Groupwork Assignment Submission 3 M7

Group: (19/06) MScFE 610 Econometrics (C19-S2) - Group 5L

Submitted By:

1. Jinesh Joseph (jineshjoseph91@gmail.com)
2. Rose Kagumba (wangui.kagumba@gmail.com)
3. Abhijeet Kant Sinha (aks.abhikant@gmail.com)
4. Siddhartha Sarma (siddhartha.krishnatre@gmail.com)

INTRODUCTION

The MACD, short for moving average convergence/divergence, indicator (or "oscillator") is a collection of three time series calculated from historical price data, most often the closing price. The indicator depends on three-time parameters, namely the time constants of the three EMAs. It is a lagging indicator. The notation "MACD (a, b, c)" usually denotes the indicator where the MACD series is the difference of EMAs with characteristic times a and b, and the average series is an EMA of the MACD series with characteristic time c. These parameters are usually measured in days.

The most commonly used values are 12, 26, and 9 days, that is, MACD (12,26,9). As true with most of the technical indicators, MACD also finds its period settings from the old days when technical analysis used to be mainly based on the daily charts. The reason was the lack of the modern trading platforms which show the changing prices every moment. As the working week used to be 6-days, the period settings of (12, 26, 9) represent 2 weeks, 1 month and one and a half week. Now when the trading weeks have only 5 days, possibilities of changing the period settings cannot be overruled. However, it is always better to stick to the period settings which are used by the majority of traders as the buying and selling decisions based on the standard settings further push the prices in that direction.

Mathematically, MACD is given by:

MACD Line: (**12-day EMA - 26-day EMA**)
Signal Line: **9-day EMA of *MACD* Line**
MACD Histogram: ***MACD* Line – Signal**

In this project, we have used MACD on Alphabet Inc's Class C shares listed on NASDAQ with ticker symbol: **GOOG**.

The section below is systematic and illustrates the process taken in order to execute the MACD method.

```
installedPackages = installed.packages()
if("quantmod" %in% installedPackages == FALSE){
  install.packages("quantmod")
}

library(TTR)
library(quantmod)
```

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

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

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

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

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

```
getSymbols("GOOG",src="yahoo") # from yahoo finance
```

```
## '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.
```

```
## [1] "GOOG"
```

```
GOOGDf = data.frame(GOOG)


GOOGDf$date = index(GOOG)


# To date conversion
as.Date(GOOGDf$date[1], "%Y-%m-%d")+100
```

```
## [1] "2007-04-13"
```

```
GOOGDf$date = as.Date(GOOGDf$date, "%Y-%m-%d")

GOOGDf$date[1] < GOOGDf$date[2]
```

```
## [1] TRUE
```

```r
# For our convenience, we create data object and save GOOGDf object into data object
data = GOOGDf
## Lets start with Google Open prices
# We subset only open prices and dates
data = subset(data, select = c("date", "GOOG.Open"))
# Consider first one year data for initial analysis
#subData = data
# In the initial stage, to get better understanding of the data,
# lets consider only first one year data
subData = subset(data, data$date < data$date[1] + 365)
head(subData)
```

```
##                 date GOOG.Open
## 2007-01-03 2007-01-03  232.1299
## 2007-01-04 2007-01-04  233.6243
## 2007-01-05 2007-01-05  240.3491
## 2007-01-08 2007-01-08  242.9344
## 2007-01-09 2007-01-09  241.8186
## 2007-01-10 2007-01-10  241.3105
```

