ba2500_TS_HW6

BARAN AKYOL

```
getSymbols("VALE", from="2006-01-01")
```

```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now 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 more details.
```

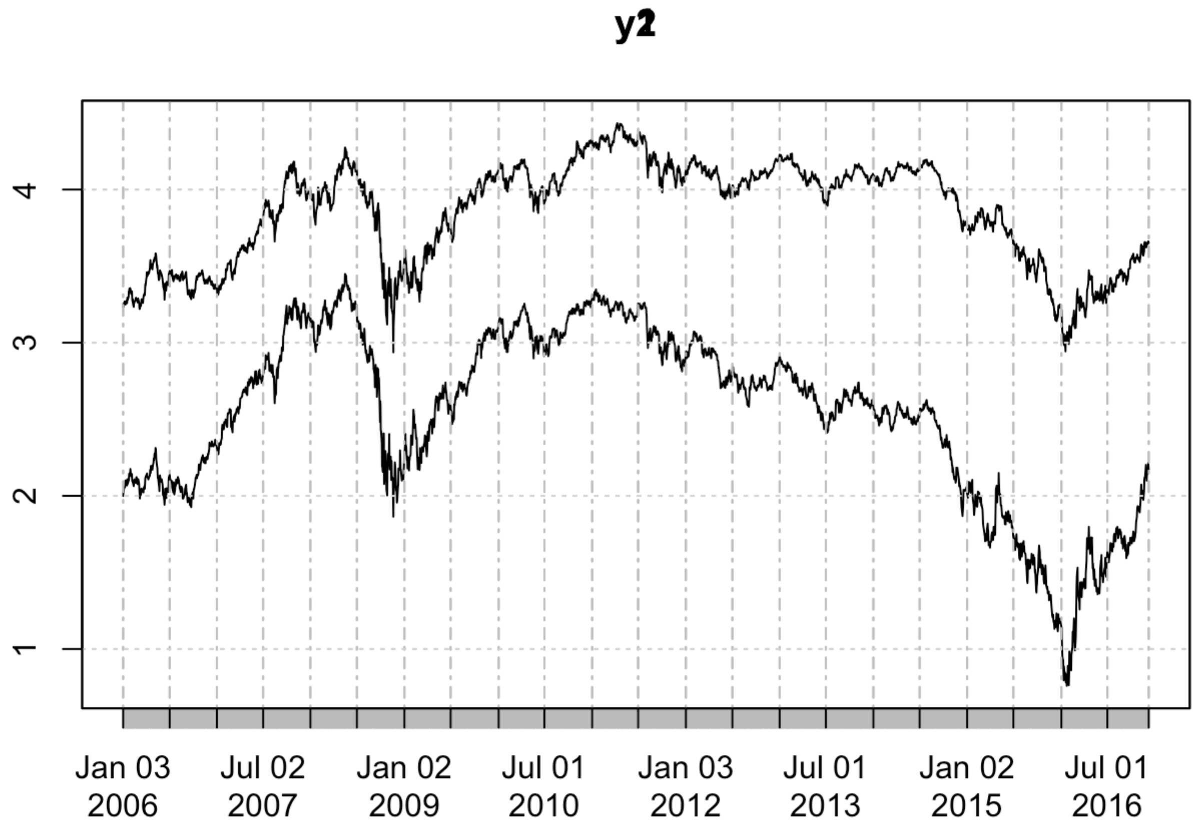
```
## [1] "VALE"
```

```
getSymbols("BHP", from="2006-01-01")
```

```
## [1] "BHP"
```

```
y1 = log(BHP[,6])
y2 = log(VALE[,6])

plot(y1, ylim=range(c(y1,y2)), ylab = "")
#second plot  EDIT: needs to have same ylim
par(new = TRUE)
plot(y2, ylim=range(c(y1,y2)), axes = FALSE, xlab = "", ylab = "")
```


