Date:

Apr 30, 2022

Course:

Financial Time Series

Lecturer:

Xiaodong Lin

Dow Jones Stocks Trading Strategy Based on Time Series Theory

Rutgers Business School
Newark and New Brunswick

Junlong Wu

Liuqing Yang

Jianqi Zhu

Balarama V Nyayapathi

Contents

1. Researching Object

As one of the most influential stock market indexes, Dow Jones Industrial Average index contains 30 stocks of large, publicly-owned blue-chip companies trading on the New York Stock Exchange (NYSE) and Nasdaq. We take the component stocks, though some of which are not DJI's member at the beginning data point, as candidates for our arbitrage strategy:

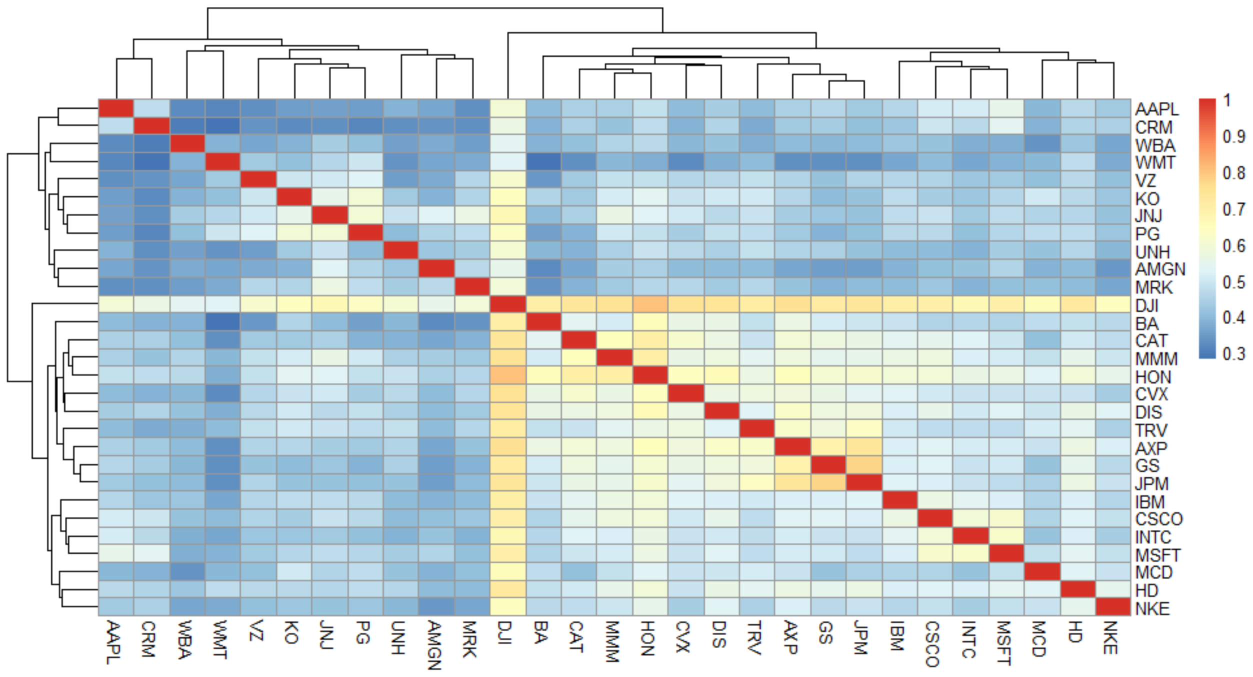
Company	Ticker
3M	MMM
American Express	AXP
Amgen	AMGN
Apple Inc.	AAPL
Boeing	BA
Caterpillar	CAT
Chevron	CVX
Cisco Systems	CSCO
The Coca-Cola Company	KO
Goldman Sachs	GS
The Home Depot	HD
Honeywell	HON
IBM	IBM
Intel	INTC
Johnson & Johnson	JNJ
JPMorgan Chase	JPM
McDonald's	MCD
Merck & Co.	MRK
Microsoft	MSFT
NIKE	NKE
Proctor & Gamble	PG
Salesforce	CRM
The Travelers Companies	TRV
UnitedHealth Group	UNH
Verizon	VZ
Walmart	WMT
Walgreens Boots Alliance	WBA
The Walt Disney Company	DIS

We use the logarithm return rate of the 28 stocks above as the sample. The time period of sample data is from 2007.01.01-2022.4.22, totalling 3854 observations for each single stock. For each

stock, the first 3000 observations will be treated as the training set, and the latter 854 observations will be used as the testing sample.
Notice that Dow Inc(DOW) and Visa(V) are excluded from the other 28 DJI stocks, for they were not publicly traded on board at the beginning of 2007.

The data source is Yahoo Finance via the "Quantmod" package in R-studio.

To apply a stable ARIMA or GARCH on as many stocks as possible. We considered industry property to divide stocks into different groups. Here is a heatmap based on Pearson correlation coefficients of log return time series.



Because all constituents compose DJI index, every stock has obvious correlation with DJI index. However, they may not have a significant relationship with each other. We find that four stocks—-JPM, GS, AXP, TRV are highly correlated. The first three are american big banks and the last one is a big insurance company. The Pearson coefficient is 0.6 at least, and it is reasonable to set ARIMA or GARCH models with fixed parameters for these four stocks in the finance industry.

2. Trading Strategy

We try to construct two kinds of time series models to predict the trend of the stock price change. The first model is ARIMA, and the second model is ARMA+GARCH which takes volatility into account.

Regarding the property of the prediction, we separately design two strategies for ARIMA and ARMA-GARCH models.

For ARIMA: As 1-step predicted log return is positive, buy and hold the stock. No sell-short trades in this strategy. And it is a momentum strategy.

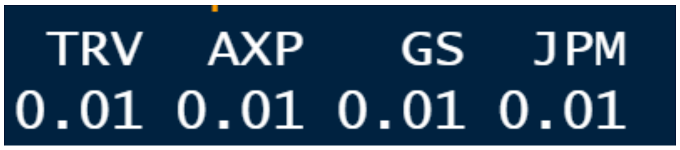
For ARMA-GARCH: Every day We sort the cross-sectional data of 1-step prediction on each stock's daily log return. Then pick up top 2 oversold stocks with the lowest predicted returns to buy and hold. We allocate the same amount of money on every stock and then get a mean-reversion strategy.

Those stocks being forecasted to sharply drop are highly likely to rebound in the next day. So it is logical to buy oversold stocks. The reason why we don't consider selling short is that in day frequency U.S market seldom continuously goes down and sell-short signal is not as accurate as buy-long signal.

