- 1 Introduction
- 2 Data
- 3 Statiscal Modelling
  - 3.1 Auto Arima Models
  - 3.2 Exponential Smoothing Models
  - 3.3 Univariate Garch Models and EWMA
  - 3.4 Multivariate Garch Models
- 4 Mean Squared Error (MSE)
  - 4.1 Auto Arima Models
  - 4.2 Exponential Smoothing Models
  - 4.3 Univariate Garch Models and EWMA
  - 4.4 Multivariate Garch Models
- 5 Model Comparison
- 6 Conclusion
- 7 Appendix
  - 7.1 Documenting File Creation
  - 7.2 Reference

binary.com Interview Question I (Extention)

®γσ, Lian Hu ®

2018-08-30

1 Introduction

By refer to *Ryo Eng Lian Hu (2017)*[1], I tried to compare few models for financial trading. Today I am using ~~1 minute, 3 minutes, 5 minutes, 10 minutes, 15 minutes, 30 minutes, 1 hour and also daily data instead of~~[2] only daily data. By the way, I try to compare the accuracy of forecasting instead of invesmnent.

Besides, this paper will modelling multivariate Garch compare to previous paper which applied univariate Garch models.

2 Data

I select the data from 2013-01-01 to 2017-08-31[3] via **the Bonnot Gang** *1 Minute Historical Data Download*[4].

Data Summary of 1min, 3mins, 5mins, 10mins, 30mins, 1hr, 1d

Currency	Rows.1m	Cols.1m	Currency	Rows.3m	Cols.3m	Currency	Rows
AUDUSD	1053044	1	AUDUSD	366183	4	AUDUSD	221
EURUSD	986674	1	EURUSD	359214	4	EURUSD	220
GBPUSD	1060889	1	GBPUSD	366116	4	GBPUSD	221
USDCAD	1108344	1	USDCAD	367191	4	USDCAD	221
USDCHF	1106537	1	USDCHF	367092	4	USDCHF	221
USDCNY	654090	1	USDCNY	279772	4	USDCNY	183
USDJPY	1123206	1	USDJPY	367164	4	USDJPY	221

```
## $AUDUSD
##       Index              AUDUSD.Open      AUDUSD.High      AUDUSD.Low
##  Min.    :2013-01-03   Min.    :0.6858   Min.    :0.6867   Min.    :0.6831
##  1st Qu.:2014-03-24   1st Qu.:0.7490   1st Qu.:0.7513   1st Qu.:0.7458
##  Median :2015-04-16   Median :0.7776   Median :0.7808   Median :0.7752
##  Mean   :2015-04-03   Mean   :0.8295   Mean   :0.8325   Mean   :0.8266
##  3rd Qu.:2016-05-03   3rd Qu.:0.9243   3rd Qu.:0.9280   3rd Qu.:0.9215
##  Max.   :2017-05-15   Max.   :1.0585   Max.   :1.0596   Max.   :1.0569
##   AUDUSD.Close
##  Min.   :0.6862
##  1st Qu.:0.7490
##  Median :0.7781
##  Mean   :0.8295
##  3rd Qu.:0.9242
##  Max.   :1.0575
##
## $EURUSD
##       Index              EURUSD.Open     EURUSD.High     EURUSD.Low
##  Min.    :2013-01-03   Min.    :1.040   Min.    :1.040   Min.    :1.039
##  1st Qu.:2014-03-25   1st Qu.:1.100   1st Qu.:1.104   1st Qu.:1.097
##  Median :2015-04-14   Median :1.135   Median :1.139   Median :1.131
##  Mean   :2015-04-02   Mean   :1.202   Mean   :1.205   Mean   :1.199
##  3rd Qu.:2016-05-01   3rd Qu.:1.328   3rd Qu.:1.332   3rd Qu.:1.325
##  Max.   :2017-05-15   Max.   :1.393   Max.   :1.399   Max.   :1.391
##   EURUSD.Close
##  Min.   :1.039
##  1st Qu.:1.101
##  Median :1.135
##  Mean   :1.202
##  3rd Qu.:1.329
##  Max.   :1.393
##
```

```
## $GBPUSD
##      Index              GBPUSD.Open     GBPUSD.High     GBPUSD.Low
##  Min.   :2013-01-03   Min.   :1.204   Min.   :1.206   Min.   :1.203
##  1st Qu.:2014-03-25   1st Qu.:1.426   1st Qu.:1.431   1st Qu.:1.421
##  Median :2015-04-16   Median :1.530   Median :1.535   Median :1.525
##  Mean   :2015-04-03   Mean   :1.499   Mean   :1.503   Mean   :1.495
##  3rd Qu.:2016-05-03   3rd Qu.:1.602   3rd Qu.:1.606   3rd Qu.:1.598
##  Max.   :2017-05-15   Max.   :1.716   Max.   :1.719   Max.   :1.715
##   GBPUSD.Close
##  Min.   :1.205
##  1st Qu.:1.426
##  Median :1.530
##  Mean   :1.499
##  3rd Qu.:1.601
##  Max.   :1.716
##
## $USDCAD
##      Index              USDCAD.Open       USDCAD.High       USDCAD.Low
##  Min.   :2013-01-03   Min.   :0.9827   Min.   :0.9837   Min.   :0.9815
##  1st Qu.:2014-03-24   1st Qu.:1.0869   1st Qu.:1.0887   1st Qu.:1.0844
##  Median :2015-04-18   Median :1.2439   Median :1.2461   Median :1.2404
##  Mean   :2015-04-03   Mean   :1.2030   Mean   :1.2060   Mean   :1.1999
##  3rd Qu.:2016-05-03   3rd Qu.:1.3148   3rd Qu.:1.3185   3rd Qu.:1.3115
##  Max.   :2017-05-15   Max.   :1.4579   Max.   :1.4582   Max.   :1.4572
##   USDCAD.Close
##  Min.   :0.9832
##  1st Qu.:1.0870
##  Median :1.2437
##  Mean   :1.2030
##  3rd Qu.:1.3146
##  Max.   :1.4577
##
## $USDCHF
##      Index              USDCHF.Open       USDCHF.High       USDCHF.Low
##  Min.   :2013-01-03   Min.   :0.8537   Min.   :0.8622   Min.   :0.8383
##  1st Qu.:2014-03-24   1st Qu.:0.9209   1st Qu.:0.9234   1st Qu.:0.9173
##  Median :2015-04-19   Median :0.9585   Median :0.9617   Median :0.9544
##  Mean   :2015-04-04   Mean   :0.9525   Mean   :0.9550   Mean   :0.9496
##  3rd Qu.:2016-05-03   3rd Qu.:0.9843   3rd Qu.:0.9875   3rd Qu.:0.9815
##  Max.   :2017-05-15   Max.   :1.0298   Max.   :1.0327   Max.   :1.0296
##   USDCHF.Close
##  Min.   :0.8581
##  1st Qu.:0.9208
##  Median :0.9583
##  Mean   :0.9524
##  3rd Qu.:0.9839
##  Max.   :1.0302
##
## $USDCNY
##      Index              USDCNY.Open     USDCNY.High     USDCNY.Low
##  Min.   :2013-01-03   Min.   :6.036   Min.   :6.042   Min.   :6.035
##  1st Qu.:2014-03-17   1st Qu.:6.146   1st Qu.:6.155   1st Qu.:6.136
##  Median :2015-03-31   Median :6.223   Median :6.232   Median :6.217
##  Mean   :2015-03-19   Mean   :6.352   Mean   :6.362   Mean   :6.343
##  3rd Qu.:2016-04-06   3rd Qu.:6.524   3rd Qu.:6.536   3rd Qu.:6.519
##  Max.   :2017-05-15   Max.   :6.957   Max.   :6.963   Max.   :6.957
##   USDCNY.Close
##  Min.   :6.041
##  1st Qu.:6.147
##  Median :6.223
##  Mean   :6.353
##  3rd Qu.:6.530
##  Max.   :6.957
##
## $USDJPY
##      Index              USDJPY.Open       USDJPY.High       USDJPY.Low
##  Min.   :2013-01-03   Min.   : 87.06   Min.   : 87.16   Min.   : 87.04
##  1st Qu.:2014-03-27   1st Qu.:101.83   1st Qu.:102.04   1st Qu.:101.48
##  Median :2015-04-20   Median :107.92   Median :108.25   Median :107.50
##  Mean   :2015-04-05   Mean   :108.99   Mean   :109.31   Mean   :108.64
##  3rd Qu.:2016-05-05   3rd Qu.:118.28   3rd Qu.:118.61   3rd Qu.:117.98
##  Max.   :2017-05-15   Max.   :125.65   Max.   :125.81   Max.   :125.59
##   USDJPY.Close
##  Min.   : 87.12
##  1st Qu.:101.82
##  Median :107.89
##  Mean   :109.00
##  3rd Qu.:118.32
##  Max.   :125.61
```