y2
4
3
2
1
Jan 03 2006
Jul 02 2007
Jan 02 2009
Jul 01 2010
Jan 03 2012
Jul 01 2013
Jan 02 2015
Jul 01 2016

```r
data.hw6 = data.frame(log(BHP[,6]), log(VALE[,6])) #p1=log(BHP[,6]) and p2=log(VALE[,6])
n = nrow(data.hw6)
t = 200
coeff = NULL
error = matrix(0, ncol=(n-t), nrow=t)
a = numeric(ncol(error))
h = numeric(length(a))
s = NULL
r_Ts = NULL
sigma = NULL
delta = NULL

for(i in 1:(n-t)){
    temp1 = data.hw6[i:(i+t-1),]
    obj = lm(temp1[,1] ~ temp1[,2], data = temp1)
    coeff = rbind.data.frame(coeff, coef(obj))
    names(coeff) = c("alpha", "beta")
    error[,i] = obj$residuals
    a[i] = coef(lm(error[2:t, i] ~ 0 + error[1:(t-1), i]))
    h[i] = log(0.5)/abs(a[i])
    s_h = ifelse(h[i] >= 1, floor(h[i]), 1)
    error_T = tail(error[,i], 1)
    if(error_T>0){
      temp2 = -0.5*(data.hw6[(i+t-1+s_h),1] - data.hw6[(i+t-1),1]) +
               0.5*(data.hw6[(i+t-1+s_h),2] - data.hw6[(i+t-1),2])
    } else {
      temp2 = 0.5*(data.hw6[(i+t-1+s_h),1] - data.hw6[(i+t-1),1]) -
              0.5*(data.hw6[(i+t-1+s_h),2] - data.hw6[(i+t-1),2])
    }
    s  = c(s, s_h)
    r_Ts = c(r_Ts, temp2)
    temp3 = var(error[,i])
    sigma = c(sigma, temp3)
    temp4 = abs(tail(error[,i], 1))/sqrt(temp3)
    delta = c(delta, temp4)
    rm(temp1, obj, temp2, error_T, s_h, temp3, temp4)
}


data.resume = cbind.data.frame(coeff, a, h, s, r_Ts, sigma, delta)
head(data.resume)
```

```
##      alpha      beta         a          h s          r_Ts       sigma
## 1 1.945640 0.6898287 0.9745318 -0.7112617 1  0.0021221440 0.005161027
## 2 1.960928 0.6827064 0.9746657 -0.7111641 1 -0.0030017683 0.005117595
## 3 1.969618 0.6787762 0.9722651 -0.7129199 1 -0.0004944919 0.005078688
## 4 1.980400 0.6738983 0.9729582 -0.7124121 1 -0.0015535618 0.005013557
## 5 1.986213 0.6714032 0.9734354 -0.7120628 1  0.0279978466 0.004959784
## 6 1.996056 0.6668561 0.9710257 -0.7138299 1 -0.0002435239 0.004913481
##        delta
## 1 0.08011858
## 2 0.04769191
## 3 0.19974860
## 4 0.17699591
## 5 0.26917266
## 6 0.32108152
```

```
tail(data.resume)
```

```
##         alpha      beta         a          h s         r_Ts       sigma
## 2550 2.522120 0.5337199 0.9390921 -0.7381035 1 -0.013906169 0.002433588
## 2551 2.517771 0.5361183 0.9405981 -0.7369218 1 -0.003486021 0.002416522
## 2552 2.515469 0.5373167 0.9410697 -0.7365524 1 -0.013775845 0.002406297
## 2553 2.515072 0.5373245 0.9435148 -0.7346437 1 -0.011972135 0.002405859
## 2554 2.517251 0.5358255 0.9432576 -0.7348440 1  0.011898343 0.002413906
## 2555 2.515155 0.5368734 0.9415400 -0.7361845 1  0.003601753 0.002409673
##          delta
## 2550 0.2084866
## 2551 0.4717257
## 2552 0.3773384
## 2553 0.7693311
## 2554 0.8653116
## 2555 0.6152449
```

```
error = as.data.frame(error)
names(error) = paste("Residuals W", 1:ncol(error), sep="_")
head(error[1:6])
```

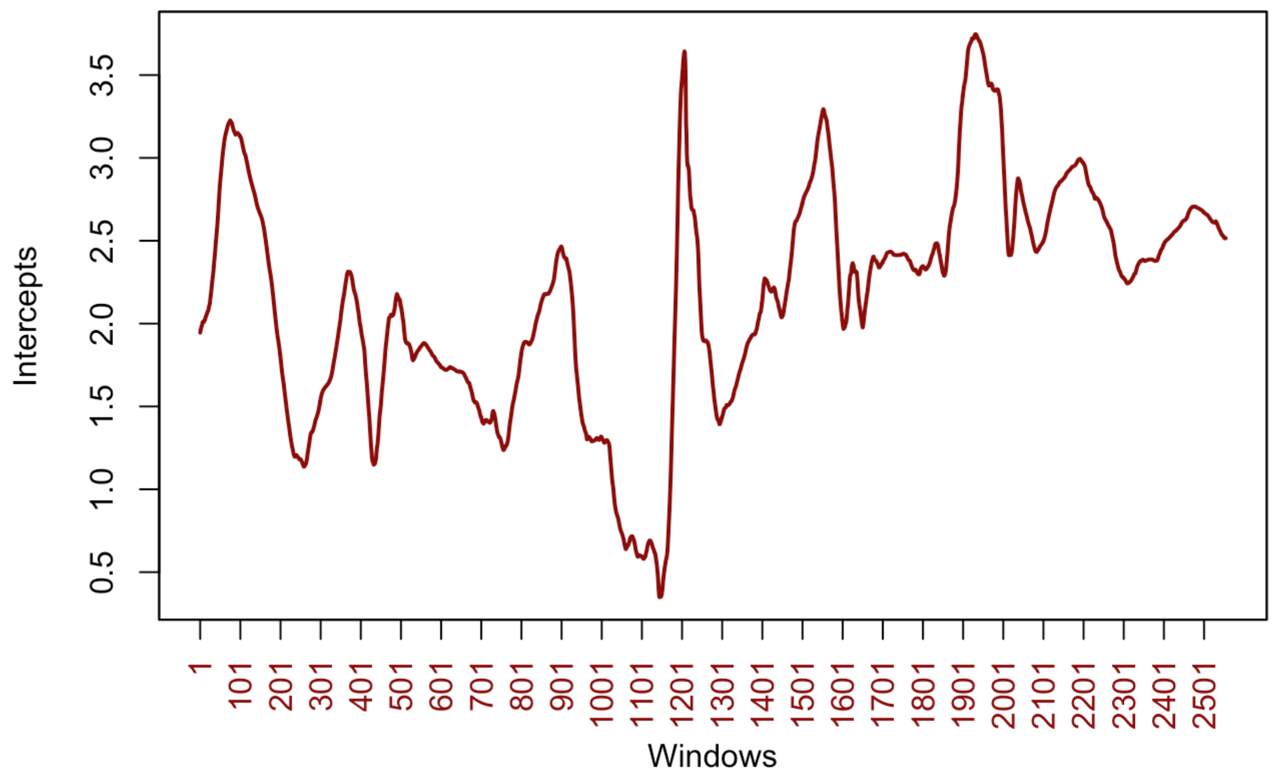
```
##   Residuals W_1 Residuals W_2 Residuals W_3 Residuals W_4 Residuals W_5
## 1   -0.09251079   -0.08879261   -0.11411946   -0.10485434   -0.09836423
## 2   -0.08792787   -0.11341171   -0.10404710   -0.09768665   -0.11130693
## 3   -0.11258882   -0.10339387   -0.09694401   -0.11062388   -0.10906963
## 4   -0.10266979   -0.09634285   -0.10987053   -0.10842467   -0.14734012
## 5   -0.09571310   -0.10926073   -0.10774579   -0.14676807   -0.13680250
## 6   -0.10861535   -0.10719600   -0.14623174   -0.13621731   -0.11138095
##   Residuals W_6
## 1    -0.1118010
## 2    -0.1094943
## 3    -0.1476319
## 4    -0.1371183
## 5    -0.1117305
## 6    -0.1171587
```

```
tail(error[1:6])
```

