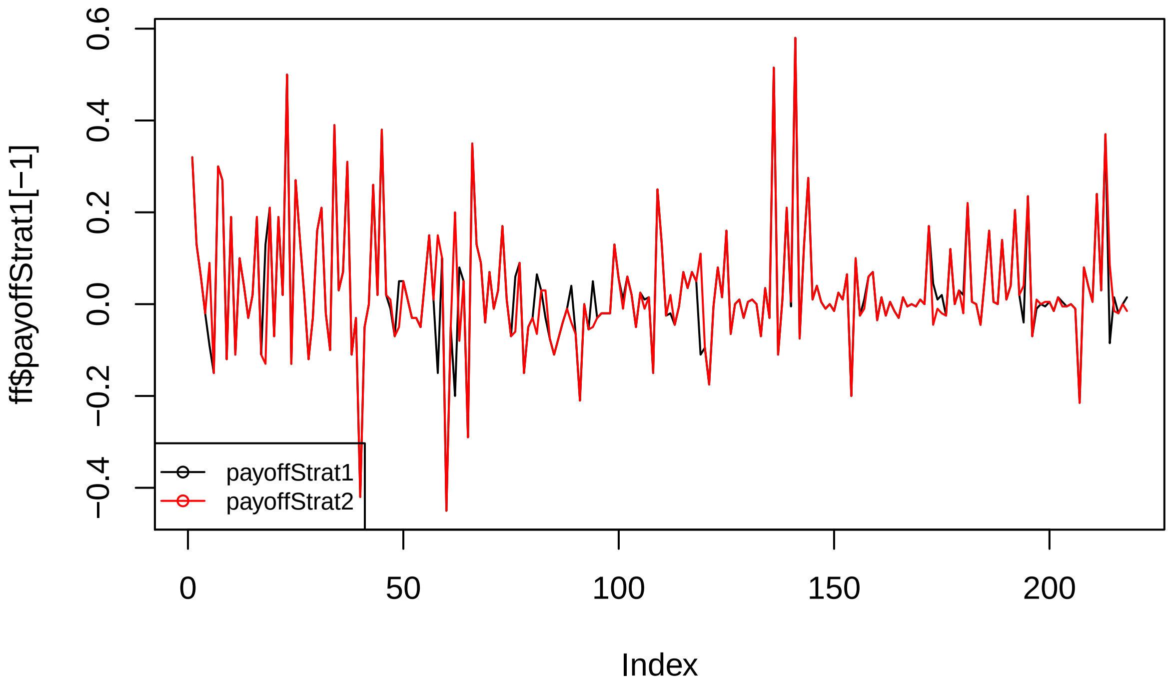



```
##      [,1]      [,2]
## [1,] List,12   List,12
## [2,] Raw,35992 Raw,35992
## [3,] NULL      NULL
```

Problem 5

a

```
ff$ar4Pred <- fitted(ar1)
ff$arMaPred <- fitted(arMa1)

payoffSitu1 <- ff$ffo[-1] - ff$Future[-1]
payoffSitu2 <- ff$Future[-1] - ff$ffo[-1]

myDiffSitu1 <- as.numeric(ff$ar4Pred[-1] - ff$Future[-1] > 0)
myDiffSitu2 <- as.numeric(ff$ar4Pred[-1] - ff$Future[-1] <= 0)

myDiffSitu12 <- as.numeric(ff$arMaPred[-1] - ff$Future[-1] > 0)
myDiffSitu22 <- as.numeric(ff$arMaPred[-1] - ff$Future[-1] <= 0)

payoffStrat1 <- c(NA, myDiffSitu1 * payoffSitu1 + myDiffSitu2 * payoffSitu2)
payoffStrat2 <- c(NA, myDiffSitu12 * payoffSitu1 + myDiffSitu22 * payoffSitu2)

ff <- cbind(ff, payoffStrat1, payoffStrat2)

plot(ff$payoffStrat1[-1], type = "l")
lines(ff$payoffStrat2[-1], col = "red")
legend("bottomleft", c("payoffStrat1","payoffStrat2"), pch=1, col=c('black', 'red'), lty=1, cex=.75)
```



