Empirical Assignment 6

Nitish Ramkumar, Ian Laker, Prasanth Kumar

Portfolios Sort

1a

The winner stocks based on the sorting strategy are as follows

```
##            [,1]      [,2]
## 2012-01-01 "Stock B" "Stock C"
## 2013-01-01 "Stock B" "Stock D"
## 2014-01-01 "Stock A" "Stock D"
## 2015-01-01 "Stock C" "Stock D"
## 2016-01-01 "Stock A" "Stock C"
```

The time series returns for the winners and losers are as below

```
##            Winner  Loser
## 2013-01-01   0.09  0.010
## 2014-01-01   0.15 -0.015
## 2015-01-01   0.02  0.010
## 2016-01-01  -0.05  0.020
```

The mean returns for the winners and losers are as below

```
##  Winner   Loser
## 0.05250 0.00625
```

1b

Yes, we can see the expected momentum result, as the winners gain substantially more than the losers.

The average market returns is

```
## [1] 0.038
```

1c

The time series returns after creating the long short portfolio is

```
##            LongShort
## 2013-01-01     0.080
## 2014-01-01     0.165
## 2015-01-01     0.010
## 2016-01-01    -0.070
```

The average long-short returns is **4.625%**

When we run a regression between the long short returns and the market returns, we get a correlation (market beta) of **0.003169**. Given this small value of beta, we can say that CAPM doesn't hold good, as explained returns are not correlated to the market.

2

The time series returns from the quartile sorts is

```
##            Sort1 Sort2 Sort3 Sort4
## 2013-01-01 -0.03  0.05  0.03  0.15
## 2014-01-01  0.07 -0.10  0.25  0.05
## 2015-01-01  0.03 -0.01 -0.06  0.10
## 2016-01-01  0.33 -0.29  0.08 -0.18
```

The mean of the quartile sorts is

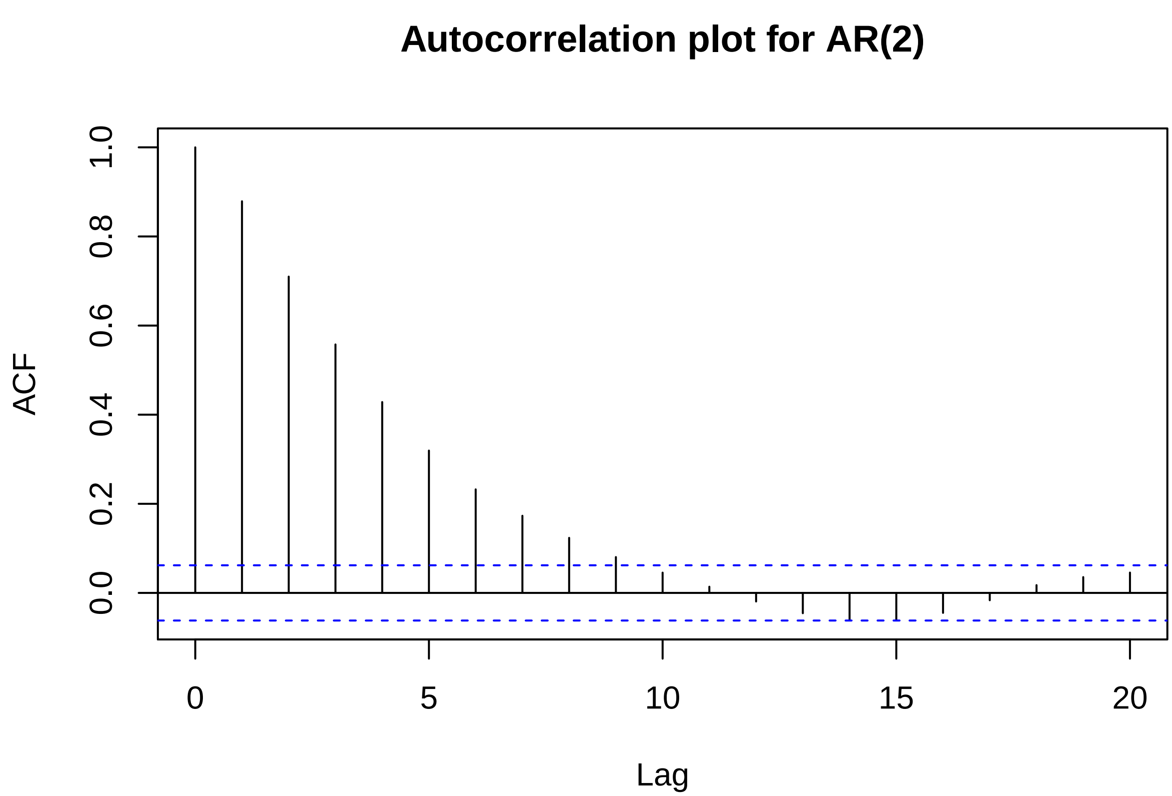
```
##   Sort1   Sort2   Sort3   Sort4
##  0.1000 -0.0875  0.0750  0.0300
```

The expected momentum result is not achieved as the maximum quartile in this momentum strategy has return less than the minimum quartile.