3. Model Fitting and Evaluation-ARIMA Model

3.1. Stationarity

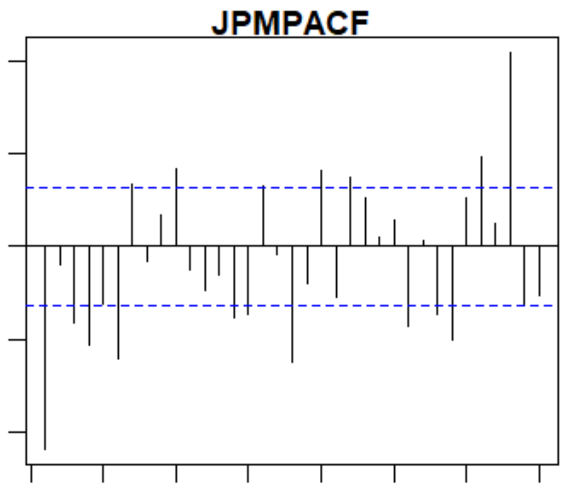
We test JPM, GS, AXP, TRV time series stationarity through ADF test. P-values of all stocks are less than 0.01 and reject the null hypothesis that the time series is not stationary. Hence, differencing is not necessary.

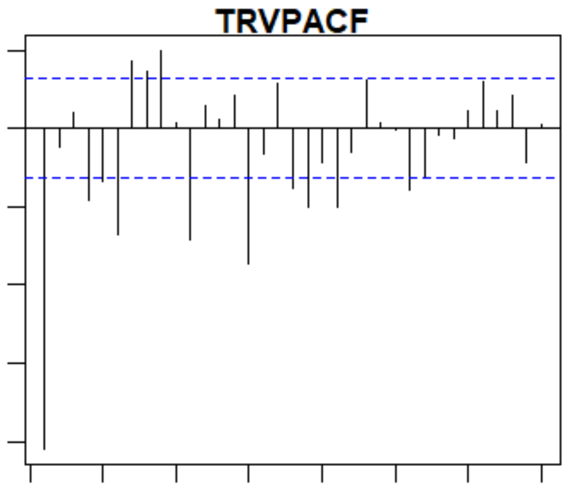


```
 TRV   AXP    GS   JPM
0.01  0.01  0.01  0.01
```

3.2. Determine the Order of lags

Firstly, we select some stocks as a sample to observe their autocorrelations and partial autocorrelations. Like JPM, both ACF and PACF are cut-off, so it is proper to use the ARMA model.

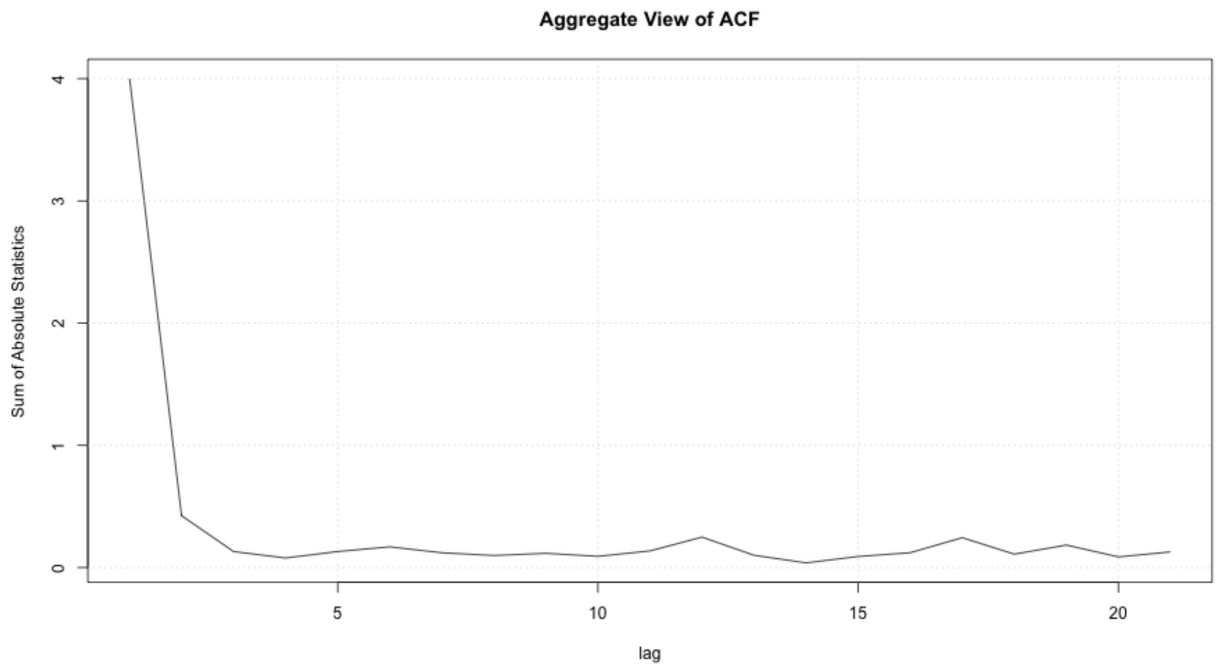


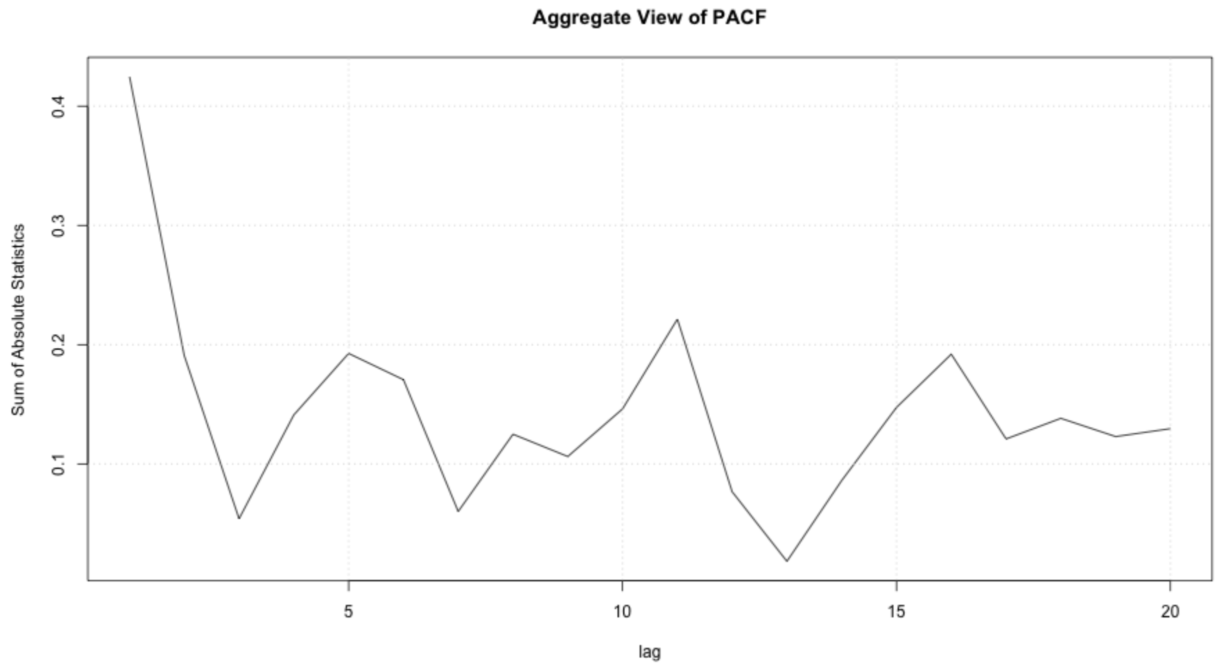


To roughly determine the order of (p,0,q). We draw the following two aggregate view plots. For each lag, we sum up 4 stocks' ACF or PACF values and then plot the curve. We can see that AR lag should be equal to 1 or 2 and MA lag should be less than 3.

$$f(j) = \sum_{i=1}^{4} \gamma_{i,j}$$

where $\gamma_{i,j}$ is the statistic of stock i with lag j in its ACF/PACF.