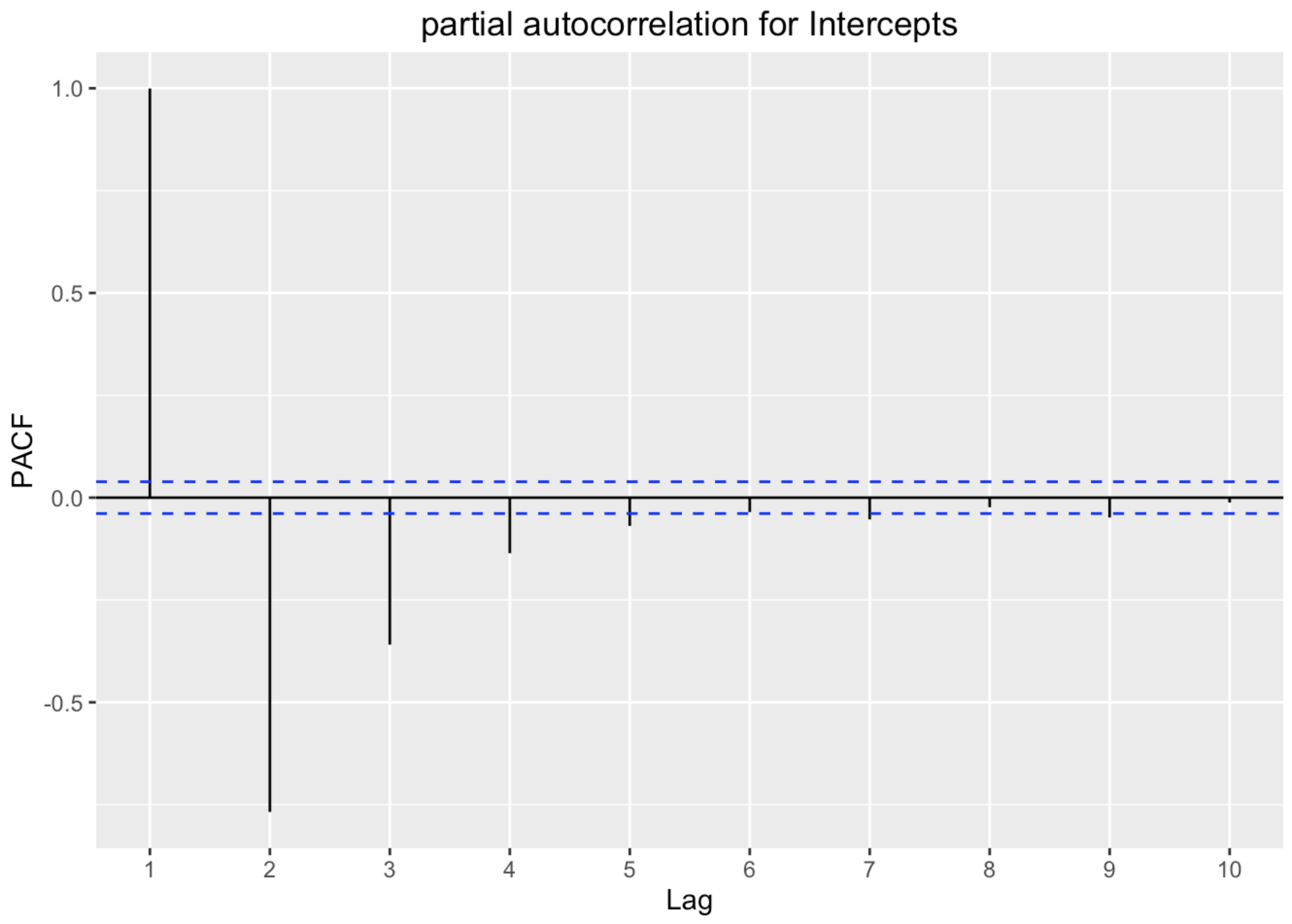
```
##     Residuals W_1 Residuals W_2 Residuals W_3 Residuals W_4 Residuals W_5
## 195  -0.034210684  -0.044608875  -0.027836617  -0.009864911   0.012444030
## 196  -0.044240608  -0.027467798  -0.009396344   0.012926572   0.004170778
## 197  -0.027259117  -0.009016007   0.013287931   0.004703978   0.002119857
## 198  -0.008786453   0.013581885   0.005164371   0.002636403   0.013409291
## 199   0.013654900   0.005538122   0.003064237   0.013885699   0.012038126
## 200   0.005755741   0.003411754   0.014235068   0.012532456   0.018956682
##     Residuals W_6
## 195   0.003949772
## 196   0.001929203
## 197   0.013291782
## 198   0.011887959
## 199   0.018859257
## 200  -0.022506603
```