Due to the dataset always happened errors and not completed somemore wrong figures, here I get the daily data from 2012-01-01 to 2017-08-31 via Yahoo. However only take the data from 2014-01-01 to 2017-08-31 as experiments.

```
## Read yahooData dataset.
USDAUD <- readRDS('./data/USDAUD.rds')
USDEUR <- readRDS('./data/USDEUR.rds')
USDGBP <- readRDS('./data/USDGBP.rds')
USDCAD <- readRDS('./data/USDCAD.rds')
USDCHF <- readRDS('./data/USDCHF.rds')
USDCNY <- readRDS('./data/USDCNY.rds')
USDJPY <- readRDS('./data/USDJPY.rds')

mbase <- list(USDAUD = USDAUD, USDEUR = USDEUR, USDGBP = USDGBP,
              USDCAD = USDCAD, USDCHF = USDCHF, USDCNY = USDCNY,
              USDJPY = USDJPY)
rm(USDAUD, USDEUR, USDGBP, USDCAD, USDCHF, USDCNY, USDJPY)

llply(mbase, summary)
```

```
## $USDAUD
##      Index              USDAUD.Open      USDAUD.High      USDAUD.Low
##  Min.   :2012-01-02   Min.   :0.925   Min.   :0.927   Min.   :0.921
##  1st Qu.:2013-05-31   1st Qu.:1.037   1st Qu.:1.042   1st Qu.:1.031
##  Median :2014-10-31   Median :1.148   Median :1.153   Median :1.142
##  Mean   :2014-10-31   Mean   :1.176   Mean   :1.181   Mean   :1.171
##  3rd Qu.:2016-03-30   3rd Qu.:1.322   3rd Qu.:1.327   3rd Qu.:1.316
##  Max.   :2017-08-30   Max.   :1.458   Max.   :1.464   Max.   :1.447
##   USDAUD.Close    USDAUD.Volume USDAUD.Adjusted
##  Min.   :0.9253   Min.   :0     Min.   :0.9253
##  1st Qu.:1.0369   1st Qu.:0     1st Qu.:1.0369
##  Median :1.1478   Median :0     Median :1.1478
##  Mean   :1.1759   Mean   :0     Mean   :1.1759
##  3rd Qu.:1.3216   3rd Qu.:0     3rd Qu.:1.3216
##  Max.   :1.4575   Max.   :0     Max.   :1.4575
##
## $USDEUR
##      Index              USDEUR.Open       USDEUR.High       USDEUR.Low
##  Min.   :2012-01-02   Min.   :0.7180   Min.   :0.7190   Min.   :0.7150
##  1st Qu.:2013-05-31   1st Qu.:0.7580   1st Qu.:0.7610   1st Qu.:0.7560
##  Median :2014-10-31   Median :0.8080   Median :0.8130   Median :0.8050
##  Mean   :2014-10-31   Mean   :0.8285   Mean   :0.8318   Mean   :0.8256
##  3rd Qu.:2016-03-30   3rd Qu.:0.8980   3rd Qu.:0.9020   3rd Qu.:0.8940
##  Max.   :2017-08-30   Max.   :0.9620   Max.   :1.3150   Max.   :0.9600
##   USDEUR.Close    USDEUR.Volume USDEUR.Adjusted
##  Min.   :0.7178   Min.   :0     Min.   :0.7178
##  1st Qu.:0.7582   1st Qu.:0     1st Qu.:0.7582
##  Median :0.8081   Median :0     Median :0.8081
##  Mean   :0.8285   Mean   :0     Mean   :0.8285
##  3rd Qu.:0.8981   3rd Qu.:0     3rd Qu.:0.8981
##  Max.   :0.9624   Max.   :0     Max.   :0.9624
##
## $USDGBP
##      Index              USDGBP.Open       USDGBP.High       USDGBP.Low
##  Min.   :2012-01-02   Min.   :0.5830   Min.   :0.5830   Min.   :0.5820
##  1st Qu.:2013-05-30   1st Qu.:0.6250   1st Qu.:0.6260   1st Qu.:0.6230
##  Median :2014-10-31   Median :0.6460   Median :0.6490   Median :0.6440
##  Mean   :2014-10-30   Mean   :0.6702   Mean   :0.6732   Mean   :0.6681
##  3rd Qu.:2016-03-30   3rd Qu.:0.6950   3rd Qu.:0.6990   3rd Qu.:0.6920
##  Max.   :2017-08-30   Max.   :0.8310   Max.   :1.5690   Max.   :0.8270
##   USDGBP.Close    USDGBP.Volume USDGBP.Adjusted
##  Min.   :0.5827   Min.   :0     Min.   :0.5827
##  1st Qu.:0.6247   1st Qu.:0     1st Qu.:0.6247
##  Median :0.6463   Median :0     Median :0.6463
##  Mean   :0.6702   Mean   :0     Mean   :0.6702
##  3rd Qu.:0.6952   3rd Qu.:0     3rd Qu.:0.6952
##  Max.   :0.8306   Max.   :0     Max.   :0.8306
##
## $USDCAD
##      Index              USDCAD.Open      USDCAD.High      USDCAD.Low
##  Min.   :2012-01-02   Min.   :0.968   Min.   :0.971   Min.   :0.963
##  1st Qu.:2013-05-30   1st Qu.:1.029   1st Qu.:1.032   1st Qu.:1.026
##  Median :2014-10-30   Median :1.127   Median :1.131   Median :1.123
##  Mean   :2014-10-30   Mean   :1.167   Mean   :1.171   Mean   :1.164
##  3rd Qu.:2016-03-29   3rd Qu.:1.308   3rd Qu.:1.313   3rd Qu.:1.303
##  Max.   :2017-08-30   Max.   :1.458   Max.   :1.469   Max.   :1.449
##   USDCAD.Close    USDCAD.Volume USDCAD.Adjusted
##  Min.   :0.9683   Min.   :0     Min.   :0.9683
```

```
##  1st Qu.:1.0286   1st Qu.:0     1st Qu.:1.0286
##  Median :1.1263   Median :0     Median :1.1263
##  Mean   :1.1673   Mean   :0     Mean   :1.1673
##  3rd Qu.:1.3076   3rd Qu.:0     3rd Qu.:1.3076
##  Max.   :1.4578   Max.   :0     Max.   :1.4578
## 
## $USDCHF
##      Index             USDCHF.Open      USDCHF.High       USDCHF.Low    
##  Min.   :2012-01-02   Min.   :0.8540   Min.   :0.8710   Min.   :0.7330  
##  1st Qu.:2013-06-03   1st Qu.:0.9220   1st Qu.:0.9250   1st Qu.:0.9180  
##  Median :2014-11-03   Median :0.9540   Median :0.9570   Median :0.9500  
##  Mean   :2014-11-01   Mean   :0.9504   Mean   :0.9539   Mean   :0.9469  
##  3rd Qu.:2016-03-31   3rd Qu.:0.9780   3rd Qu.:0.9810   3rd Qu.:0.9730  
##  Max.   :2017-08-30   Max.   :1.0300   Max.   :1.0330   Max.   :1.0280  
##   USDCHF.Close    USDCHF.Volume USDCHF.Adjusted
##  Min.   :0.8544   Min.   :0     Min.   :0.8544
##  1st Qu.:0.9216   1st Qu.:0     1st Qu.:0.9216
##  Median :0.9538   Median :0     Median :0.9538
##  Mean   :0.9504   Mean   :0     Mean   :0.9504
##  3rd Qu.:0.9775   3rd Qu.:0     3rd Qu.:0.9775
##  Max.   :1.0302   Max.   :0     Max.   :1.0302
## 
## $USDCNY
##      Index             USDCNY.Open     USDCNY.High      USDCNY.Low   
##  Min.   :2012-01-02   Min.   :6.031   Min.   :6.040   Min.   :2.201  
##  1st Qu.:2013-05-29   1st Qu.:6.189   1st Qu.:6.195   1st Qu.:6.185  
##  Median :2014-10-30   Median :6.284   Median :6.295   Median :6.270  
##  Mean   :2014-10-30   Mean   :6.365   Mean   :6.375   Mean   :6.355  
##  3rd Qu.:2016-03-30   3rd Qu.:6.524   3rd Qu.:6.529   3rd Qu.:6.515  
##  Max.   :2017-08-30   Max.   :7.478   Max.   :7.481   Max.   :6.945  
##   USDCNY.Close   USDCNY.Volume USDCNY.Adjusted
##  Min.   :6.031   Min.   :0     Min.   :6.031  
##  1st Qu.:6.190   1st Qu.:0     1st Qu.:6.190  
##  Median :6.285   Median :0     Median :6.285  
##  Mean   :6.365   Mean   :0     Mean   :6.365  
##  3rd Qu.:6.524   3rd Qu.:0     3rd Qu.:6.524  
##  Max.   :6.960   Max.   :0     Max.   :6.960  
## 
## $USDJPY
##      Index             USDJPY.Open       USDJPY.High        USDJPY.Low    
##  Min.   :2012-01-02   Min.   : 76.18   Min.   : 76.20   Min.   : 76.05  
##  1st Qu.:2013-05-29   1st Qu.: 97.86   1st Qu.: 98.29   1st Qu.: 97.46  
##  Median :2014-10-30   Median :103.91   Median :104.19   Median :103.54  
##  Mean   :2014-10-30   Mean   :103.71   Mean   :104.07   Mean   :103.32  
##  3rd Qu.:2016-03-30   3rd Qu.:114.27   3rd Qu.:114.72   3rd Qu.:113.74  
##  Max.   :2017-08-30   Max.   :125.60   Max.   :125.82   Max.   :124.97  
##   USDJPY.Close    USDJPY.Volume USDJPY.Adjusted
##  Min.   : 76.18   Min.   :0     Min.   : 76.18 
##  1st Qu.: 97.85   1st Qu.:0     1st Qu.: 97.85 
##  Median :103.93   Median :0     Median :103.93 
##  Mean   :103.71   Mean   :0     Mean   :103.71 
##  3rd Qu.:114.24   3rd Qu.:0     3rd Qu.:114.24 
##  Max.   :125.63   Max.   :0     Max.   :125.63
```