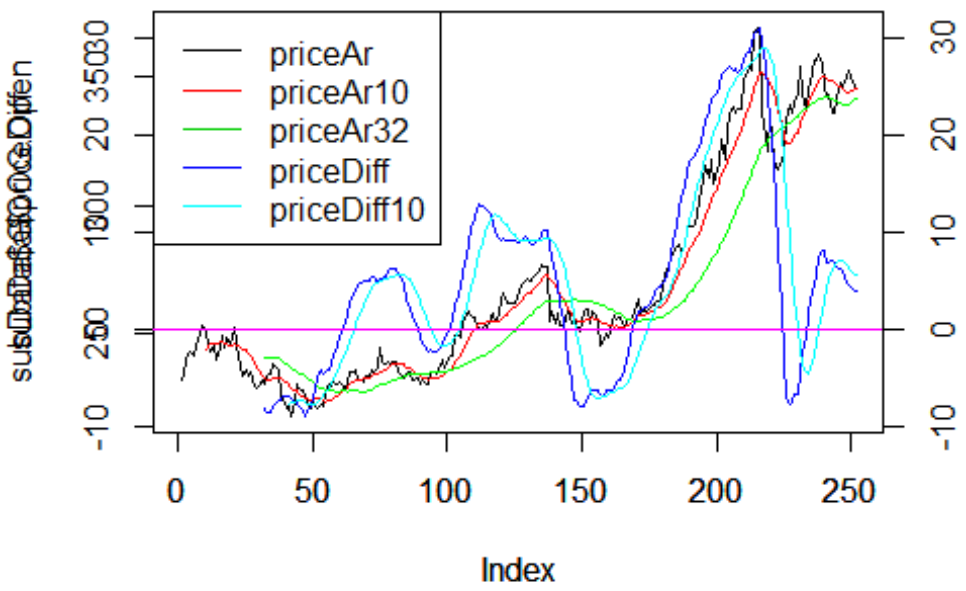
```r
dim(data)
```

```
## [1] 3159    2
```