AR(p) Process

3a

The plot for the AR(2) plot, given ϕ_1 and ϕ_2 is



3b

As modulus of both characteristic roots of $1 - \phi_1 x - \phi_2 x^2 = 0$ (0.32087,0.77913) are less than 1, this is a **stationary process**

3c

Let's assume $r_{t+6} - \mu = x_{t+6}$. As μ is mean for all elements of the time series, this can be extended to x_{t+5}, x_{t+4} as well.

To get the dynamic multiplier, we can start with $x_{t+6} = \phi_1 x_{t+5} + \phi_2 x_{t+4} + \epsilon_{t+6}$, and then substitute x_{t+5} and x_{t+4} with further lag values.

$= \epsilon_{t+6} + \phi_1 \epsilon_{t+5} + ({\phi_1}^2 + \phi_2) x_{t+4} + \phi_1 \phi_2 x_{t+3}$

$= \epsilon_{t+6} + \phi_1 \epsilon_{t+5} + ({\phi_1}^2 + \phi_2) \epsilon_{t+4} + ({\phi_1}^3 + 2\phi_1\phi_2) x_{t+3} + ({\phi_1}^2 \phi_2 + {\phi_2}^2) x_{t+2}$

$= \epsilon_{t+6} + \phi_1 \epsilon_{t+5} + ({\phi_1}^2 + \phi_2) \epsilon_{t+4} + ({\phi_1}^3 + 2\phi_1\phi_2) \epsilon_{t+3} + ({\phi_1}^4 + 3{\phi_1}^2 \phi_2 + {\phi_2}^2) x_{t+2} + ({\phi_1}^3 \phi_2 + 2\phi_1 {\phi_2}^2) x_{t+1}$

$= ....({\phi_1}^4 + 3{\phi_1}^2\phi_2 + {\phi_2}^2)\epsilon_{t+2} + ({\phi_1}^5 + 4{\phi_1}^3\phi_2 + 3\phi_1{\phi_2}^2)x_{t+1} + ({\phi_1}^4\phi_2 + 3{\phi_1}^2{\phi_2}^2 + {\phi_2}^3)x_t$

$= .....({\phi_1}^5 + 4{\phi_1}^3\phi_2 + 3\phi_1{\phi_2}^2)\epsilon_{t+1} + ({\phi_1}^6 + 5{\phi_1}^4\phi_2 + 6{\phi_1}^2{\phi_2}^2 + {\phi_2}^3)x_t + ({\phi_1}^5\phi_2 + 4{\phi_1}^3{\phi_2}^3 + 3\phi_1{\phi_2}^3)x_{t-1}$

$= .....({\phi_1}^6 + 5{\phi_1}^4\phi_2 + 6{\phi_1}^2{\phi_2}^2 + {\phi_2}^3)\epsilon_t + ...$

So, Dynamic multiplier for 6 period ago shock $= \frac{\partial x_{t+6}}{\partial \epsilon_t} =$

$= \boldsymbol{{\phi_1}^6 + 5{\phi_1}^4\phi_2 + 6{\phi_1}^2{\phi_2}^2 + {\phi_2}^3}$

We know, ϕ_1 = 1.1, ϕ_2 = -0.25, So Dynamic Multiplier is **0.379561**

3d

The Dynamic multiplier is **6.778241**. This is very high compared to the previous value, which explains that the shock value is pretty significant and this might be a point against the mean reversion.

One of the characteristic roots has an absolute value greater than 1 (-0.5512492,1.451249). This means that this process is **not stationary**. From an intuition point of view, even though there was a small reversion in the AR(1) part (phi_1 = 0.9), the significantly positive AR(2) part (phi_2 = 0.8) doesn't allow the process to mean revert.