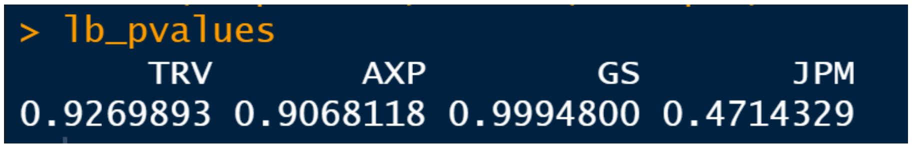




We try to observe the significance of each lag in each ARIMA(p,0,q) and take AIC into account. Finally we define the ARIMA order as (1,0,1).

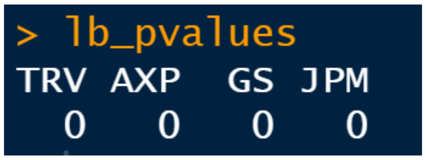
$$X_t = \phi_0 + \phi_1 X_{t-1} + Z_t + \theta_1 Z_{t-1} \quad Z_t \sim N(0, \sigma^2)$$

Then we run Ljung-Box test on each stock's ARIMA(1,0,1) to test whether the residual has autocorrelations. The result is acceptable: No stock rejects the null hypothesis that the time series has no autocorrelations, which means the ARIMA(1,0,1) is adequate enough to explain time series fluctuation.



```
> lb_pvalues
      TRV       AXP        GS       JPM 
0.9269893 0.9068118 0.9994800 0.4714329 
```

In addition, we find that the time-series of squared residuals rejects Ljung-Box test. This is the most important reason why we want to try GARCH model in the next step.



```
> lb_pvalues
TRV AXP  GS JPM 
  0   0   0   0 
```

3.3. Backtest in Testing Sample

After the model determination in the training sample, we apply our model on the testing sample. We fit updated ARIMA(1,0,1) day by day and roll on the last 854 observations. Each time the model can have a new 1-step prediction, and we buy those stocks with positive predicted log returns.

Refer to 6. Strategy Realization

4. Model Fitting and Evaluation-ARIMA+GARCH Model

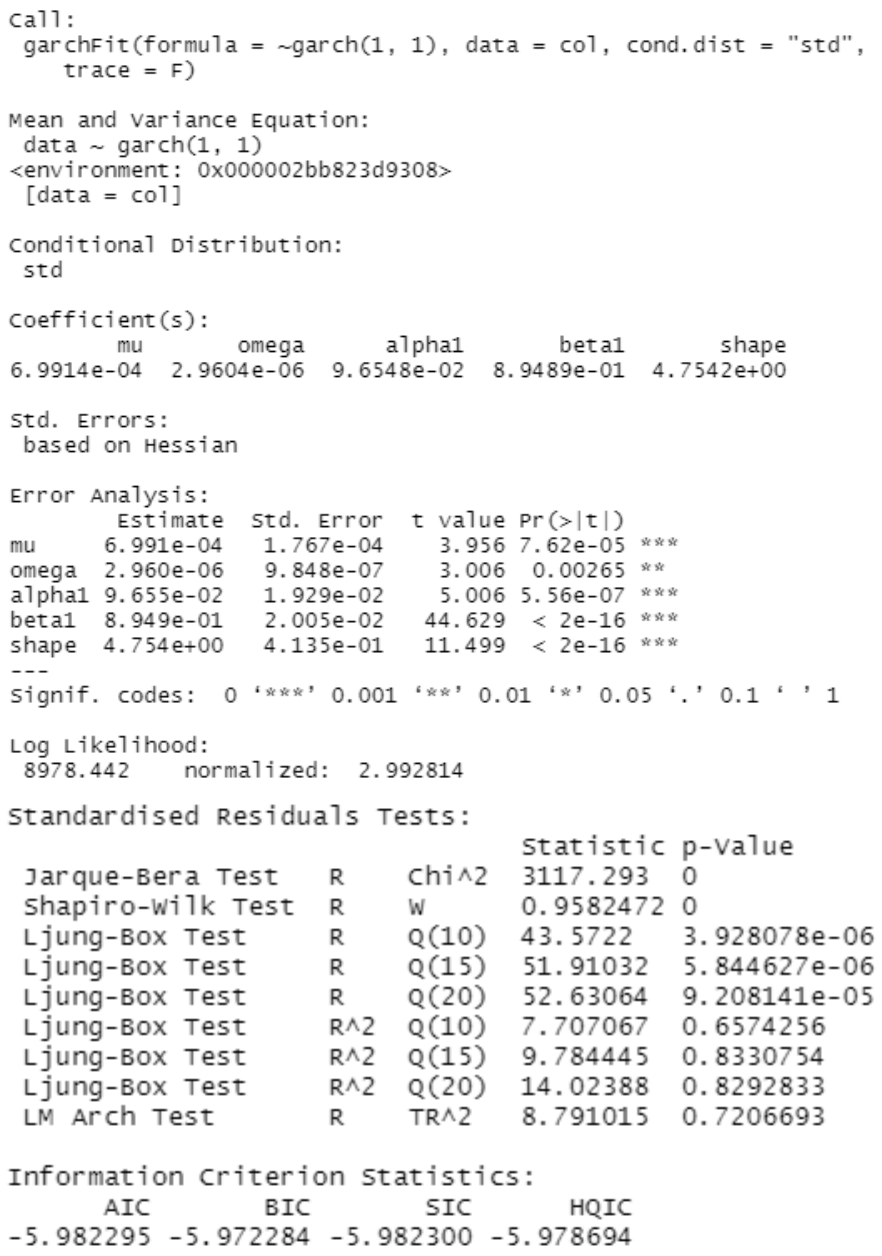
4.1 Fitting a GARCH model

As we can easily detect ARCH effect in the residuals of most ordinary ARMA models via Ljung-Box test , there exists significant serial correlation in model residuals. Therefore, it is reasonable to consider a Generalized Auto-Regression Conditional Heteroskedasticity(GARCH) model for the stock returns.