question 1

```
data.resume = zoo(data.resume)
plot(data.resume$alpha, xaxt="n", xlab="Windows", ylab="Intercepts", lwd=2, col="darkre
d")
x = seq(1, (n-t), by=100)
axis(1, at=x, labels=x, col.axis="darkred", las=2)
```

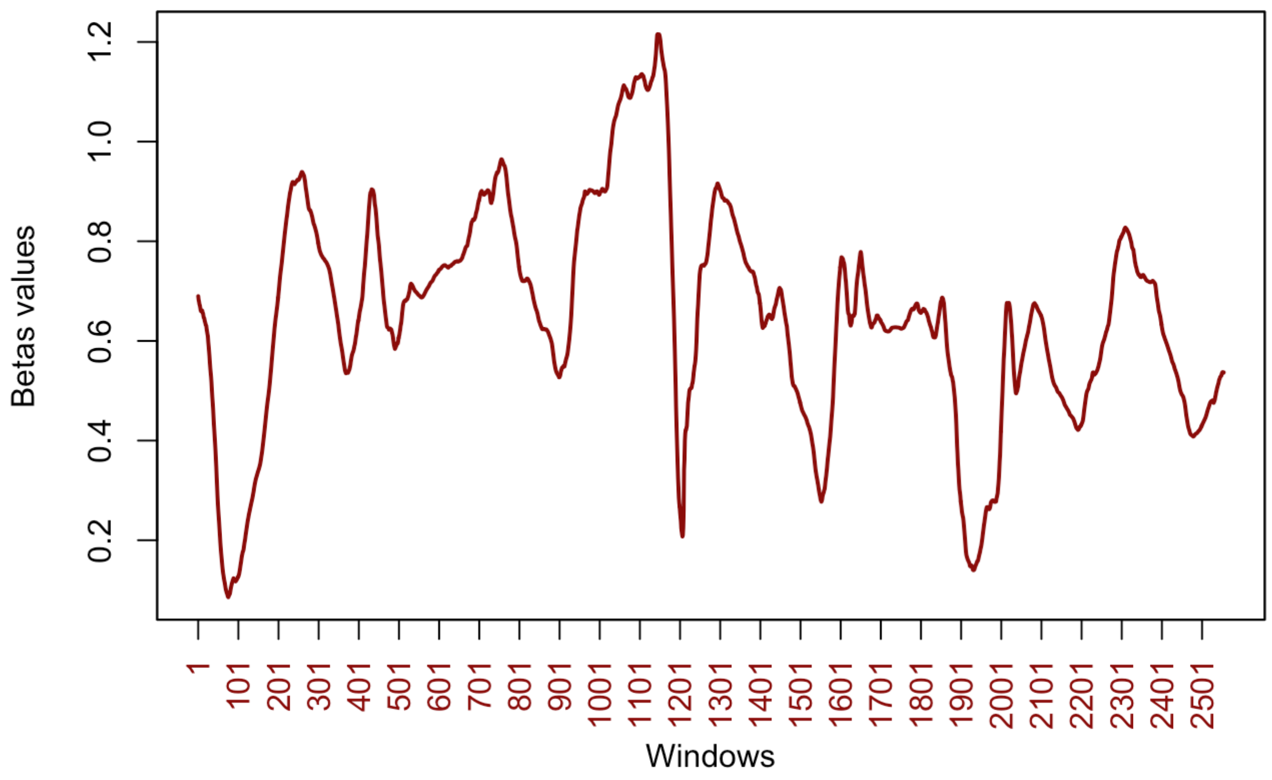


```
#fBasics::pacfPlot(data.resume$alpha, lag.max=10)
forecast::ggPacf(data.resume$alpha, lag.max=10, main="partial autocorrelation for Interc
epts")
```