3 Statiscal Modelling

3.1 Auto Arima Models

Below I use arima model to analyse Yahoo data.

```
## Using closing price to forecast.
.baseDate = '2014-01-01 00:00:00'
.baseDate = ymd(ymd_hms(.baseDate))

## 3 months
yahooD3M <- llply(mbase, function(x) {
  z = simAutoArima(x, .prCat = 'Cl',
                   .baseDate = .baseDate, .verbose = TRUE,
                   .maPeriod = 'months', .unit = 3)
  nm = names(Cl(x)) %>% str_replace_all('.Close', '')
  saveRDS(z, paste0('./data/', nm, '.yahooAutoArima.d3m.rds'))
  cat(paste0('Saved... ./data/', nm, '.yahooAutoArima.d3m.rds\n'))
})


## 6 months
yahooD6M <- llply(mbase, function(x) {
```

```
yahooD6M <- llply(mbase, function(x) {
  z = simAutoArima(x, .prCat = 'Cl',
                 .baseDate = .baseDate, .verbose = TRUE,
                 .maPeriod = 'months', .unit = 6)
  nm = names(Cl(x)) %>% str_replace_all('.Close', '')
  saveRDS(z, paste0('./data/', nm, '.yahooAutoArima.d6m.rds'))
  cat(paste0('Saved... ./data/', nm, '.yahooAutoArima.d6m.rds\n'))
})


## 1 year
yahooD1Y <- llply(mbase, function(x) {
  z = simAutoArima(x, .prCat = 'Cl',
               .baseDate = .baseDate, .verbose = TRUE,
               .maPeriod = 'years', .unit = 1)
  nm = names(Cl(x)) %>% str_replace_all('.Close', '')
  saveRDS(z, paste0('./data/', nm, '.yahooAutoArima.d1y.rds'))
  cat(paste0('Saved... ./data/', nm, '.yahooAutoArima.d1y.rds\n'))
  })


## 18 months
yahooD18M <- llply(mbase, function(x) {
  z = simAutoArima(x, .prCat = 'Cl',
               .baseDate = .baseDate, .verbose = TRUE,
               .maPeriod = 'months', .unit = 18)
  nm = names(Cl(x)) %>% str_replace_all('.Close', '')
  saveRDS(z, paste0('./data/', nm, '.yahooAutoArima.d18m.rds'))
  cat(paste0('Saved... ./data/', nm, '.yahooAutoArima.d18m.rds\n'))
  })


## 2 years
yahooD2Y <- llply(mbase, function(x) {
  z = simAutoArima(x, .prCat = 'Cl',
               .baseDate = .baseDate, .verbose = TRUE,
               .maPeriod = 'years', .unit = 2)
  nm = names(Cl(x)) %>% str_replace_all('.Close', '')
  saveRDS(z, paste0('./data/', nm, '.yahooAutoArima.d2y.rds'))
  cat(paste0('Saved... ./data/', nm, '.yahooAutoArima.d2y.rds\n'))
  })
```

3.2 Exponential Smoothing Models

Now I analyse the Yahoo data by using Exponential Time Series Smoothing models.