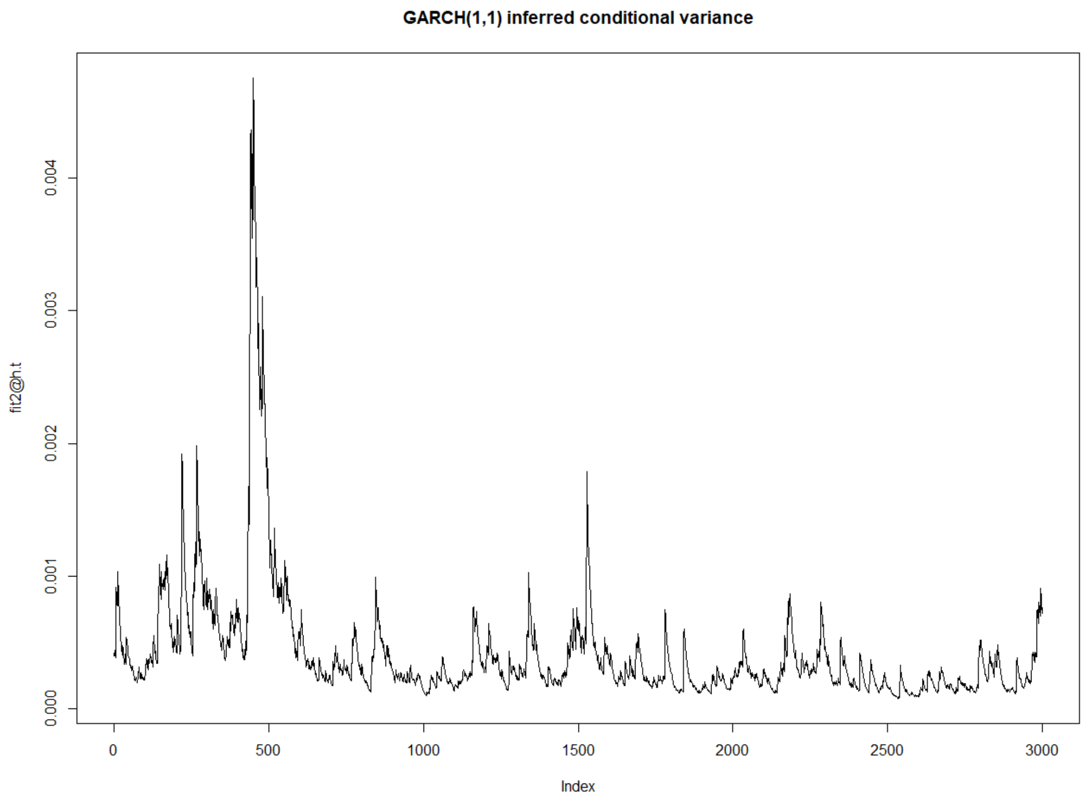
We use the TRV as an example to illustrate the process of evaluating the effectiveness of the GARCH model below:

Firstly, we fit an ordinary GARCH(1,1) model with the log returns

$$r_t = \mu + a_t$$
$$a_t = \sigma_t \epsilon_t$$
$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$



```
Call:
 garchFit(formula = ~garch(1, 1), data = col, cond.dist = "std",
    trace = F)

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x000002bb823d9308>
 [data = col]

Conditional Distribution:
 std

Coefficient(s):
        mu       omega      alpha1       beta1       shape
6.9914e-04  2.9604e-06  9.6548e-02  8.9489e-01  4.7542e+00

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     6.991e-04   1.767e-04    3.956 7.62e-05 ***
omega  2.960e-06   9.848e-07    3.006  0.00265 **
alpha1 9.655e-02   1.929e-02    5.006 5.56e-07 ***
beta1  8.949e-01   2.005e-02   44.629  < 2e-16 ***
shape  4.754e+00   4.135e-01   11.499  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 8978.442    normalized:  2.992814

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  3117.293  0
 Shapiro-Wilk Test  R    W      0.9582472 0
 Ljung-Box Test     R    Q(10)  43.5722   3.928078e-06
 Ljung-Box Test     R    Q(15)  51.91032  5.844627e-06
 Ljung-Box Test     R    Q(20)  52.63064  9.208141e-05
 Ljung-Box Test     R^2  Q(10)  7.707067  0.6574256
 Ljung-Box Test     R^2  Q(15)  9.784445  0.8330754
 Ljung-Box Test     R^2  Q(20)  14.02388  0.8292833
 LM Arch Test       R    TR^2   8.791015  0.7206693

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.982295 -5.972284 -5.982300 -5.978694
```



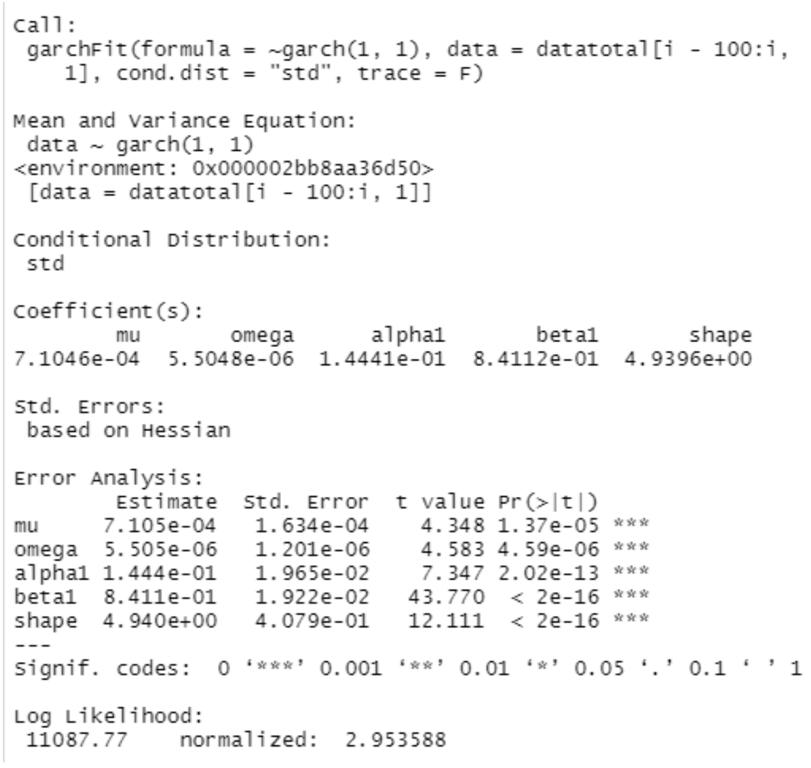
The estimated parameters are basically all significant for every stock. In the output of the Ljung-Box test, the Q-statistics of the residuals of the model are mostly insignificant, which indicates there is no serial correlation in residuals. Next, we try to find an improvement for the GARCH(1,1) model, and we will add ARMA terms in order to enhance the estimation.