b

```
kable(head(ff))
```

date	ffo	Future	ar4Pred	arMaPred	payoffStrat1	payoffStrat2
1988-12-01	8.76	NA	8.407434	8.441712	NA	NA
1989-01-01	9.12	8.80	8.819827	8.804694	0.32	0.32
1989-02-01	9.36	9.23	9.287104	9.287896	0.13	0.13
1989-03-01	9.85	9.91	9.507284	9.459843	0.06	0.06
1989-04-01	9.84	9.86	10.067184	10.084525	-0.02	-0.02
1989-05-01	9.81	9.90	9.909377	9.798209	-0.09	0.09

```
cat("Mean ar4: ", mean(ff$payoffStrat1[-1]), "Sd ar4: ", sd(ff$payoffStrat1[-1]))
```

```
## Mean ar4:  0.02885321 Sd ar4:  0.1319512
```

```
cat("Mean arMa: ", mean(ff$payoffStrat2[-1]), "Sd arMa: ", sd(ff$payoffStrat2[-1]))
```

```
## Mean arMa:  0.02995413 Sd arMa:  0.1317045
```

c

```
cat("Test Stat ar4: ",mean(ff$payoffStrat1[-1])/(sd(ff$payoffStrat1[-1])/sqrt(length(ff$payoffStrat1[-1
```

```
## Test Stat ar4:  3.228561
```

```r
cat("Test Stat arMa: ", mean(ff$payoffStrat2[-1])/(sd(ff$payoffStrat2[-1])/sqrt(length(ff$payoffStrat2[-
```

```
## Test Stat arMa:  3.358028
```

For both of these strategies, I reject the null hypothesis that their returns could be random. (Yay, profit!)

d

```r
d <- ff$payoffStrat1[-1] - ff$payoffStrat2[-1]

mean(d)/(sd(d)/sqrt(length(d)))
```

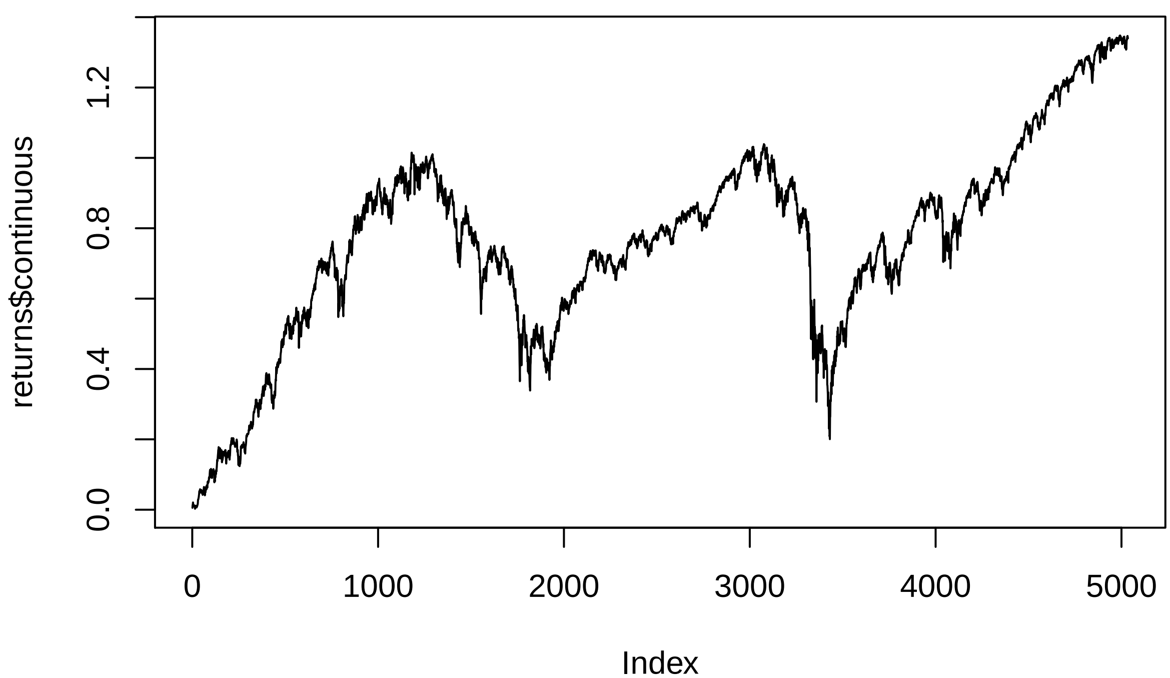
```
## [1] -0.326534
```

I fail to reject the null that the average returns for the two models are the null.