3.3 Univariate Garch Models and EWMA

```
## Using closing price to forecast.
#'@ source('./function/Garch.d3m.R')
AUDUSD.Garch.d3m <- readRDS('./data/AUDUSD.Garch.d3m.rds')
EURUSD.Garch.d3m <- readRDS('./data/EURUSD.Garch.d3m.rds')
GBPUSD.Garch.d3m <- readRDS('./data/GBPUSD.Garch.d3m.rds')
USDCAD.Garch.d3m <- readRDS('./data/USDCAD.Garch.d3m.rds')
USDCHF.Garch.d3m <- readRDS('./data/USDCHF.Garch.d3m.rds')
USDCNY.Garch.d3m <- readRDS('./data/USDCNY.Garch.d3m.rds')
USDJPY.Garch.d3m <- readRDS('./data/USDJPY.Garch.d3m.rds')
```

```
## Using closing price to forecast.
#'@ source('./function/Garch.d6m.R')
AUDUSD.Garch.d6m <- readRDS('./data/AUDUSD.Garch.d6m.rds')
EURUSD.Garch.d6m <- readRDS('./data/EURUSD.Garch.d6m.rds')
GBPUSD.Garch.d6m <- readRDS('./data/GBPUSD.Garch.d6m.rds')
USDCAD.Garch.d6m <- readRDS('./data/USDCAD.Garch.d6m.rds')
USDCHF.Garch.d6m <- readRDS('./data/USDCHF.Garch.d6m.rds')
USDCNY.Garch.d6m <- readRDS('./data/USDCNY.Garch.d6m.rds')
USDJPY.Garch.d6m <- readRDS('./data/USDJPY.Garch.d6m.rds')
```

```
## Using closing price to forecast.
#'@ source('./function/Garch.d1y.R')
AUDUSD.Garch.d1y <- readRDS('./data/AUDUSD.Garch.d1y.rds')
EURUSD.Garch.d1y <- readRDS('./data/EURUSD.Garch.d1y.rds')
GBPUSD.Garch.d1y <- readRDS('./data/GBPUSD.Garch.d1y.rds')
USDCAD.Garch.d1y <- readRDS('./data/USDCAD.Garch.d1y.rds')
USDCHF.Garch.d1y <- readRDS('./data/USDCHF.Garch.d1y.rds')
USDCNY.Garch.d1y <- readRDS('./data/USDCNY.Garch.d1y.rds')
```

```
USDJPY.Garch.dly <- readRDS('./data/USDJPY.Garch.dly.rds')
```

3.4 Multivariate Garch Models

Kindly refer to paper binary.com Interview Question I - Multivariate GARCH Models.

4 Mean Squared Error (MSE)

4.1 Auto Arima Models

Now we look at the accuracy of prediction.

MSE Table

Currency	MSE.3M	MSE.6M	MSE.1Y
AUDUSD.AutoArima	0.0000252	0.0000245	0.0000243
EURUSD.AutoArima	0.0000390	0.0000393	0.0000385
GBPUSD.AutoArima	0.0000722	0.0000689	0.0000676
USDCAD.AutoArima	0.0000434	0.0000423	0.0000416
USDCHF.AutoArima	0.0000589	0.0000586	0.0000564
USDCNY.AutoArima	0.0002294	0.0002117	0.0002072
USDJPY.AutoArima	0.4468217	0.4273710	0.4172619
Mean =	0.0638985	0.0611166	0.0596711

Below is the summary of Yahoo data.

MSE Table

Currency	MSE.3M	MSE.6M	MSE.1Y	MSE.18M	MSE
USDAUD.yahooAutoArima	0.0000725	0.0000691	0.0000680	0.0000683	0.0000
USDCAD.yahooAutoArima	0.0000426	0.0000416	0.0000411	0.0000411	0.0000
USDCHF.yahooAutoArima	0.0000581	0.0000562	0.0000569	0.0000564	0.0000
USDCNY.yahooAutoArima	0.0002035	0.0001980	0.0001953	0.0001981	0.0001
USDEUR.yahooAutoArima	0.0000268	0.0000264	0.0000261	0.0000268	0.0000
USDGBP.yahooAutoArima	0.0000219	0.0000203	0.0000190	0.0000194	0.0000
USDJPY.yahooAutoArima	0.4893502	0.4700268	0.4593798	0.4691871	0.4666
Mean =	0.0699680	0.0672055	0.0656837	0.0670853	0.0667

4.2 Exponential Smoothing Models

Similar with univariate `auto.arima` model, here we try to compare the the accuracy of prediction of univariate `ets` models among the portfolios where each portfolio contains USD exchanged with other currencies.