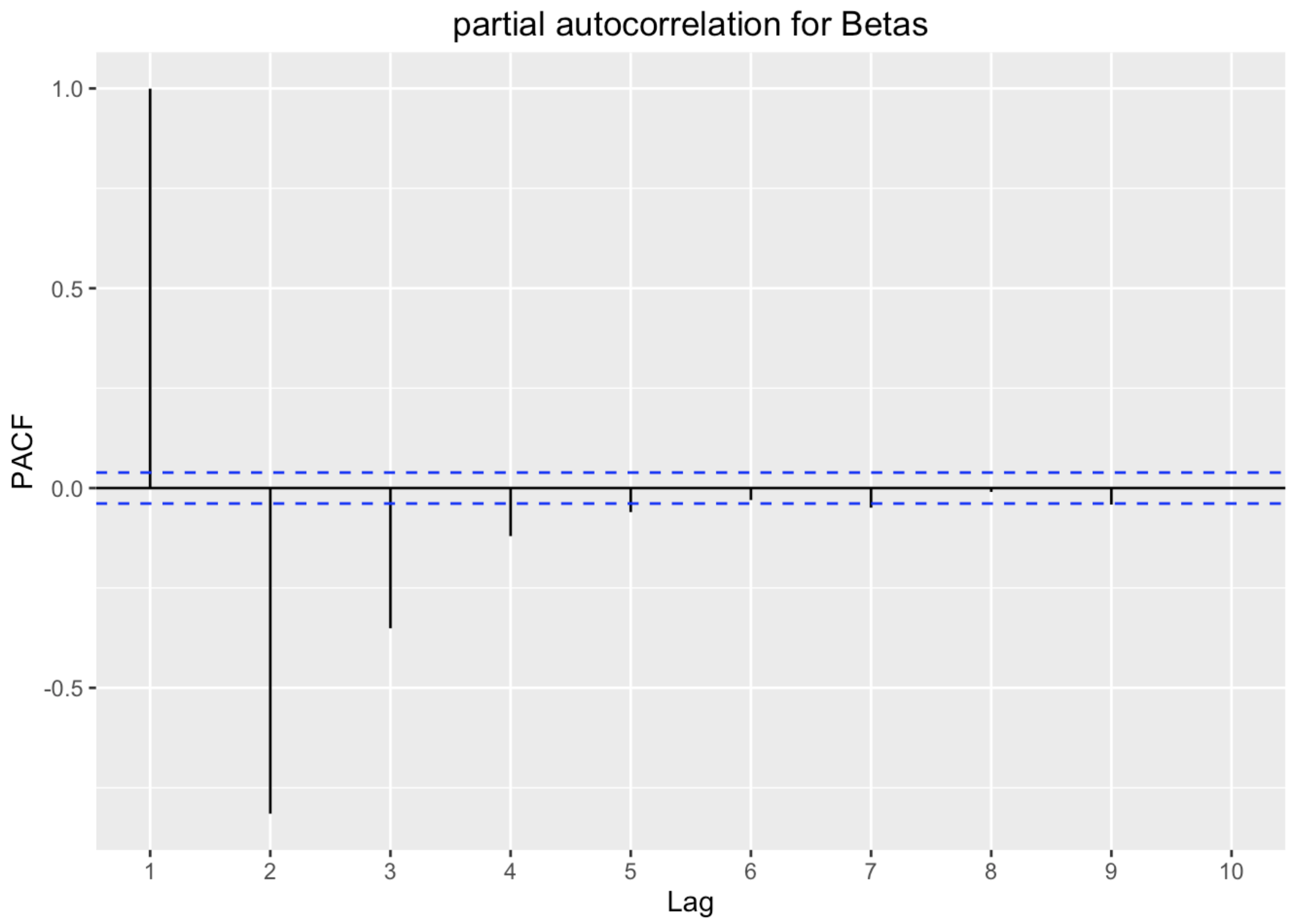


question 2

```
plot(data.resume$beta, xaxt="n", xlab="Windows", ylab="Betas values", lwd=2, col="darkre
d")
x = seq(1, (n-t), by=100)
axis(1, at=x, labels=x, col.axis="darkred", las=2)
```