4.2 Adding ARMA terms to the GARCH model

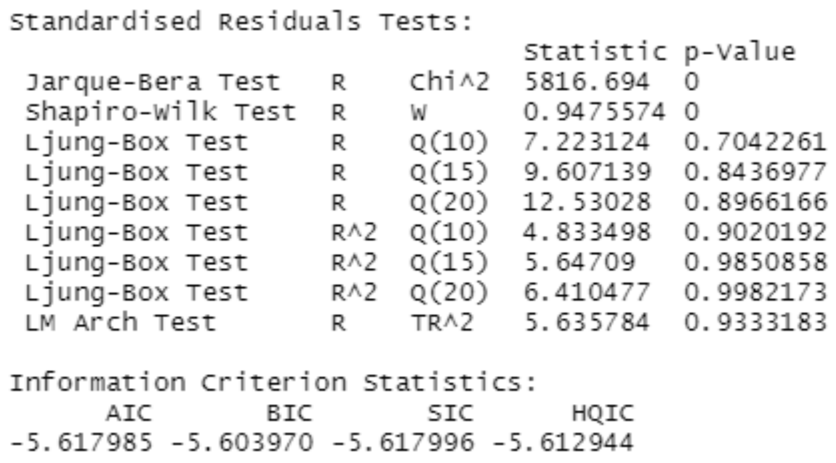
Based on experience and empirical testing, we tune the parameters of the ARMA+GARCH model to achieve an ideal model based on the criterion of: 1) Efficiency of Ljung-Box test; 2) Significance of estimated coefficients; 3) AIC.

Aiming to achieve an adequate model, finally we choose an ARMA(1,1)+GARCH(1,1) model. It will provide insignificant Q-statistics for the Ljung-Box test, both residuals and the square of residuals have no serial correlation.

$$\begin{aligned} r_t &= \mu + \phi_1 r_{t-1} + a_t + \theta_1 a_{t-1} \\ a_t &= \sigma_t \epsilon_t \\ \sigma_t^2 &= \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \end{aligned}$$

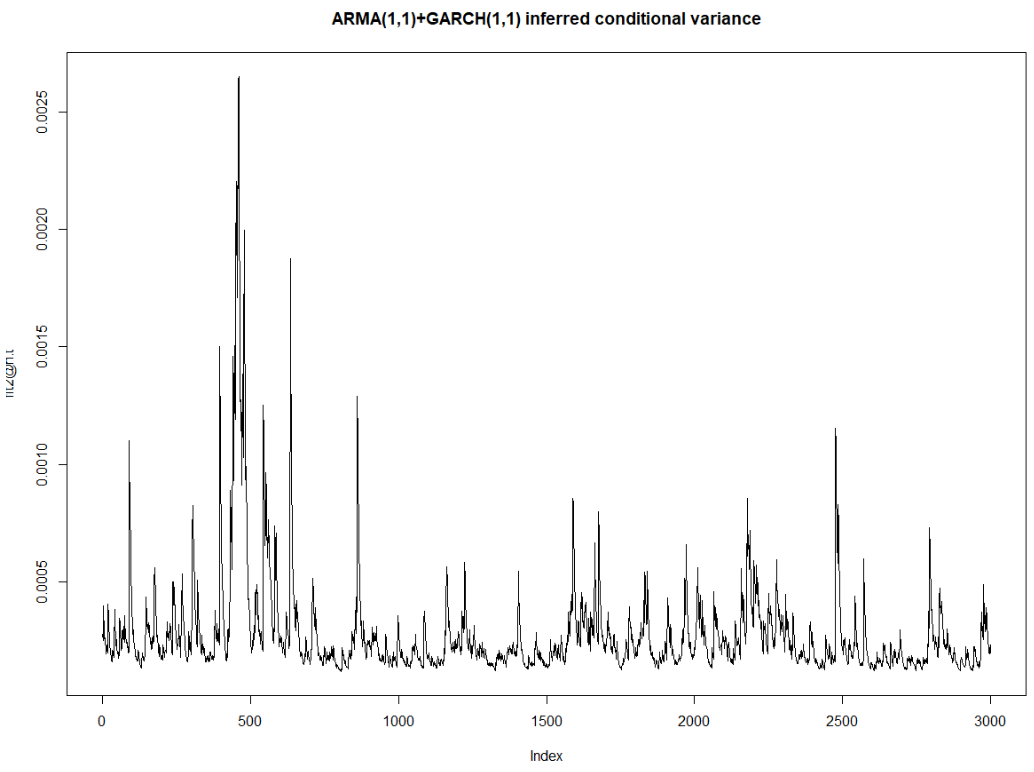


```
Call:
 garchFit(formula = ~garch(1, 1), data = datatotal[i - 100:i,
    1], cond.dist = "std", trace = F)

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x000002bb8aa36d50>
 [data = datatotal[i - 100:i, 1]]

Conditional Distribution:
 std

Coefficient(s):
        mu       omega      alpha1       beta1       shape
7.1046e-04  5.5048e-06  1.4441e-01  8.4112e-01  4.9396e+00

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     7.105e-04   1.634e-04    4.348 1.37e-05 ***
omega  5.505e-06   1.201e-06    4.583 4.59e-06 ***
alpha1 1.444e-01   1.965e-02    7.347 2.02e-13 ***
beta1  8.411e-01   1.922e-02   43.770  < 2e-16 ***
shape  4.940e+00   4.079e-01   12.111  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 11087.77    normalized:  2.953588
```



```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  5816.694  0
 Shapiro-Wilk Test  R    W      0.9475574 0
 Ljung-Box Test     R    Q(10)  7.223124  0.7042261
 Ljung-Box Test     R    Q(15)  9.607139  0.8436977
 Ljung-Box Test     R    Q(20)  12.53028  0.8966166
 Ljung-Box Test     R^2  Q(10)  4.833498  0.9020192
 Ljung-Box Test     R^2  Q(15)  5.64709   0.9850858
 Ljung-Box Test     R^2  Q(20)  6.410477  0.9982173
 LM Arch Test       R    TR^2   5.635784  0.9333183

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.617985 -5.603970 -5.617996 -5.612944
```

For models with lower AR or MA lags, the residuals will have serial correlation; For higher AR or MA lags, as well as GARCH lags, the estimated coefficients turn out to be invalid synchronously in every stock. Next, we choose an ARMA(1,1)+GARCH(1,1) model to fit the stock returns.



It will produce better Ljung-Box test statistics and smaller AIC than the GARCH(1, 1) model. Consider from an aggregate view, such a model is suitable for the entity of all stocks, because nearly in all cases the parameters are valid. We consider it reasonable to fit the ARMA(1,1)+GARCH(1,1) model for all stocks.