MSE Table

Model	MSE.3M	MSE.6M	MSE.1Y
AAN	0.0638216	0.0602470	0.0596556
AAZ	0.0638216	0.0602470	0.0596556
ANN	0.0580867	0.0577726	0.0577719
ANZ	0.0580867	0.0577726	0.0577719
AZN	0.0620494	0.0589684	0.0585482
AZZ	0.0620494	0.0589684	0.0585482
MAN	0.0641544	0.0604866	0.0598502
MAZ	0.0641544	0.0604866	0.0598502

MMN	0.0646267	0.0602590	0.0591698
MMZ	0.0646267	0.0602590	0.0591698
MNN	0.0580857	0.0577605	0.0577702
MNZ	0.0580857	0.0577605	0.0577702
MZN	0.0622906	0.0589968	0.0586481
MZZ	0.0622906	0.0589968	0.0586481
ZAN	0.0640052	0.0603903	0.0597147
ZAZ	0.0640052	0.0603903	0.0597147
ZMN	0.0646267	0.0602590	0.0591698
ZMZ	0.0646267	0.0602590	0.0591698
ZNN	0.0580782	0.0577591	0.0577670
ZNZ	0.0580782	0.0577591	0.0577670
ZZN	0.0622681	0.0590432	0.0586071
ZZZ	0.0622681	0.0590432	0.0586071

Application of ETS models onto the Yahoo data as below.

MSE Table

Model	MSE.3M	MSE.6M	MSE.1Y	MSE.18M	MSE.2Y
AAN	0.1277581	0.2158374	0.7374775	2.167684	5.761272
ANN	0.1188879	0.2085196	0.7365478	2.176026	5.740889
MAN	0.1288244	0.2155510	0.7361620	NA	NA
MNN	0.1187965	0.2083754	0.7365477	2.176137	5.740912
ZNN	0.1188117	0.2084939	0.7365720	2.175976	5.740875

Application of ETS models onto the Yahoo data as below.

MSE Table

Model	MSE.3M	MSE.6M	MSE.1Y	MSE.18M	MSE.2Y
AAN	0.1277581	0.2158374	0.7374775	2.167684	5.761272
ANN	0.1188879	0.2085196	0.7365478	2.176026	5.740889
MAN	0.1288244	0.2155510	0.7361620	NA	NA
MNN	0.1187965	0.2083754	0.7365477	2.176137	5.740912
ZNN	0.1188117	0.2084939	0.7365720	2.175976	5.740875

4.3 Univariate Garch Models and EWMA

Few forecasted data price had bias standard error to cause the result not accutate.

MSE Table

Model	MSE.3M	MSE.6M	MSE.1Y
csGARCH	1.009370	1.009370	NA
eGARCH	7.975536	7.975536	7.975536
gjrGARCH	NA	1802.384383	NA
iGARCH	NA	1802.384383	NA
sGARCH	1802.384383	NA	NA

4.4 Multivariate Garch Models

Kindly refer to paper binary.com Interview Question I - Multivariate GARCH Models.