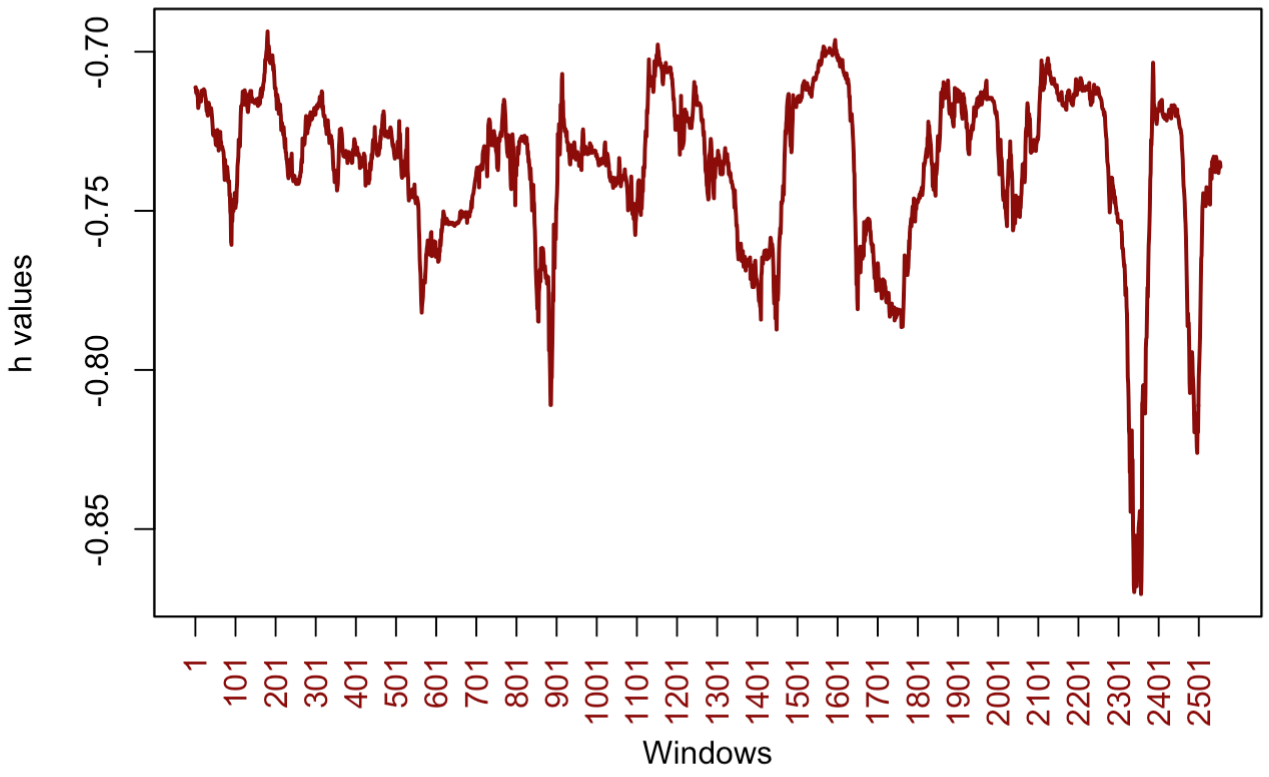


```
#fBasics::pacfPlot(coeff$beta, lag.max=10)
forecast::ggPacf(data.resume$beta, lag.max=10, main="partial autocorrelation for Betas")
```