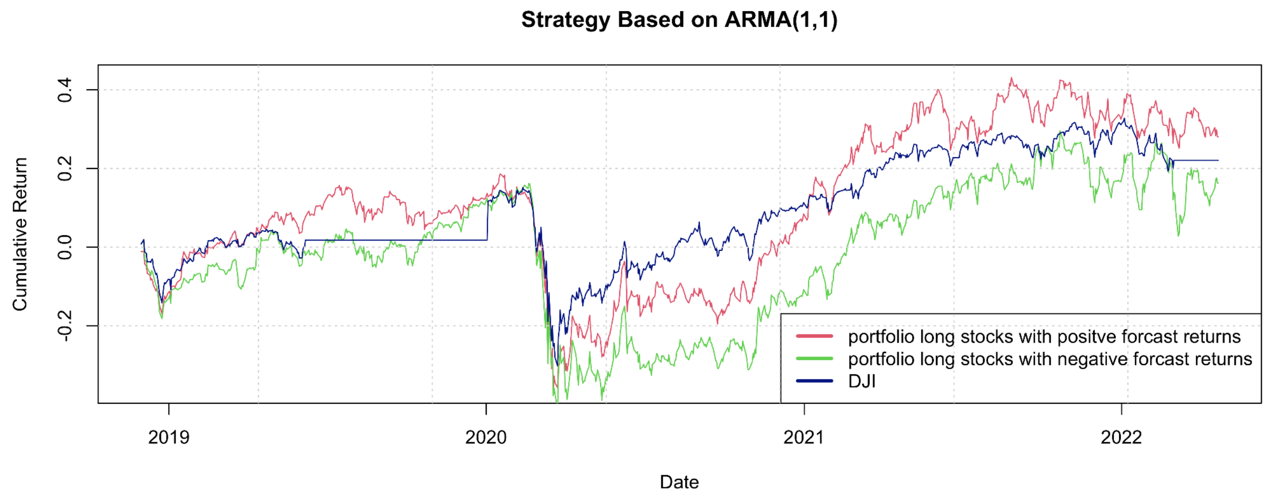
4.3 One-Step Prediction

Next, we use the ARMA(1,1)+GARCH(1,1) model and the data sample from both training sample and testing sample to generate a one-step prediction model with a moving window of 500 observations.
Firstly, we use the last 500 data point from the training sample to fit an ARMA(1,1)+GARCH(1,1) model to generate the first prediction; then, the second prediction will be generated by the last 499 data point and the first 1 data from the testing sample, and so on until the end.

5. Strategy Realization

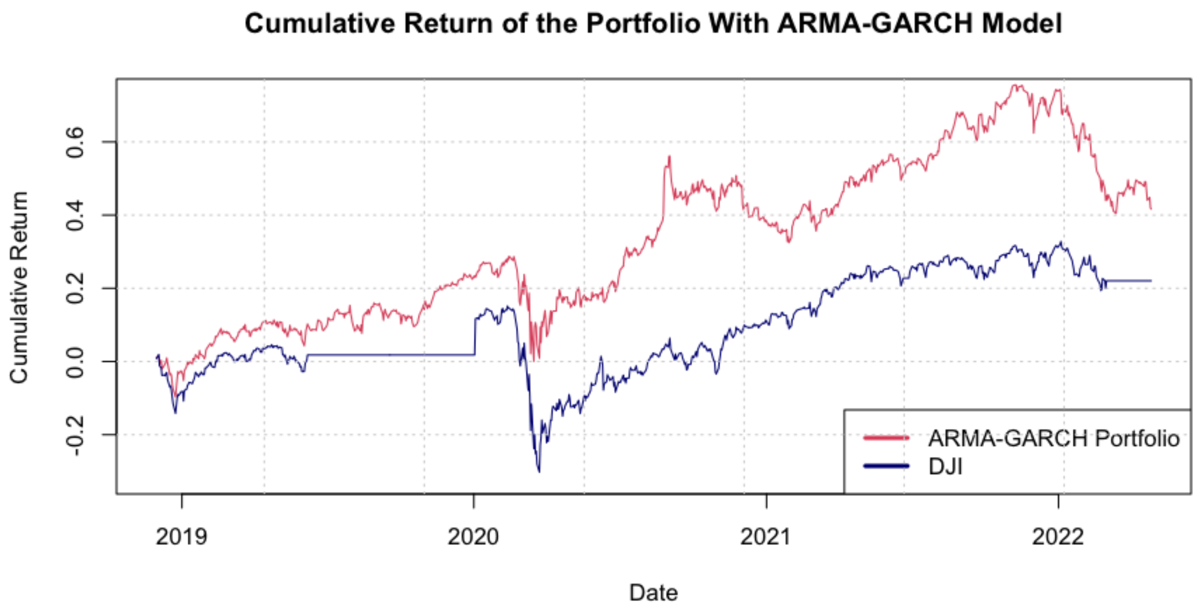
5.1 ARIMA(1, 0, 1) Performance

The curve of DJI benchmark is flat at some time, because R package-Quantmod can't successfully request all data from Yahoo. In total, ARIMA(1,0,1) is effective to select better-performance stocks. Although it can't totally beat the DJI index, buying top 2 stocks can significantly beat buying the last 2 stocks.



5.2 ARMA(1,1)+GARCH(1,1) Performance

The return strategy realized by the ARMA(1,1)+GARCH(1,1) model has a positive return in a mass. It significantly performs better than DJI as the benchmark.



6. Risk Management Measure

	Annualized Return	**Annualized Standard deviation**	**Annualized Sharpe Ratio**	**Daily VaR at 5%**
DJI	**6.019%**	**27.823%**	**21.309%**	**-1.874%**
ARIMA(1,0,1) (positive)	**7.518%**	**40.478%**	**18.353%**	**-2.707%**
ARMA-GARCH	**10.755%**	**26.547%**	**40.179%**	**-2.050%**

Instead of the initial ARIMA(0,1,14) model (in the presentation), the ARIMA(1,0,1) model after improvement helps our strategy generate a satisfactory return. By longing the stocks with a positive forecasting return, the portfolio outperforms DJI about 1.5% annually. The strategy based on the later ARMA-GARCH has better performance than both investing only on DJI and the strategy based on the ARIMA(1,0,1) model with a significantly higher annualized return and Sharpe ratio.

7. Conclusion

We try to train a stable ARIMA and ARMA-GARCH in the training sample, and use models with fixed sets of parameters to construct quant strategy. From the performance on the testing sample, we can see that both models under different strategies can yield good profits. Although ARIMA(1,0,1) doesn't beat the benchmark, DJI, the classification of good and bad stocks in the finance industry is obvious. In this momentum strategy, stocks with higher 1-step forecasting value actually perform better than lower ones. And ARMA(1,1)-GARCH(1,1) verifies mean-reversion phenomenon, and beats DJI.