R Code

```
suppressMessages(library(xts))

stockreturns <- matrix(c(2012,-0.12,0.29,0.15,-0.03,2013,-0.03,0.15,0.03,0.05,2014,0.07,0.05,-0.1,0.25,
stockreturns.xts <- as.xts(stockreturns[,-1],as.Date(paste0(stockreturns[,1],"-01-01")))
colnames(stockreturns.xts) <- c("Stock A","Stock B","Stock C","Stock D")

findMedian <- function(vec){
  vec.sorted <- sort(vec)
  if(length(vec.sorted) %% 2 == 0){
    (vec.sorted[length(vec.sorted)/2] + vec.sorted[(length(vec.sorted)/2)+1])/2
  }
  else{
```

```
    vec.sorted[length(vec.sorted)%/%2 + 1]
  }
}

##1a
#Store positions of winning stocks and convert it into xts
Winner.positions <- xts(t(apply(stockreturns.xts,1,function(x){which(x > findMedian(x))})),order.by=inde

#PRINT WINNERS
Winner.names <- t(sapply(1:nrow(stockreturns.xts),
                         function(count){
                         #Print column name of calculated indices
                         colnames(stockreturns.xts[count,Winner.positions[count]])
                 }))
Winner.names <- xts(Winner.names,index(stockreturns.xts))
Winner.names

#CALCULATE RETURNS
WinnerLoserReturnsFunc <- function(row){
                           #get previous winners using lag
                           previousWinners <- stats::lag(Winner.positions)[-1][index(row),]
                           #calculate mean of winners(containing previousWinners) and losers
                           c(mean(row[,previousWinners]),mean(row[,-previousWinners]))
                         }

WinnerLoserReturns <- t(sapply(1:nrow(stockreturns.xts[-1]),function(count)
                                    {
                                      WinnerLoserReturnsFunc(stockreturns.xts[-1][count,])
                                    }))

colnames(WinnerLoserReturns) <- c("Winner","Loser")
WinnerLoserReturns <- xts(WinnerLoserReturns,index(stockreturns.xts[-1]))
WinnerLoserReturns

#MEAN OF RETURNS
apply(WinnerLoserReturns,2,mean)

##1b

Market.weights <- c(0.3,0.2,0.353,0.147)
Market.returns <- stockreturns.xts%*%Market.weights

#MEAN MARKET
mean(Market.returns)

LongShort.Returns <- WinnerLoserReturns[,'Winner'] - WinnerLoserReturns[,'Loser']
colnames(LongShort.Returns) <- "LongShort"
LongShort.Returns

#MEAN
LongShort.mean <- mean(LongShort.Returns)

#REGRESSION
```

```
reg <- summary(lm(LongShort.Returns ~ Market.returns[-1]))
betaEst <- reg$coefficients[2,1]

##2

Positions<- t(apply(stockreturns.xts,1,
                        function(timeVal){
                            sortedVal <- sort(coredata(timeVal))
                            sapply(sortedVal,function(elem){which(timeVal==elem)})
                          }))
Positions.xts <- xts(Positions,index(stockreturns.xts))
Positions.xts.lag <- stats::lag(Positions.xts)[-1]

QuartileSorts <- t(sapply(1:nrow(Positions.xts.lag),function(x){
  val <- stockreturns.xts[-1][x,Positions.xts.lag[x,]]
  colnames(val) <- NULL
  val
  }))

colnames(QuartileSorts) <- c("Sort1","Sort2","Sort3","Sort4")
QuartileSorts <- xts(QuartileSorts,index(Positions.xts.lag))
QuartileSorts

#MEAN
apply(QuartileSorts,2,mean)

#AR(p) Process
##3a
phi1 <- 1.1
phi2 <- -0.25

set.seed('1234')
ar.sim1 <-arima.sim(model=list(ar=c(phi1,phi2)),n=1000)
acf(ar.sim1,lag.max = 20,main="Autocorrelation plot for AR(2)")

##3b
x1 = (phi1 + sqrt(phi1^2 + 4*phi2))/(-2 * phi2)
x2 = (phi1 - sqrt(phi1^2 + 4*phi2))/(-2 * phi2)

root1 <- 1/x1
root2 <- 1/x2

##3c
dynamicMult_1 <- phi1^6 + 5*phi1^4*phi2 + 6*phi1^2*phi2^2 + phi2^3

##3d
phi1 = 0.9
phi2 = 0.8

dynamicMult_2 <- phi1^6 + 5*phi1^4*phi2 + 6*phi1^2*phi2^2 + phi2^3

x1_2 = (phi1 + sqrt(phi1^2 + 4*phi2))/(-2 * phi2)
x2_2 = (phi1 - sqrt(phi1^2 + 4*phi2))/(-2 * phi2)
```

```r
root1_2 <- 1/x1_2
root2_2 <- 1/x2_2
```