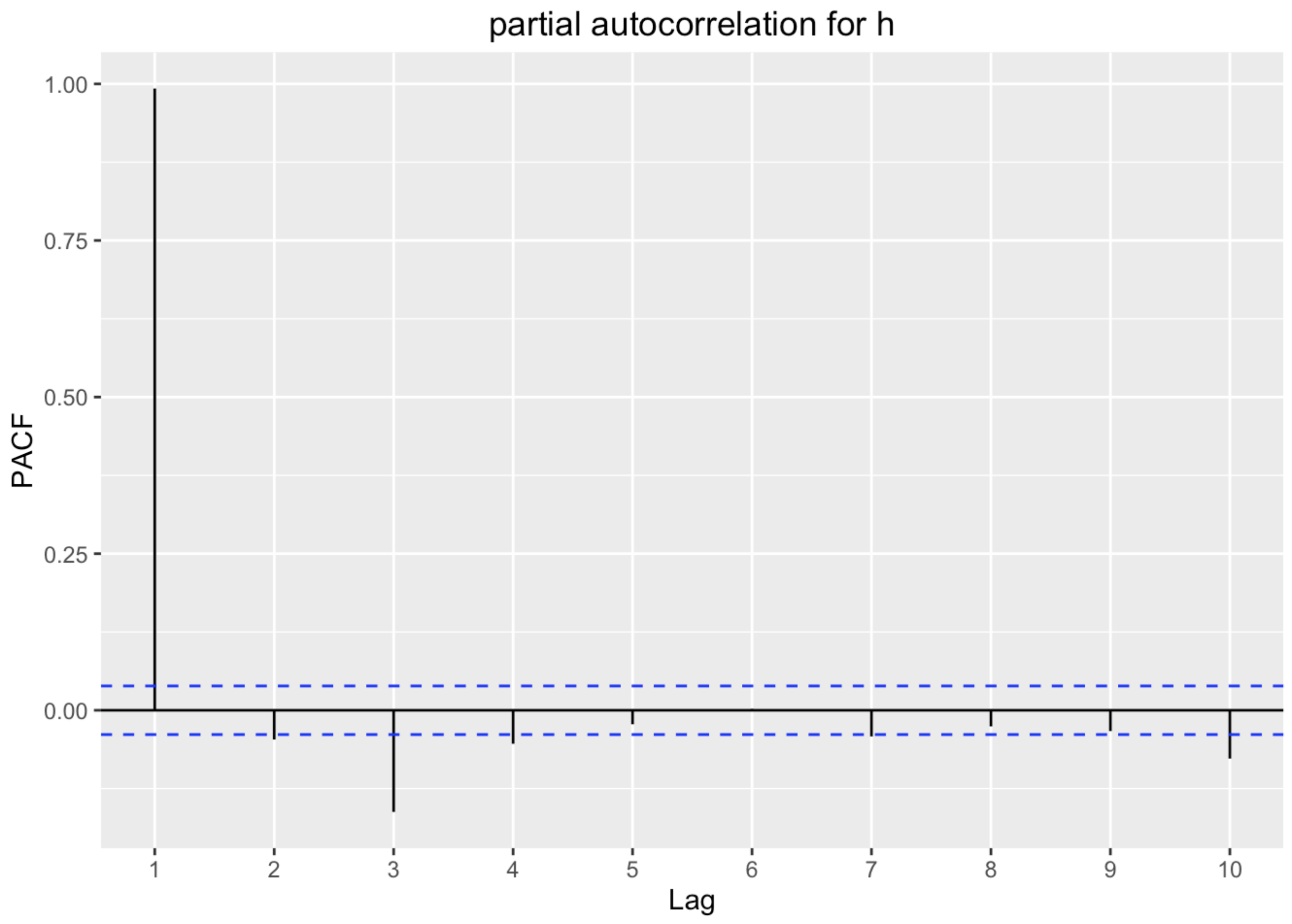


question 3

```
plot(data.resume$h, xaxt="n", xlab="Windows", ylab="h values", lwd=2, col="darkred")
x = seq(1, (n-t), by=100)
axis(1, at=x, labels=x, col.axis="darkred", las=2)
```



```
#fBasics::pacfPlot(data.resume$h, lag.max=10)
forecast::ggPacf(data.resume$h, lag.max=10, main="partial autocorrelation for h")
```