Appendix: Codes Lists - Group 2

April 30, 2022

1 Code List 1: Visualization, Backtesting, Order Determination

```
library(zoo)
library(tseries)
library(forecast)
library(quantmod)

symbols_list <- c("MMM","AXP", "AMGN","AAPL","BA","CAT","CVX","CSCO","KO","DIS","DOW",
"GS","HD","HON","IBM",
    "INTC","JNJ", "JPM","MCD", "MRK","MSFT","NKE","PG","CRM","TRV","UNH","VZ","V",
    "WBA","WMT")

getSymbols(symbols_list)

getSymbols("DJI",src='yahoo')
seri <- dailyReturn(DJI,type='log')
for (symbol in symbols_list){
  sub_seri <- dailyReturn(get(symbol),type='log')
  seri <- merge(seri,sub_seri)
}



col_names=c('DJI',"MMM","AXP", "AMGN","AAPL","BA","CAT","CVX","CSCO","KO","DIS","DOW",
"GS","HD","HON","IBM","INTC","JNJ", "JPM","MCD", "MRK","MSFT","NKE","PG","CRM","TRV",
"UNH","VZ","V", "WBA", "WMT")
data=data.frame(seri)
names(data)=col_names

data[is.na(data)] <- 0
data <- subset (data, select = -c(V,DOW))
#data = na.locf(data,na.rm=FALSE,fromLast=TRUE)




#training_data <- data[1:3000,]
#testing_data <- data[3001:nrow(data),]
training_data <- read.csv('insample.csv')
row.names(training_data) <- training_data$X
training_data$X <- NULL
```

```
acf_ord <- list()
pacf_ord <- list()
symbols <- names(training_data)
for (ticker in symbols){
  ts <- training_data[ticker][,1]
  ts_acf <- acf(ts,plot=FALSE,lag.max = 40)
  ts_pacf <- pacf(ts,plot=FALSE,lag.max = 40)

  lag <- 35
  crit <- 1.96/sqrt(dim(training_data)[1]-lag)

  is_acf_sig <- (abs(ts_acf$acf)>crit)
  is_pacf_sig <- (abs(ts_pacf$acf)>crit)

  sig_ord_acf <- c()
  sig_ord_pacf <- c()

  for (ord in index(is_acf_sig)){
    #print(ticker)
    #print(is_acf_sig)
    if (is_acf_sig[ord]){
      sig_ord_acf <- c(sig_ord_acf,ord)
    }
  }
  for (ord in index(is_pacf_sig)){
    if (is_pacf_sig[ord]){
      sig_ord_pacf <- c(sig_ord_pacf,ord)
    }
  }
  acf_ord <- c(acf_ord,list(sig_ord_acf))
  pacf_ord <- c(pacf_ord,list(sig_ord_pacf))
}
#acf_ord
#pacf_ord
names(acf_ord) <- symbols
names(pacf_ord) <- symbols

count_acf <- c()
for (i in 1:length(acf_ord)){
  count_acf <- c(count_acf,unlist(acf_ord[i],use.names = FALSE))
}
png(file="Count of Significant Lags in ACF (with differencing).png",
width=900,height=510)

hist(count_acf,breaks=0:(length(tabulate(count_acf))+1),labels = TRUE,
xlab="lag",ylab="Frequency",main="Count of Significant Lags in ACF (with differencing)")
dev.off


count_pacf <- c()
for (i in 1:length(pacf_ord)){
  count_pacf <- c(count_pacf,unlist(pacf_ord[i],use.names = FALSE))
}
png(file="Count of Significant Lags in PACF (with differencing).png",
width=900,height=510)
```

```
hist(count_pacf, breaks=0:(length(tabulate(count_pacf))+1), labels = TRUE,
xlab="lag", ylab="Frequency", main="Count of Significant Lags in PACF (with differencing)"
dev.off

symbols <- names(pred)

first_time <- TRUE
for (ticker in symbols){
  ts <- training_data[ticker][,1]
  ts_acf <- acf(ts, plot=FALSE, lag.max = 20)
  ts_pacf <- pacf(ts, plot=FALSE, lag.max = 20)

  if (first_time){
    acf_cum <- abs(ts_acf$acf)
    pacf_cum <- abs(ts_pacf$acf)
    first_time <- FALSE
  }else{
    acf_cum <- acf_cum + abs(ts_acf$acf)
    pacf_cum <- pacf_cum + abs(ts_pacf$acf)
  }
}


png(file="Aggregate View of ACF (without differencing).png", width=900, height=510)
plot(1:length(c(acf_cum)), c(acf_cum), type='l', xlab = 'lag',
ylab = 'Sum of Absolute Statistics', main='Aggregate View of ACF',)
grid()
dev.off


png(file="Aggregate View of PACF for (without differencing).png", width=900, height=510)
plot(1:length(c(pacf_cum)), c(pacf_cum), type='l', xlab = 'lag',
ylab = 'Sum of Absolute Statistics', main='Aggregate View of PACF')
grid()
dev.off


plot(1:length(pacf_cum), c(pacf_cum))
plot(1:length(acf_cum), c(acf_cum))


testing_data <- read.csv('testing_data.csv', row.names = 1)

garch <- read.csv('new.csv', row.names = 1)
garch[is.na(garch)] <-  0

arim <- read.csv('ARIMA.csv', row.names = 1)
arim[is.na(arim)] <-  0

back_testing_garch <- data.frame(rep(NaN, length(row.names(testing_data))))
back_testing_arima <- data.frame(rep(NaN, length(row.names(testing_data))))

row.names(back_testing_garch) <- row.names(testing_data)
row.names(back_testing_arima) <- row.names(testing_data)
```

```
colnames(back_testing_garch) <- c('Return')
colnames(back_testing_arima) <- c('Return')

num_sto <- 3
for (date in row.names(back_testing_arima)){

  cur_row_g <- garch[date,]
  order_vec_g <- rank(cur_row_g)

  long_stoc_g <- names(cur_row_g[which(order_vec_g>=(30-num_sto))])
  long_g <- testing_data[date,long_stoc_g]
  long_ret_g <- mean(unlist(long_g))

  short_stoc_g <- names(cur_row_g[which(order_vec_g<=num_sto)])
  short_g <- testing_data[date,short_stoc_g]
  short_ret_g <- mean(unlist(short_g))

  back_testing_garch[date,1] <-  short_ret_g

  cur_row_a <- arim[date,]
  order_vec_a <- rank(cur_row_a)

  long_stoc_a <- names(cur_row_a[which(order_vec_a>=(30-num_sto))])
  long_a <- testing_data[date,long_stoc_a]
  long_ret_a <- mean(unlist(long_a))

  short_stoc_a <- names(cur_row_a[which(order_vec_a<=num_sto)])
  short_a <- testing_data[date,short_stoc_a]
  short_ret_a <- mean(unlist(short_a))

  back_testing_arima[date,1] <-  short_ret_a
}


png(file="Cumulative␣Return␣of␣the␣Portfolio␣With␣ARMA-GARCH␣Model.png",
width=890,height=480,res=100)
plot(as.Date(row.names(back_testing_garch)),cumprod(1+back_testing_garch[,1])-1,
type='l',ylim=c(-0.32,0.73),xlab='Date',ylab='Cumulative␣Return',
main='Cumulative␣Return␣of␣the␣Portfolio␣With␣ARMA-GARCH␣Model',col=2)
lines(as.Date(row.names(back_testing_garch)),unlist(cumprod(1+testing_data['DJI'])-1),
col='navy')
legend("bottomright",lwd=3, legend = c("ARMA-GARCH␣Portfolio", "DJI"),
col = c(2, "navy"))
grid()
dev.off

png(file="Cumulative␣Return␣of␣the␣Portfolio␣With␣ARIMA␣Model.png",
width=890,height=480,res=100)
plot(as.Date(row.names(back_testing_arima)),cumprod(1+back_testing_arima[,1])-1,
type='l',ylim=c(-0.32,0.73),xlab='Date',ylab='Cumulative␣Return',main='Cumulative␣Return
lines(as.Date(row.names(back_testing_garch)),unlist(cumprod(1+testing_data['DJI'])-1),
col='navy')
legend("bottomright",lwd=3, legend = c("ARIMA␣Portfolio", "DJI"), col = c(2, "navy"))
grid()
dev.off
```

```
png(file="Summary␣Three.png",
width=890,height=480,res=100)
plot(as.Date(row.names(back_testing_arima)),cumprod(1+back_testing_arima[,1])-1,
type='l',ylim=c(-0.32,0.73),xlab='Date',ylab='Cumulative␣Return',
main='Cumulative␣Return␣of␣the␣Portfolio␣With␣ARIMA␣Model',col=3)
line(as.Date(row.names(back_testing_garch)),cumprod(1+back_testing_garch[,1])-1)
lines(as.Date(row.names(back_testing_garch)),unlist(cumprod(1+testing_data['DJI'])-1),
col='navy')
dev.off
```