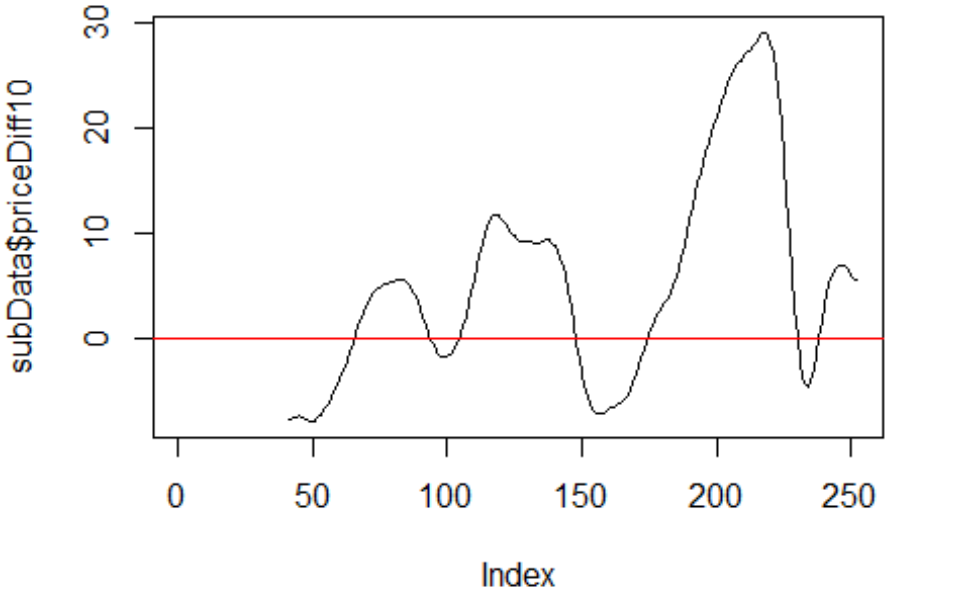
```r
dim(subData)
```

```
## [1] 252   2
```

```r
# Plot Google adjusted prices to check for any simple tradable pattern
plot(subData$GOOG.Open, type="l")

# Calculate two simple moving averages of prices with 10 days and 32 days as
# look back windows.
subData$priceAr10 = SMA(subData$GOOG.Open, 10)
subData$priceAr32 = SMA(subData$GOOG.Open, 32)


# Lets plot original prices along with calculated SMAs prices
plot(subData$GOOG.Open, type = "l", col="black")
lines(subData$priceAr10, col="red")
lines(subData$priceAr32, col="green")


# Calculate simple moving averages price differences
subData$priceDiff = subData$priceAr10 - subData$priceAr32


# Calculate simple moving average of priceDiff with 10 days look back period
subData$priceDiff10 = SMA(subData$priceDiff, 10)
par(new=T)
plot(subData$priceDiff, type="l", col=4)
lines(subData$priceDiff10, col=5)
abline(h=0, col=6)
axis(side = 4)
legend("topleft",
      legend=c("priceAr", "priceAr10", "priceAr32", "priceDiff", "priceDiff10"),
      lty=1, col=c(1,2,3,4,5))
```




```
plot(subData$priceDiff10, type="l")
abline(h=0, col=2)
```



```
## Trading signals
# Idea:
# When SMA10 price point is bigger than SMA32 price point, latest prices are
# are bigger than old prices
# Looks like we can take a buy trade in this case
# Whenever SMA10 price point is smaller than SMA32 price point, latest prices are smaller
# than old prices
# Looks like we can take sell trade in this case
## The following line is very important. This is where we generate trading signal


subData$tradeSignal = ifelse(subData$priceAr10 > subData$priceAr32, 1, -1)


## MACD trading signal
# If priceDiff data point(i.e., subData$priceAr10 - subData$priceAr32 ) is smaller than
# average of last ten priceDiff data points, i.e., priceDiff10 data point, take a sell
# trade
# If priceDiff data point(i.e., subData$priceAr10 - subData$priceAr32 ) is bigger than
# average of last ten priceDiff data points, i.e., priceDiff10 data point, take a bigger
# trade
# New trading signal 2

subData$tradeSignal = ifelse(subData$priceDiff > subData$priceDiff10, 1, -1)


#New trading signal 3

subData$tradeSignal = ifelse(subData$priceDiff > 1.05*subData$priceDiff10, 1,
                      ifelse(subData$priceDiff < 0.95*subData$priceDiff10, -1, 0 ))

# Lets define our trading position and our entry price


subData$position = rep(0, nrow(subData))
subData$price = rep(0, nrow(subData))
subData$trdNo = rep(0, nrow(subData))
presentPos = 0 # To start with
entryPrice = 0  # To start with
exitPrice = 0 # To start with
tradeNo = 0  # To start with
```

```
# We assume that our inventory limit is one stock.
# Due to this inventory limit, even we have sequence of buys, we will buy only at the
# first buy signal. We ignore rest of the buy signals till we get sell signal
# Similarly for sell signal
# We also assume that we can short sell, i.e., we can borrow stocks and sell in the
# market if our trading signal is sell
# With the above assumptions, lets check whether we make any money with our
# trading decisions


for(row_i in 1:nrow(subData)){
  signal = subData$tradeSignal[row_i]
  if(!is.na(signal)){
    if(signal == 1 & presentPos == 0){
      entryPrice = subData$GOOG.Open[row_i]
      subData$position[row_i] = 1
      subData$price[row_i] = entryPrice
      presentPos = 1
      tradeNo = tradeNo + 1
      subData$trdNo[row_i] = tradeNo

    }
    if(signal == 1 & presentPos == -1){
      exitPrice = subData$GOOG.Open[row_i]
      subData$position[row_i] = 1
      subData$price[row_i] = exitPrice
      presentPos = 0
      entryPrice = 0
      exitPrice = 0
    }
    if(signal == -1 & presentPos == 0){
      entryPrice = subData$GOOG.Open[row_i]
      subData$position[row_i] = -1
      subData$price[row_i] = entryPrice
      presentPos = -1
      tradeNo = tradeNo + 1
      subData$trdNo[row_i] = tradeNo
    }
    if(signal == -1 & presentPos == 1){
      exitPrice = subData$GOOG.Open[row_i]
      subData$position[row_i] = -1
      subData$price[row_i] = exitPrice
      presentPos = 0
      entryPrice = 0
      exitPrice = 0
    }
  }
  if(row_i == nrow(subData) & sum(subData$position) == 1){
    subData$position[row_i] = -1
    subData$price[row_i] = subData$GOOG.Open[row_i]
  }
  if(row_i == nrow(subData) & sum(subData$position) == -1){
    subData$position[row_i] = 1
    subData$price[row_i] = subData$GOOG.Open[row_i]
  }
```

```
  if(row_i == nrow(subData) & abs(sum(subData$position)) == 1){
    subData$position[row_i] = 0
  }
}# PnL calculation
sum(subData$position)-sum(subData$position*subData$price)

## [1] 53.52428

## Incorporating thresholds in MACD signal
```

```r
## Performance metrics
subData1 = subData[subData$position != 0,]
subData1$trdValue = rep(0, nrow(subData1))
subData1$pnl = rep(0, nrow(subData1))
subData1$trdDuration = rep(0, nrow(subData1))
for(row_i in 1:nrow(subData1)){
  if(row_i %% 2 == 0){
    subData1$trdValue[row_i] = subData1$price[row_i-1] + subData1$price[row_i]
    subData1$pnl[row_i] = -(subData1$price[row_i-1]*subData1$position[row_i-1] +
                            subData1$position[row_i]*subData1$price[row_i])
    subData1$trdDuration[row_i] = subData1$date[row_i] - subData1$date[row_i-1]