Problem 6

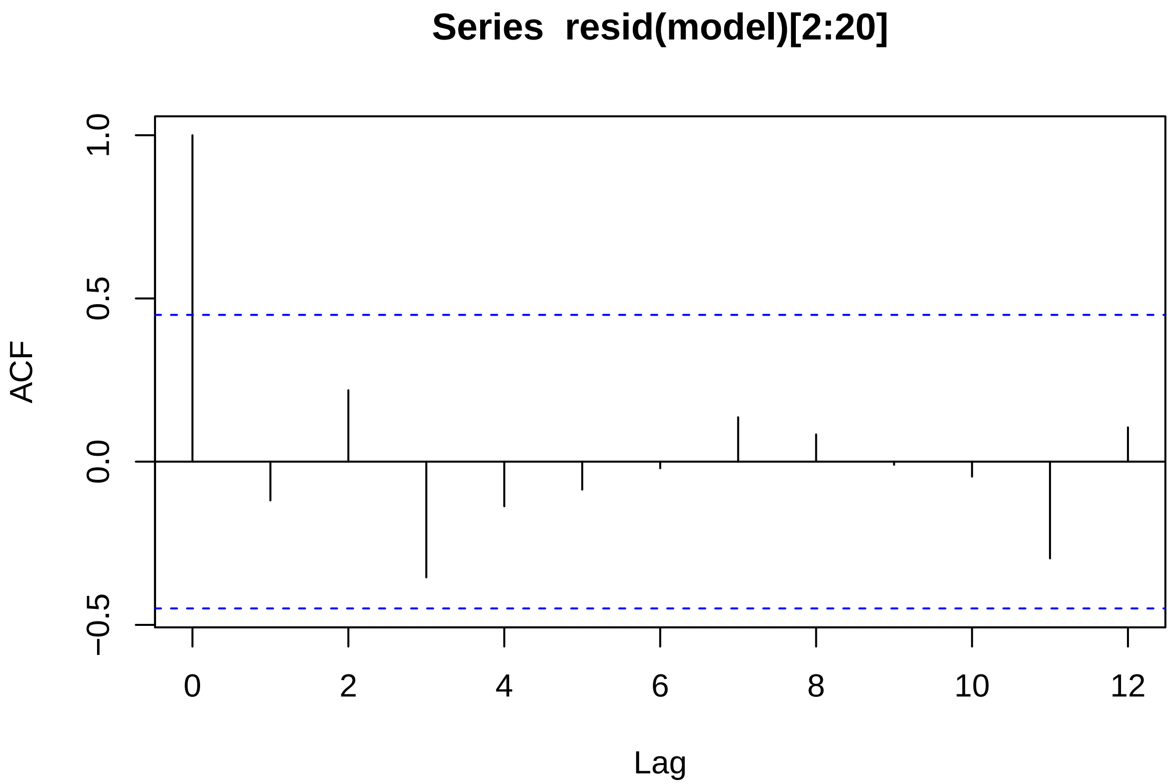
a

```r
returns <- read.csv("table.csv")
returns$DateTime <- as.Date(returns$Date, "%Y-%m-%d")
returns <- returns[order(returns$DateTime, decreasing=F),]

returns <- returns[complete.cases(returns),]

returns$continuous <- log(returns$Close) - log(returns$Open[1])

plot(returns$continuous, type = "l")
```