5 Model Comparison

```
## set all models provided by ets function.
ets.m <- c('AAN', 'AAZ', 'ANN', 'ANZ', 'AZN', 'AZZ', 'MAN', 'MAZ', 'MMN',
           'MMZ', 'MNN', 'MNZ', 'MZN', 'MZZ', 'ZAN', 'ZAZ', 'ZMN', 'ZMZ',
           'ZNN', 'ZNZ', 'ZZN', 'ZZZ')

## Multiple Garch models inside `rugarch` package.
garch.m <- c('sGARCH', 'fGARCH', 'eGARCH', 'gjrGARCH', 'apARCH', 'iGARCH', 'csGARC
H', 'realGARCH')
garch.m <- garch.m[-length(garch.m)] #skip `realGARCH`

solver <- c('hybrid', 'solnp', 'nlminb', 'gosolnp', 'nloptr', 'lbfgs')

sub.garch.m <<- c('GARCH', 'TGARCH', 'AVGARCH', 'NGARCH', 'NAGARCH', 'APARCH', 'GJ
RGARCH', 'ALLGARCH')

dist.model <- c('norm', 'snorm', 'std', 'sstd', 'ged', 'sged', 'nig', 'ghyp', 'js
u')
```

```
getSymbols('JPY=X', from = Sys.Date() %m-% years(1), to = Sys.Date())
USDJPY <- `JPY=X` %>% Cl %>% na.omit; rm(`JPY=X`)
names(USDJPY) %<>% str_replace_all('JPY=X', 'USDJPY')

## Auto Arima
pre1 <- auto.arima(USDJPY)
saveRDS(pre1, './data/pre1.rds')

## ETS
pre2 <- llply(ets.m, function(x) ets(USDJPY, model = x))
names(pre2) = ets.m
saveRDS(pre2, './data/pre2.rds')

## Garch
#'@ pre3 <- llply(garch.m, function(x){
#'@   gm = llply(dist.model, function(y){
#'@       if(x == 'fGARCH') {
#'@         sgm = llply(sub.garch.m, function(z) {
#'@           spec = ugarchspec(variance.model = list(
#'@             model = x, garchOrder = c(1, 1),
#'@             submodel = z, external.regressors = NULL,
#'@             variance.targeting = FALSE), distribution.model = y)
#'@           fit = ugarchfit(spec, USDJPY, solver = 'hybrid')
#'@             })
#'@         names(sgm) = sub.garch.m; sgm
#'@           } else {
#'@             spec = ugarchspec(variance.model = list(
#'@               model = x, garchOrder = c(1, 1),
#'@               submodel = NULL, external.regressors = NULL,
#'@               variance.targeting = FALSE), distribution.model = y)
#'@             fit = ugarchfit(spec, USDJPY, solver = 'hybrid')
#'@           }
#'@       })
#'@   names(gm) = dist.model; gm
#'@   })
#'@ names(pre3) = garch.m

## Garch models
pre3b <- llply(garch.m, function(x){
  armaOrder = armaSearch(USDJPY)
  armaOrder %<>% dplyr::filter(AIC==min(AIC)) %>% .[c('p', 'q')] %>% unlist
  spec = ugarchspec(
    variance.model = list(
      model = x, garchOrder = c(1, 1),
      submodel = NULL, external.regressors = NULL,
      variance.targeting = FALSE),
    mean.model = list(
      armaOrder = armaOrder,
      include.mean = TRUE, archm = FALSE,
      archpow = 1, arfima = FALSE,
      external.regressors = NULL,
      archex = FALSE),
    distribution.model = 'snorm')
```

```
      distribution.model = 'snorm')
      fit = ugarchfit(spec, USDJPY, solver = 'hybrid')
    }
  })
saveRDS(pre3b, './data/pre3b.rds')

## gjrGARCH model's distributions...
pre3c <- llply(dist.model, function(x){
  armaOrder = armaSearch(USDJPY)
  armaOrder %<>% dplyr::filter(AIC==min(AIC)) %>% .[c('p', 'q')] %>% unlist
  spec = ugarchspec(
    variance.model = list(
      model = 'gjrGARCH', garchOrder = c(3, 3),
      submodel = NULL, external.regressors = NULL,
      variance.targeting = FALSE),
    mean.model = list(
      armaOrder = armaOrder,
      include.mean = TRUE, archm = FALSE,
      archpow = 1, arfima = FALSE,
      external.regressors = NULL,
      archex = FALSE),
    distribution.model = x)
  fit = ugarchfit(spec, USDJPY, solver = 'hybrid')
  })
saveRDS(pre3c, './data/pre3c.rds')

## gjrGARCH model's distributions...
pre3d <- llply(solver, function(x){
  armaOrder = armaSearch(USDJPY)
  armaOrder %<>% dplyr::filter(AIC==min(AIC)) %>% .[c('p', 'q')] %>% unlist
  spec = ugarchspec(
    variance.model = list(
      model = 'gjrGARCH', garchOrder = c(3, 3),
      submodel = NULL, external.regressors = NULL,
      variance.targeting = FALSE),
    mean.model = list(
      armaOrder = armaOrder,
      include.mean = TRUE, archm = FALSE,
      archpow = 1, arfima = FALSE,
      external.regressors = NULL,
      archex = FALSE),
    distribution.model = 'snorm')
  fit = ugarchfit(spec, USDJPY, solver = x)
  })
saveRDS(pre3d, './data/pre3d.rds')
```

```
pre1 <- readRDS('./data/pre1.rds')
pre2 <- readRDS('./data/pre2.rds')
pre3 <- readRDS('./data/pre3.rds') %>% unlist
pre3b <- readRDS('./data/pre3b.rds') %>% unlist
names(pre3b) = paste0(names(pre3b),'.optimalArmaPQ')
pre3c <- readRDS('./data/pre3c.rds') %>% unlist
names(pre3c) = paste0('gjrGARCH.', dist.model)
pre3d <- readRDS('./data/pre3d.rds') %>% unlist
names(pre3d) = paste0('gjrGARCH.', solver)
```

```
aic1 <- data.frame(.id = 'Auto.Arima', V1 = AIC(pre1))
aic2 <- ldply(pre2, AIC)
aic3 <- ldply(pre3, function(x) infocriteria(x)[1])
aic3b <- ldply(pre3b, function(x) infocriteria(x)[1])
aic3c <- ldply(pre3c, function(x) infocriteria(x)[1])
aic3d <- ldply(pre3d, function(x) tryCatch(infocriteria(x), error = function(e) NA
)[1]) %>% na.omit

aic.s <- bind_rows(aic1, aic2, aic3, aic3b, aic3c, aic3d)
names(aic.s) <- c('.id', 'AIC')

aic.s %>% formattable(list(

    .id = color_tile('white', 'darkgoldenrod'),

    AIC = formatter('span', style = x ~ formattable::style(
        color = ifelse(rank(x) >= 3, 'gray', 'green')),
        x ~ sprintf("%.6f (rank: %01.0f)", x, rank(x)))))
```

.id	AIC
Auto.Arima	530.718477 (rank: 42)

