

Model 1

2023-12-30

Data Preprocessing

Load the required packages

```
library(quantmod)
library(tidyverse)
library(Hmisc)
library(moments)
library(reshape2)
library(fGarch)
```

Define the stock symbol and date range

```
tickers <- c("AAPL", "AMGN", "AXP", "BA", "CAT", "CRM", "CSCO", "CVX", "DIS", "DOW",
             "GS", "HD", "HON", "IBM", "INTC", "JNJ", "JPM", "KO", "MCD", "MMM",
             "MRK", "MSFT", "NKE", "PG", "TRV", "UNH", "V", "VZ", "WBA", "WMT")
start_date <- as.Date("2000-01-01")
end_date <- as.Date("2020-12-31")
```

Fetch the stock prices

```
closing_prices <- lapply(tickers, function(ticker) {
  getSymbols(ticker, src = 'yahoo', from = start_date, to = end_date, auto.assign = FALSE)[,6]
})

closing_prices <- as_tibble(do.call(cbind, closing_prices))

date <- index(getSymbols("AAPL", src = 'yahoo', from = start_date, to = end_date, auto.assign = FALSE))
closing_prices <- cbind(date, closing_prices)

# Remove columns with NA & clean column names
closing_prices <- closing_prices[ , colSums(is.na(closing_prices))==0]

names(closing_prices)[-1] <- substr(names(closing_prices)[-1], 1, nchar(names(closing_prices)[-1]) - 9)

# Calculate daily returns
daily_returns <- closing_prices %>%
  mutate_at(vars(-1), ~log(.) - log(lag(.))) %>%
```

```
na.omit()

days <- nrow(daily_returns)

daily_returns_long <- pivot_longer(daily_returns, cols = -1, names_to = "ticker", values_to = "returns")
```

Helper functions

```
# function to fetch the correlation matrix given start and end dates

# function to get prediction of standard deviations given start and end dates
```

Fit Individual $Garch(1,1)$ Models

```
fit <- garchFit(data = daily_returns$AAPL)

##
## Series Initialization:
## ARMA Model:          arma
## Formula Mean:        ~ arma(0, 0)
## GARCH Model:         garch
## Formula Variance:    ~ garch(1, 1)
## ARMA Order:          0 0
## Max ARMA Order:      0
## GARCH Order:         1 1
## Max GARCH Order:     1
## Maximum Order:       1
## Conditional Dist:    norm
## h.start:             2
## llh.start:           1
## Length of Series:    5282
## Recursion Init:      mci
## Series Scale:        0.02656626
##
## Parameter Initialization:
## Initial Parameters:   $params
## Limits of Transformations: $U, $V
## Which Parameters are Fixed? $includes
## Parameter Matrix:
##      mu      U      V  params includes
##      --      -      -  --
## mu    -0.35946004  0.35946 0.035946    TRUE
## omega 0.00000100 100.00000 0.100000    TRUE
## alpha1 0.00000001  1.00000 0.100000    TRUE
## gamma1 -0.99999999  1.00000 0.100000    FALSE
## beta1  0.00000001  1.00000 0.800000    TRUE
## delta  0.00000000  2.00000 2.000000    FALSE
## skew   0.10000000 10.00000 1.000000    FALSE
```

```

##      shape  1.00000000  10.00000  4.000000  FALSE
## Index List of Parameters to be Optimized:
##      mu  omega alpha1  beta1
##      1    2    3      5
## Persistence:                0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
## 0:      6713.5473: 0.0359460 0.100000 0.100000 0.800000
## 1:      6632.1828: 0.0359479 0.0732812 0.103908 0.788065
## 2:      6573.7514: 0.0359519 0.0641144 0.129782 0.798933
## 3:      6572.1709: 0.0359564 0.0355571 0.136201 0.795078
## 4:      6529.5715: 0.0359578 0.0457900 0.143315 0.802989
## 5:      6512.7045: 0.0359631 0.0354150 0.151679 0.809336
## 6:      6507.9233: 0.0359665 0.0359301 0.156529 0.815457
## 7:      6503.9257: 0.0359720 0.0283750 0.155942 0.817411
## 8:      6499.0186: 0.0359821 0.0291553 0.155925 0.825194
## 9:      6495.7412: 0.0359943 0.0238198 0.153741 0.830477
## 10:     6493.0355: 0.0360110 0.0236220 0.152025 0.838097
## 11:     6490.7998: 0.0360359 0.0196199 0.148437 0.843745
## 12:     6489.1965: 0.0360838 0.0189568 0.145261 0.850742
## 13:     6488.1321: 0.0361962 0.0162152 0.141167 0.855960
## 14:     6487.3456: 0.0364337 0.0165532 0.138712 0.859343
## 15:     6483.3211: 0.0407999 0.0126067 0.112087 0.881903
## 16:     6483.0925: 0.0453403 0.0163166 0.114243 0.878101
## 17:     6480.6453: 0.0476109 0.0140269 0.115356 0.878769
## 18:     6479.8267: 0.0522707 0.0126181 0.118028 0.880869
## 19:     6479.0113: 0.0569306 0.0123453 0.119161 0.877352
## 20:     6477.7100: 0.0615907 0.0139863 0.121511 0.875343
## 21:     6477.1499: 0.0662511 0.0139140 0.123329 0.872847
## 22:     6476.8101: 0.0709128 0.0144260 0.123666 0.872657
## 23:     6476.5025: 0.0728846 0.0133773 0.116399 0.879285
## 24:     6476.4574: 0.0748877 0.0134786 0.117161 0.878251
## 25:     6476.4563: 0.0752965 0.0135063 0.117233 0.878115
## 26:     6476.4563: 0.0752977 0.0135033 0.117237 0.878117
## 27:     6476.4563: 0.0752960 0.0135039 0.117236 0.878116
##
## Final Estimate of the Negative LLH:
## LLH:  -12687.24    norm LLH:  -2.401976
##      mu      omega      alpha1      beta1
## 2.000334e-03 9.530592e-06 1.172363e-01 8.781164e-01
##
## R-optimhess Difference Approximated Hessian Matrix:
##      mu      omega      alpha1      beta1
## mu      -15555360.68 -2.901099e+08    -13027.6    -78771.09
## omega  -290109923.16 -1.998808e+12 -342862409.5 -554947724.08
## alpha1    -13027.60 -3.428624e+08    -123065.8    -156723.87
## beta1     -78771.09 -5.549477e+08    -156723.9    -223024.68
## attr("time")
## Time difference of 0.131686 secs

```

```
##  
## --- END OF TRACE ---  
##  
##  
## Time to Estimate Parameters:  
## Time difference of 0.6115191 secs
```

```
pred <- predict(fit, n.ahead = 10)
```