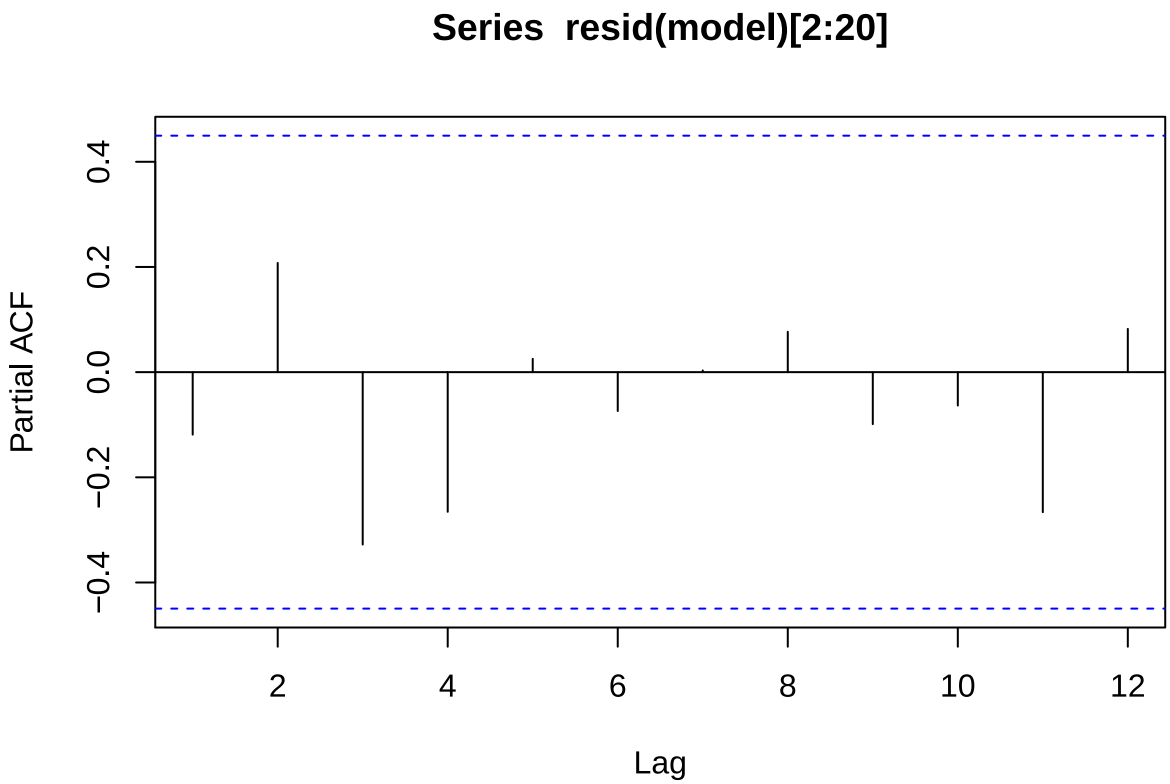


b

```r
model <- arima(returns$continuous, order = c(1,0,0))
acf(resid(model)[2:20])
```