AAN	1240.940439 (rank: 64)
AAZ	1240.940439 (rank: 64)
ANN	1238.362466 (rank: 56)
ANZ	1238.362466 (rank: 56)
AZN	1238.362466 (rank: 56)
AZZ	1238.362466 (rank: 56)
MAN	1238.414089 (rank: 60)
MAZ	1238.414089 (rank: 60)
MMN	1237.532891 (rank: 52)
MMZ	1237.532891 (rank: 52)
MNN	1237.141959 (rank: 46)
MNZ	1237.141959 (rank: 46)
MZN	1237.141959 (rank: 46)
MZZ	1237.141959 (rank: 46)
ZAN	1238.414089 (rank: 60)
ZAZ	1238.414089 (rank: 60)
ZMN	1237.532891 (rank: 52)
ZMZ	1237.532891 (rank: 52)
ZNN	1237.141959 (rank: 46)
ZNZ	1237.141959 (rank: 46)
ZZN	1237.141959 (rank: 46)
ZZZ	1237.141959 (rank: 46)
sGARCH	2.056182 (rank: 29)
fGARCH.GARCH	2.056182 (rank: 30)
fGARCH.TGARCH	2.055787 (rank: 27)
fGARCH.AVGARCH	2.063331 (rank: 36)
fGARCH.NGARCH	2.052349 (rank: 25)
fGARCH.NAGARCH	2.050824 (rank: 24)
fGARCH.APARCH	2.058716 (rank: 32)
fGARCH.GJRGARCH	2.057462 (rank: 31)
fGARCH.ALLGARCH	2.062901 (rank: 35)
eGARCH	2.066497 (rank: 37)
gjrGARCH	2.037765 (rank: 19)
apARCH	2.058972 (rank: 33)
iGARCH	2.055883 (rank: 28)
csGARCH	2.073589 (rank: 38)
sGARCH.optimalArmaPQ	1.975899 (rank: 9)
fGARCH.GARCH.optimalArmaPQ	1.975546 (rank: 8)
fGARCH.TGARCH.optimalArmaPQ	1.945521 (rank: 3)
fGARCH.AVGARCH.optimalArmaPQ	1.958105 (rank: 4)
fGARCH.NGARCH.optimalArmaPQ	1.941874 (rank: 2)
fGARCH.NAGARCH.optimalArmaPQ	1.983257 (rank: 11)
fGARCH.APARCH.optimalArmaPQ	2.010005 (rank: 15)
fGARCH.GJRGARCH.optimalArmaPQ	1.964480 (rank: 6)
fGARCH.ALLGARCH.optimalArmaPQ	2.042566 (rank: 22)
eGARCH.optimalArmaPQ	1.973064 (rank: 7)
gjrGARCH.optimalArmaPQ	1.931967 (rank: 1)

[illegible]	[illegible]
apARCH.optimalArmaPQ	2.062426 (rank: 34)
iGARCH.optimalArmaPQ	1.958820 (rank: 5)
csGARCH.optimalArmaPQ	1.980004 (rank: 10)
gjrGARCH.norm	2.040296 (rank: 21)
gjrGARCH.snorm	2.003560 (rank: 13)
gjrGARCH.std	2.054888 (rank: 26)
gjrGARCH.sstd	2.085194 (rank: 39)
gjrGARCH.ged	2.043311 (rank: 23)
gjrGARCH.sged	2.032038 (rank: 18)
gjrGARCH.nig	2.040235 (rank: 20)
gjrGARCH.ghyp	2.021680 (rank: 16)
gjrGARCH.jsu	2.117652 (rank: 41)
gjrGARCH.hybrid	2.003560 (rank: 13)
gjrGARCH.solnp	2.003560 (rank: 13)
gjrGARCH.gosolnp	2.029092 (rank: 17)
gjrGARCH.nloptr	2.088449 (rank: 40)

6 Conclusion

Due to the prediction result abnormal in Univariate Garch Models and EWMA, here I unable to determine the best fit data size for GARCH model. Therefore I refer to auto.arima model and ETS models to choose `MSE.1Y` is the best fit data size.

in order to cope with the problem. I compare the `AIC` value. Due to ETS and Arima model do not measure the volatility, there is totally different models and the `AIC` value thousands times GARCH models. The paper concludes that *gjrGARCH* is the best model.

- gjrGARCH `2.037765 (rank: 19)` is the best GARCH model (without adjust arma order) among the rest.
- gjrGARCH with order order `1.931967 (rank: 01)` is the best fit model.
- gjrGARCH with `snorm` or `hybrid` or `solnp` distribution is the best among using other distributions.

```
gjrGARCH.optimalArmaPQ        1.931967 (rank: 01)
fGARCH.NGARCH.optimalArmaPQ   1.941874 (rank: 02)
gjrGARCH.snorm                2.00356 (rank: 13)
gjrGARCH.hybrid               2.00356 (rank: 13)
gjrGARCH.solnp                2.00356 (rank: 13)
gjrGARCH                      2.037765 (rank: 19)
fGARCH.NAGARCH                2.050824 (rank: 24)
```

binary.com Interview Question I - Comparison of Univariate GARCH Models is the next paper to adjust the `d` value for arma order which determine the best lag.

7 Appendix

7.1 Documenting File Creation

It's useful to record some information about how your file was created.

- File creation date: 2015-07-22
- File latest updated date: 2018-08-30
- R version 3.5.1 (2018-07-02)
- R version (short form): 3.5.1
- **rmarkdown** package version: 1.10
- **tufte** package version: 0.4
- File version: 1.0.1
- Author Profile: ®γσ, Eng Lian Hu
- GitHub: Source Code
- Additional session information

[1] “2018-08-30 04:52:07 JST”

Category	session_info
version	R version 3.5.1 (2018-07-02)
os	Windows 10 x64
system	x86_64, mingw32
ui	RTerm
language	en
collate	Japanese_Japan.932
tz	Asia/Tokyo
date	2018-08-30

Category	Sys.info
sysname	Windows
release	10 x64
version	build 16299
nodename	RSTUDIO-SCIBROK
machine	x86-64
login	scibr
user	scibr
effective_user	scibr

7.2 Reference

1. Binary.com Interview Q1(Alternate link)

---

1. reference paper 1↩
2. Since the datasets contain few observation within a minute and also some data time period empty more than 30 minutes. Test all time period but eventually error… ↩
3. Only get 2013-01-01 to 2017-05-15 in csv format data.↩
4. Note: The CSV file is using ; (semicolon) as a separator and , (comma) as a decimal separator.↩