2 Code List 2: ARMA-GARCH

```
library(quantmod)
library(tseries)
library(fGarch)
library(forecast)
library(dplyr)
library(graphics)
# get data
getSymbols("DJI")
seri=periodReturn(DJI,type='log',period='daily')
#stocks=c("MMM","AXP", "AMGN","AAPL","BA","CAT","CVX","CSCO","KO","DIS","GS","HD",
"HON","IBM","INTC","JNJ", "JPM","MCD", "MRK","MSFT","NKE","PG","CRM","TRV","UNH",
"VZ", "WBA", "WMT")
stocks=c("DIS","TRV","AXP","GS", "JPM")
getSymbols(stocks)

for (stk in stocks){
  sub_seri=periodReturn(get(stk),type='log',period='daily')
  seri=merge(seri,sub_seri)
}
col_names=stocks
data=data.frame(seri)
names(data)=col_names
data[is.na(data)] = 0

# Use AAPL, AMGN build time series model
acf(data$AMGN)
pacf(data$AMGN)
acf(data$AMGN[1:100])
adf.test(data$AMGN)
data1=data[1:3000,]
data2=data[3001:3854,]
sumaic=0
plot
fit1=arima(data$AMGN,order=c(1,1,0))
fit1
summary(fit1)
Box.test(fit1$residuals)
fit2=garchFit(~arma(1,1)+garch(1,1),data=data1$AAPL,trace=F,cond.dist = "std")
summary(fit2)
plot(fit2@h.t,type='l',main="ARMA(1,1)+GARCH(1,1)␣inferred␣conditional␣variance")
# ARMA(1,1)+ GARCH (1,1)
p1=predict(fit2,30)
```

```
#record=data.frame()
#record1=data.frame()
for (i in 3704:3854 ){
  for (j in 1:29){
    tryCatch({fitx=garchFit(~garch(1,1),data=data[i-500:i,j],trace=F,
    cond.dist = "std",tol=1e-20)
 record1[i-2999,j]=predict(fitx,1)$meanForecast},
 error = function(e) { skip_to_next <<- TRUE})

  }
  print(i)
}
output1=record
ag1=read.csv("ag1.csv")
pd=record
plot(seq(1,855),pd[,1],type="l",ylim=range(pd))
lines(seq(1,855),pd[,4],col="red")
lines(seq(1,855),pd[,3],col="blue")
lines(seq(1,855),pd[,2],col="green")
```

3 Codes List 3: ARIMA(1,0,1)

```
library(quantmod)
library(fGarch)
library(tseries)
library(pheatmap)
library(forecast)

getSymbols("DJI",src='yahoo')
seri=periodReturn(DJI,type='log',period='daily')
stocks=c("MMM","AXP", "AMGN","AAPL","BA","CAT","CVX","CSCO","KO","DIS","GS","HD","HON",
"IBM","INTC","JNJ", "JPM","MCD", "MRK","MSFT","NKE","PG","CRM","TRV","UNH",
"VZ", "WBA", "WMT")
getSymbols(stocks)

for (stk in stocks){
  sub_seri=periodReturn(get(stk),type='log',period='daily')
  seri=merge(seri,sub_seri)
}

col_names=c('DJI',"MMM","AXP", "AMGN","AAPL","BA","CAT","CVX","CSCO","KO","DIS",
"GS","HD","HON","IBM","INTC","JNJ", "JPM","MCD", "MRK","MSFT","NKE","PG","CRM",
"TRV","UNH","VZ", "WBA", "WMT")
data=data.frame(seri)
acf(data$JPM)
title('JPMACF')
pacf(data$JPM)
title('JPMPACF')
acf(data$TRV)
title('TRVACF')
pacf(data$TRV)
title('TRVPACF')
names(data)=col_names
data[is.na(data)]=0
par(mar=c(1,1,1,1))
```

```
insample=data[1:3000,]
stk_names=names(insample)

#calculate cor to filter stocks in the same industry
cormat=cor(data,method='pearson')
pheatmap(cormat)
#we select finance industry.TRV,AXP,GS,JPM
insample=insample[,c('TRV','AXP','GS','JPM')]


adf_pvalues=c(0)
for (i in 1:4){
  test=adf.test(insample[,i])
  adf_pvalues=append(adf_pvalues,test$p.value)
}
adf_pvalues=adf_pvalues[2:5]
names(adf_pvalues)=names(insample)


#collect aic to determine the best fitted model
bestaic=0
bestp=0
bestq=0
coefs=c(rep(0,14))
for (p in 2){
  for (q in 1:3){
    totalAIC=0
    for (i in 1:4){
      temp=arima(insample[,i],order=c(p,0,q))
      totalAIC=totalAIC+temp$aic
      vals=coef(temp)/sqrt(diag(vcov(temp)))
      sigVals=which(abs(vals)>1.96)
      for (num in 1:length(sigVals)){
        coefs[num]=coefs[num]+1
      }
    }
    if (totalAIC < bestaic){
      bestaic=totalAIC
      bestp=p
      bestq=q
    }
  }
}


lb_pvalues=c()
for (i in 1:4){
  temp=arima(insample[,i],order=c(1,0,1))
  val=Box.test(temp$residuals^2,lag=1)
  lb_pvalues=append(lb_pvalues,val$p.value)
}
names(lb_pvalues)=names(insample)
```

```
predSample=data[3000:3853,c('TRV','AXP','GS','JPM')]

for (i in 1:4){
  for (d in 3000:3853){
    subSample=data[d-60:d,i]
    fit=arima(subSample,order=c(1,0,1))
    output=predict(fit,1)
    predSample[d-2999,i]=output$pred[1]
  }
  print(Sys.time())
}
```