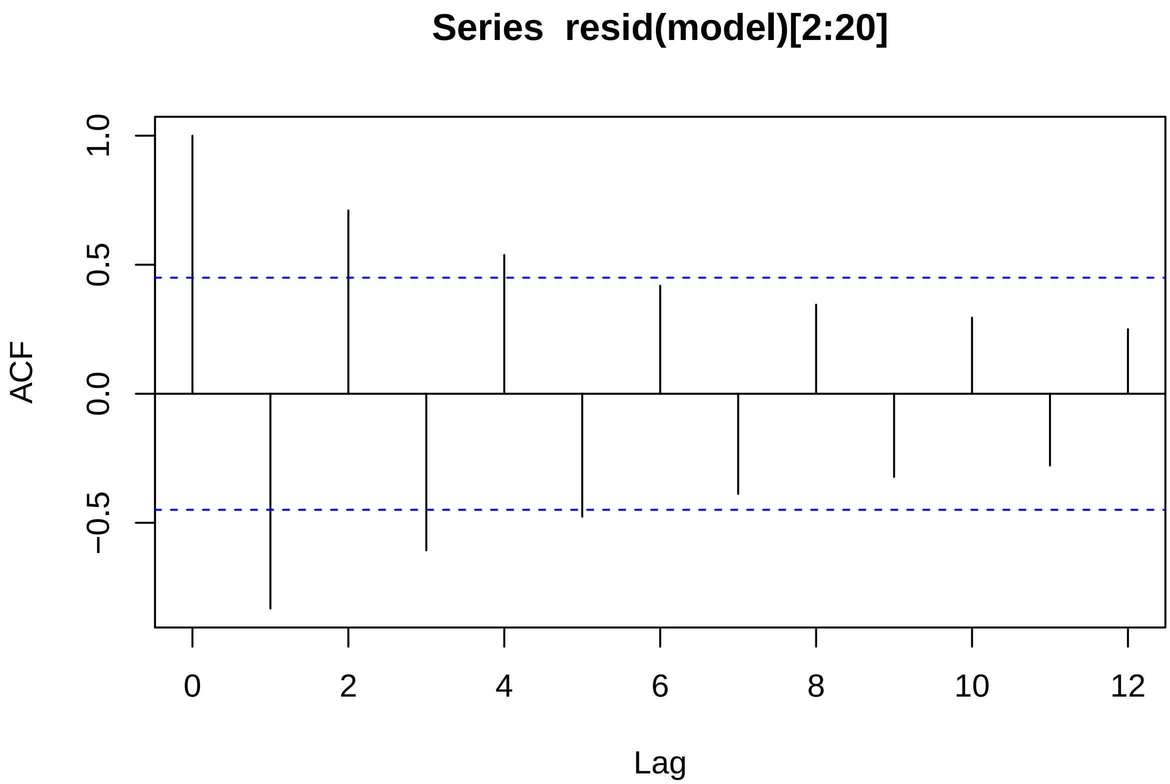

```r
pacf(resid(model)[2:20])
```



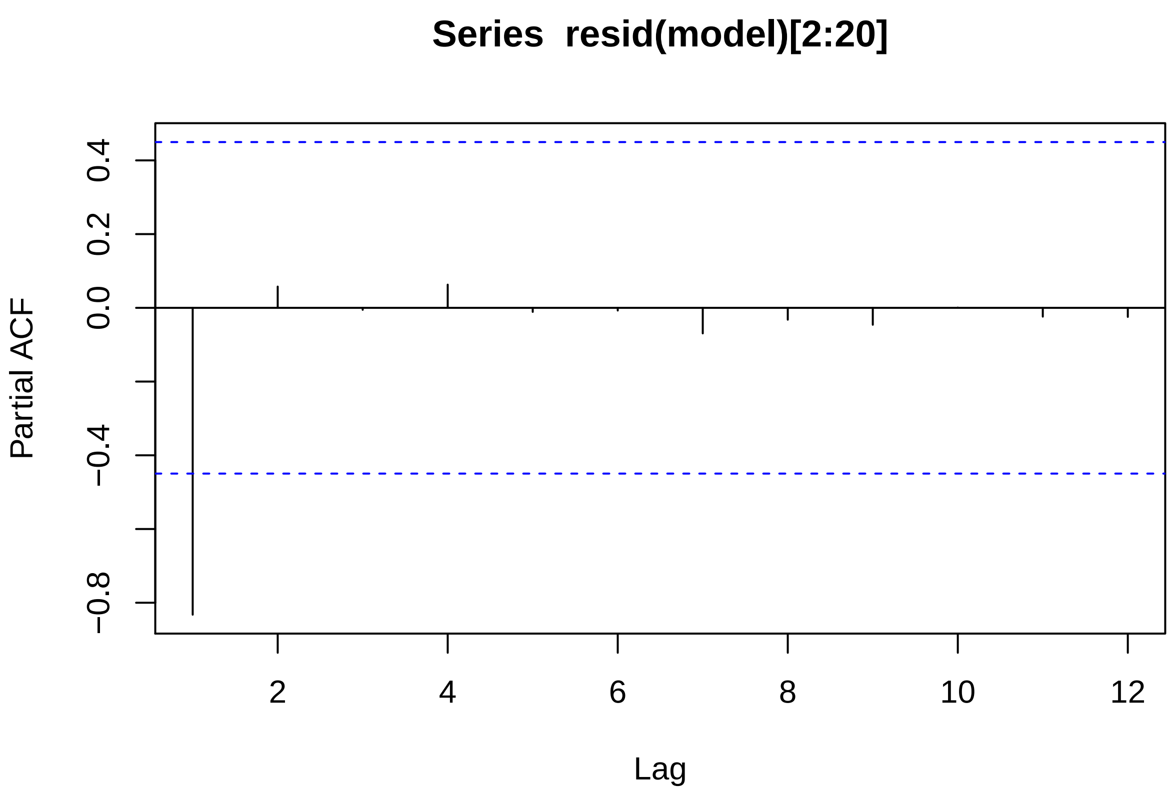
Based on this AR model, after controlling for the t-1 change, almost all of the subestquent residuals are uncorrellated. Therefore, the AR(1) fit works well.

c

```
model <- arima(returns$continuous, order = c(0,0,1))
acf(resid(model)[2:20])
```




```
pacf(resid(model)[2:20])
```



Based on the acf of this model, we see that our errors may be highly correlated if we use and MA(1) model and therefore, we should reconsider.