  }
}


# Take all even numbered rows
# This removes trade numbers
subData2 = subData1[subData1$trdNo == 0,]


# Create new trade numbers
subData2$trdNo = seq(1, nrow(subData2))


# Explain about 'seq' function by setting sequence length


# No of trades
nrow(subData2)

## [1] 12

# No of win trades
subData2$pnl>0

##  [1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
## [12]  TRUE

subData2$pnl[subData2$pnl>0]

## [1]  7.028656  7.357422 10.356170 17.848098 70.744812  9.748444  1.056030

mean(subData2$pnl[subData2$pnl>0])

## [1] 17.73423

subData2[subData2$pnl>0,]
```

```
##                 date GOOG.Open priceAr10 priceAr32 priceDiff priceDiff10
## 2007-03-08 2007-03-08  228.7526  226.5593  234.5457 -7.986409   -7.377821
## 2007-03-14 2007-03-14  220.7874  224.0647  232.1600 -8.095313   -7.841791
## 2007-05-02 2007-05-02  233.4500  238.2017  232.9279  5.273796    5.611058
## 2007-06-21 2007-06-21  254.5359  254.4323  244.3376 10.094726   11.780648
## 2007-11-12 2007-11-12  327.6419  350.9495  323.6880 27.261561   29.060764
## 2007-12-21 2007-12-21  347.6370  346.3931  340.0909  6.302252    6.989882
## 2008-01-02 2008-01-02  345.1413  345.1812  341.1790  4.002182    5.533093
##            tradeSignal position    price trdNo trdValue       pnl
## 2007-03-08          -1       -1 228.7526     1 450.4765  7.028656
## 2007-03-14           1        1 220.7874     2 448.9323  7.357422
## 2007-05-02          -1       -1 233.4500     3 456.5438 10.356170
## 2007-06-21          -1       -1 254.5359     5 491.2238 17.848098
## 2007-11-12          -1       -1 327.6419     9 584.5390 70.744812
## 2007-12-21          -1       -1 347.6370    11 685.5255  9.748444
## 2008-01-02          -1        1 345.1413    12 691.3387  1.056030
##            trdDuration
## 2007-03-08           6
## 2007-03-14           5
## 2007-05-02          48
## 2007-06-21          28
## 2007-11-12          90
## 2007-12-21          17
## 2008-01-02           9
```

```
length(subData2$pnl[subData2$pnl > 0])
```

```
## [1] 7
```

```r
# No of loss trades
length(subData2$pnl[subData2$pnl < 0])
```

```
## [1] 5
```

```r
# Winning trades Percentage
length(subData2$pnl[subData2$pnl > 0])/length(subData2$pnl)*100
```

```
## [1] 58.33333
```

```r
# Return in basis points
subData2$rtn = subData2$pnl/subData2$trdValue*100 # Return calculation for each trade
mean(subData2$rtn) # Average return of all trades
```

```
## [1] 0.8497381
```

```r
mean(subData2$rtn)*100  ## Average basis points of all trades
```

```
## [1] 84.97381
```

```r
# Max return trade
max(subData2$pnl) # Maximum pnl trade
```

```
## [1] 70.74481
```

```r
max(subData2$rtn) # Maximum rtn trade
```

```
## [1] 12.10267
```

```r
subData2[subData2$rtn == max(subData2$rtn), ] # When did maximum rtn trade occur
```

```
##                 date GOOG.Open priceAr10 priceAr32 priceDiff priceDiff10
## 2007-11-12 2007-11-12  327.6419  350.9495   323.688  27.26156    29.06076
##            tradeSignal position    price trdNo trdValue      pnl
## 2007-11-12          -1       -1 327.6419     9  584.539 70.74481
##            trdDuration      rtn
## 2007-11-12          90 12.10267
```

```r
subData2[subData2$pnl == max(subData2$pnl), ] # When did maximum pnl trade occur
```

```
##                 date GOOG.Open priceAr10 priceAr32 priceDiff priceDiff10
## 2007-11-12 2007-11-12  327.6419  350.9495   323.688  27.26156    29.06076
##            tradeSignal position    price trdNo trdValue      pnl
## 2007-11-12          -1       -1 327.6419     9  584.539 70.74481
##            trdDuration      rtn
## 2007-11-12          90 12.10267
```

```r
# Worst return trade
min(subData2$pnl) # Minimum pnl trade
```

```
## [1] -22.9241
```

```r
min(subData2$rtn) # Minimum rtn trade
```

```
## [1] -3.939803
```

```r
subData2[subData2$rtn == min(subData2$rtn), ] # When did minimum rtn trade occur?
```

```
##                 date GOOG.Open priceAr10 priceAr32 priceDiff priceDiff10
## 2007-07-20 2007-07-20  254.9942  271.4206  262.7709  8.649795    9.346578
##            tradeSignal position    price trdNo trdValue       pnl
## 2007-07-20          -1       -1 254.9942     7  530.905 -20.91661
##            trdDuration       rtn
## 2007-07-20           2 -3.939803
```

```r
subData2[subData2$pnl == min(subData2$pnl), ] # When did minimum pnl trade occur?
```

```
##                 date GOOG.Open priceAr10 priceAr32 priceDiff priceDiff10
## 2007-12-03 2007-12-03  344.2148  334.2327  336.8448 -2.612047   -3.564843
##            tradeSignal position    price trdNo trdValue      pnl
## 2007-12-03           1        1 344.2148    10 665.5055 -22.9241
##            trdDuration       rtn
## 2007-12-03          20 -3.444615
```

```r
# Max drawdown
# Define max drawdown
# Give example to show how cumsum works


subData2$cumSumPnl = cumsum(subData2$pnl)


# Give example to show how cummax works
pnlDrawdown = subData2$cumSumPnl - cummax(subData2$cumSumPnl)
min(pnlDrawdown)  # PnL Max drawdown
```

```
## [1] -40.4185
```

```r
subData2$cumSumRtn = cumsum(subData2$rtn)
rtnDrawdown = subData2$cumSumRtn - cummax(subData2$cumSumRtn)
min(rtnDrawdown)  # Return Max drawdown
```

```
## [1] -7.595261
```