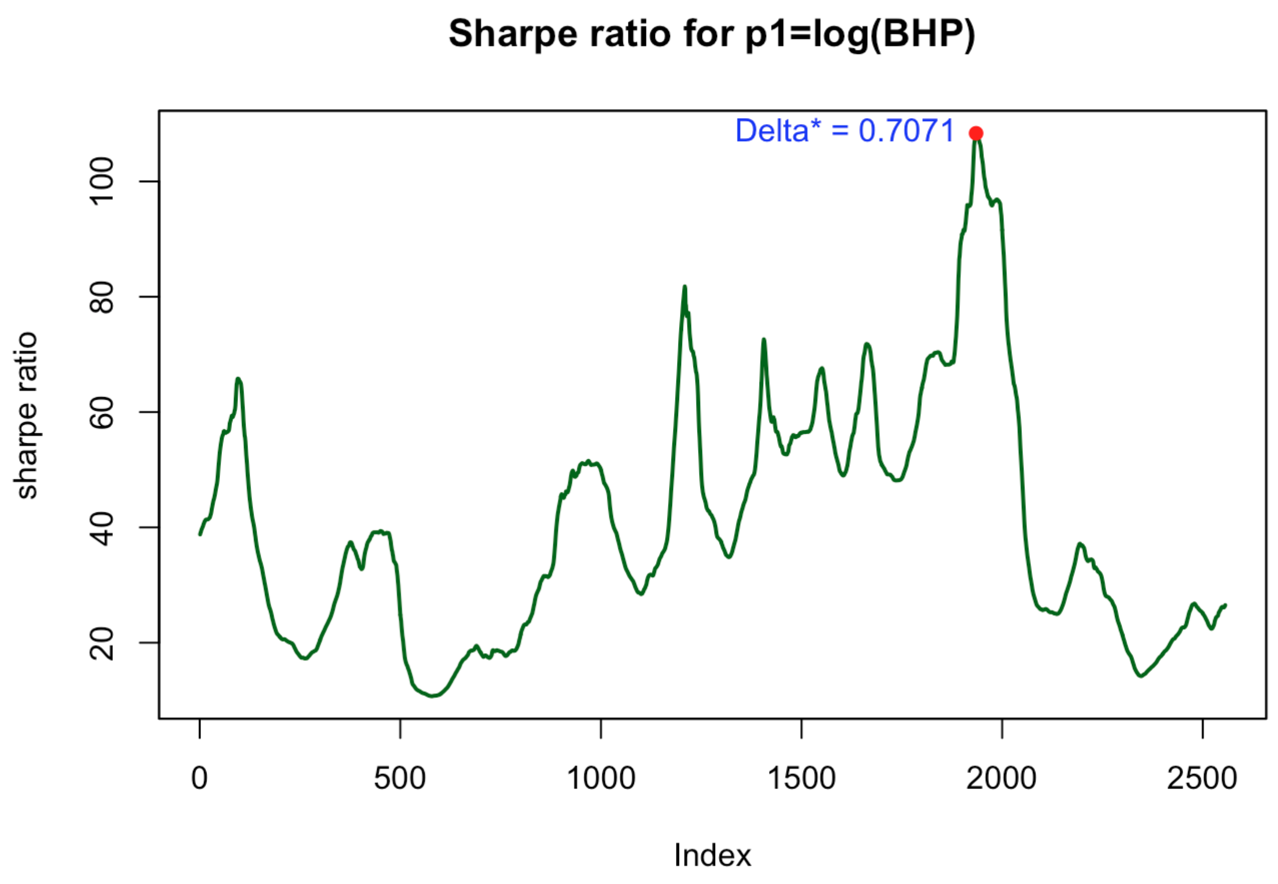


question 4

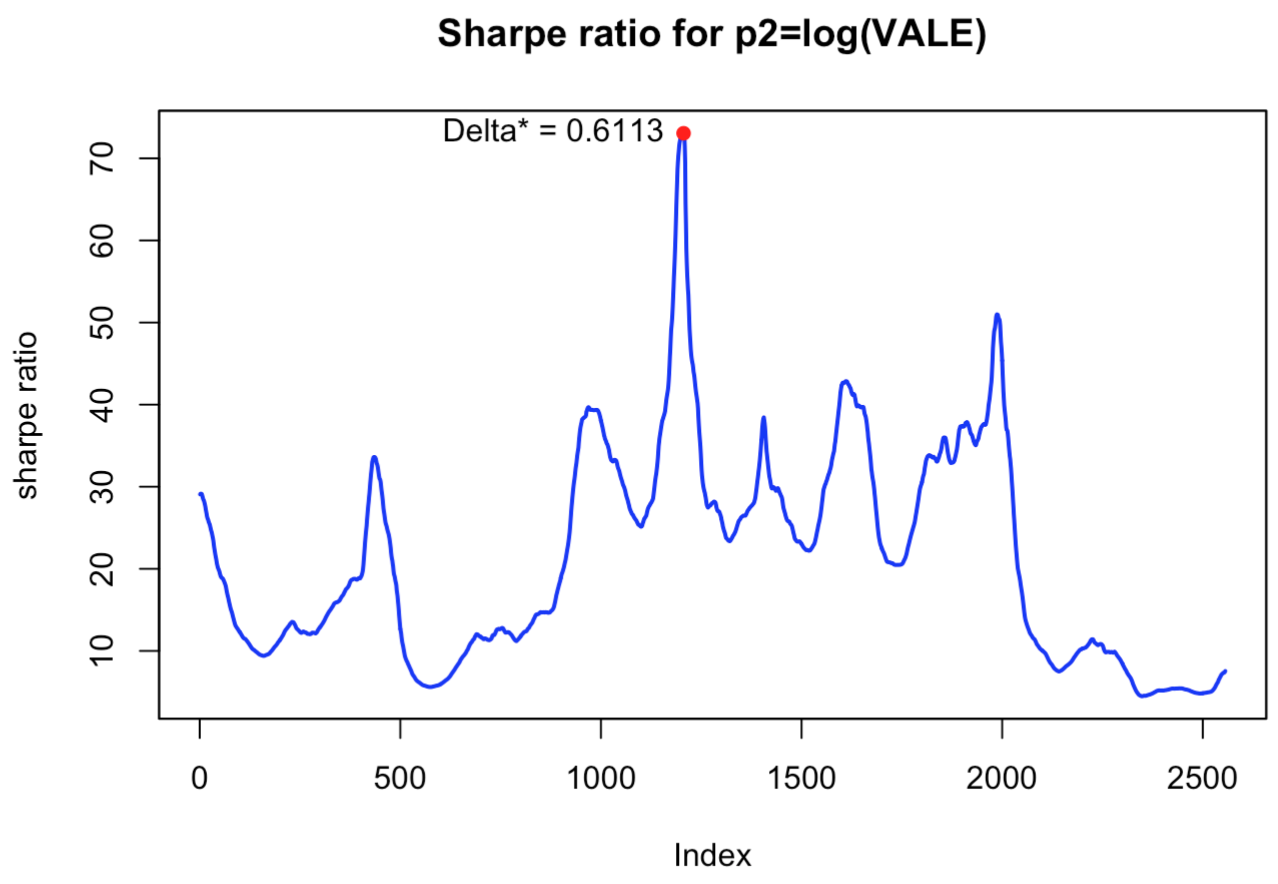
```
myrollmean = apply(data.hw6, 2, function(x){rollapply(x,t, mean)})
myrollsd = apply(data.hw6, 2, function(x){rollapply(x, t, sd)})
sharpe.p1 = myrollmean[,1]/myrollsd[,1]
sharpe.p2 = myrollmean[,2]/myrollsd[,2]



plot(zoo(sharpe.p1), ylab='sharpe ratio', lwd=2, col="darkgreen",
     main="Sharpe ratio for p1=log(BHP)")
maxid = which.max(sharpe.p1)
points(maxid, sharpe.p1[maxid], col='red', pch=20, lwd=2.5)
text(maxid, sharpe.p1[maxid], labels = paste("Delta* = ", round(delta[maxid+1], 4), sep=""
,
     pos=2, col='blue')
```



```
plot(zoo(sharpe.p2), ylab='sharpe ratio', lwd=2, col="blue",
     main="Sharpe ratio for p2=log(VALE)")
maxid = which.max(sharpe.p2)
points(maxid, sharpe.p2[maxid], col='red', pch=20, lwd=2.5)
text(maxid, sharpe.p2[maxid], labels = paste("Delta* = ", round(delta[maxid+1], 4), sep=""
,
     pos=2, col='black')
```


Sharpe ratio for p2=log(VALE)
Delta* = 0.6113
sharpe ratio
Index
0
500
1000
1500
2000
2500
10
20
30
40
50
60
70