IMPROVEMENTS

The MACD indicator can be improved by considering triple-exponential moving averages or Welles Wilder's moving average while calculating the difference between parameters (a, b). The signal line, parameter ‘c’, can be then replaced with a smoothened variable of period 5, representing one week. This cancels out multiple noises in the markets.

Another Idea can be to detect the fall of price. With identification of bearish action one can update the code to improve the returns. Though we haven’t tried the same in our code but this idea is reasonable.

Another Idea is to have a double check, i.e. using Relative Vigour Index (RVI) alongside my MACD indicator. The Relative Vigour Index will not diverge greatly to price movement and won’t confuse the users also. Thus, providing a kind of confirmation to take decision on buy, hold or exit.

REFERENCE:

1. Quantitative Trading: How to Build Your Own Algorithmic Trading Business by Ernie Chan
2. Statistical arbitrage: Algorithmic trading insights and techniques by Andrew Pole
3. The Ultimate Algorithmic Trading System Toolbox by George Pruitt
4. Algorithmic and High-Frequency Trading by Álvaro Cartea, Sebastian Jaimungal, José Penalva
5. Multivariate Time Series Analysis by Rues S Tsay
6. Practical Time Series Forecasting with R by Galeit and Kenneth
7. Improving the Strategies of Algorithmic Traders and Investigating Further Realism in their Market Environment by Steven James Alexander Stotter
8. High Frequency Trading by Peter Gomber and Martin Haferkorn
9. A forecast comparison of volatility models: does anything beat a GARCH(1,1) by Peter R. Hansen
10. Algorithmic Trading and DMA by Barry Johnson
11. Mathematics for Economics by Carl P. Simon & Lawrence E. Blume
12. Appel G., Technical Analysis, Financial Times/ Prentice Hall, 2005
13. Dobson E., Appel G., Understanding MACD, Traders Press Inc., 2008
14. Chan, E., Quantitative trading: how to build your own algorithmic trading business
15. Leshik E, and Cralle J, An introduction to algorithmic trading: basic to advanced strategies
16. Pruitt G, The Ultimate Algorithmic Trading System Toolbox
17. Cartea, Á, et al., Algorithmic and High-Frequency Trading
18. Daroczi G., et al. Introduction to R for Quantitative Finance. Packt Publishing
19. Halls, M., Successful Algorithmic Trading, 2017
20. Improving the Macd 2.0 crypto trading algorithm using bearish detection