基于Garch族模型的特斯拉股票价格分析

吴昕怡 2019213016

指导老师：李晓花

---

1. 数据来源与概览

1.1 数据获取与平稳化处理

本文分析的数据为特斯拉股票价格日数据（2010-06-29 ~ 2022-6-17），共3015个交易日的数据。数据来源为Yahoo金融，采用自编写函数`getSymbols.yahoo.fix`爬取数据。

```
getSymbols.yahoo.fix <- function (symbol,
                                  from       ="2007-01-01",
                                  to         = Sys.Date(),
                                  period     = c("daily","weekly","monthly"),
                                  envir      = globalenv(),
                                  crumb      ="YourCrumb",
                                  DLdir      ="~/Downloads/") { #1
  # build yahoo query
  query1    <-
paste("https://query1.finance.yahoo.com/v7/finance/download/",symbol,"?",sep="")
  fromPosix <- as.numeric(as.POSIXlt(from))
  toPosix   <- as.numeric(as.POSIXlt(to))
  query2    <- paste("period1=", fromPosix,"&period2=", toPosix, sep ="")
  interval  <- switch(period[1], daily ="1d", weekly ="1wk", monthly ="1mo")
  query3    <- paste("&interval=", interval,"&events=history&crumb=", crumb, sep
="")
  yahooURL  <- paste(query1, query2, query3, sep ="")
  #' requires browser to be open
```

```r
  utils::browseURL("https://www.google.com")
  #' run the query - downloads the security as a csv file
  #' DLdir defaults to download directory in browser preferences
  utils::browseURL(yahooURL)
  #' wait 500 msec for download to complete - mileage may vary
  Sys.sleep(time = 0.5)
  yahooCSV  <- paste(DLdir, symbol,".csv", sep ="")
  yahooDF   <- utils::read.csv(yahooCSV, header = TRUE)
  #' -------
  #' if you get: Error in file(file,"rt") : cannot open the connection
  #' it's because the csv file has not completed downloading
  #' try increasing the time for Sys.sleep(time = x)
  #' -------
  #' delete the csv file
  file.remove(yahooCSV)
  # convert date as character to date format
  yahooDF$Date <- as.Date(yahooDF$Date)
  # convert to xts
  yahoo.xts    <- xts(yahooDF[,-1],order.by=yahooDF$Date)
  # assign the xts file to the specified environment
  # default is globalenv()
  assign(symbol, yahoo.xts, envir = as.environment(envir))
  print(symbol)
} #1
```

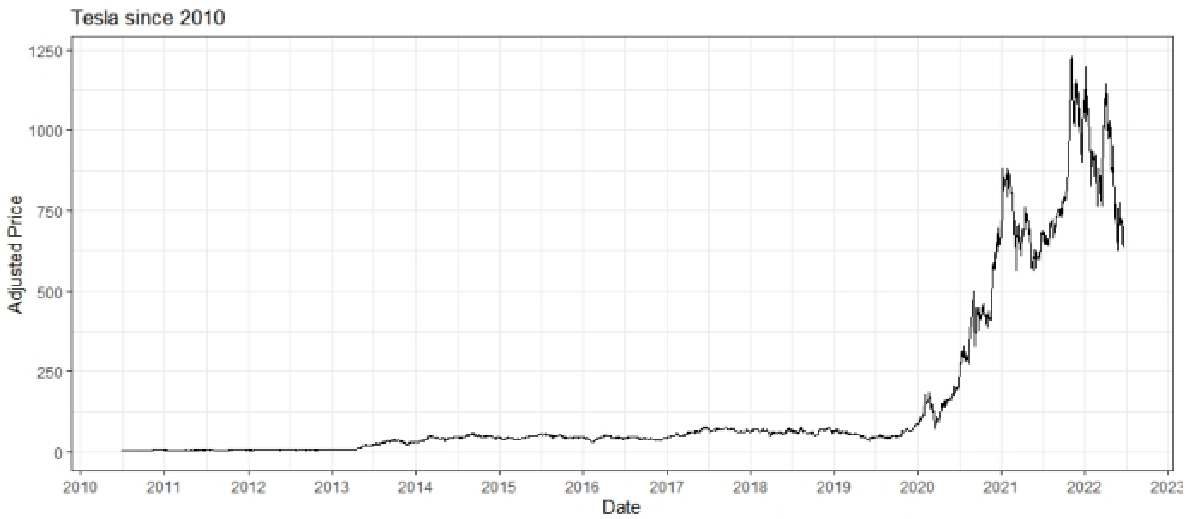
爬取数据后，经过数据导入与日期的处理，我们所得的数据为如下形式：

```
> head(TSLA)
        Date  Open  High   Low Close Adj.Close   Volume
1 2010-06-29 3.800 5.000 3.508 4.778     4.778 93831500
2 2010-06-30 5.158 6.084 4.660 4.766     4.766 85935500
3 2010-07-01 5.000 5.184 4.054 4.392     4.392 41094000
4 2010-07-02 4.600 4.620 3.742 3.840     3.840 25699000
5 2010-07-06 4.000 4.000 3.166 3.222     3.222 34334500
6 2010-07-07 3.280 3.326 2.996 3.160     3.160 34608500
```

Date日期，Open开盘价，High最高价，Low最低价，Close收盘价，Adj.Close调整后的收盘价，Volume为成交量。

调整后的收盘价是复权和分配股息后的价格，包括了在第二天开盘前任何时间发生的任何分配和公司行为。调整后的收盘价通常在检查历史收益或对历史收益进行详细分析时使用，因为它使分析师能够准确地反映公司的股票价值，而不仅仅是简单的市场价格。

所以我们选取调整后收盘价Adj.Close绘制时序图。可以看出，随着时间增大，股价呈现出了较剧烈的总体上升趋势，同时随着股价增加波动幅度也大幅增加。我们初步认为，它具有典型的金融时间序列数据的特征。



下面，借鉴常见的计算股票日收益率的处理方式，我们计算股票的日收益率Daily Returns，借助这种处理实现数据的平稳化。

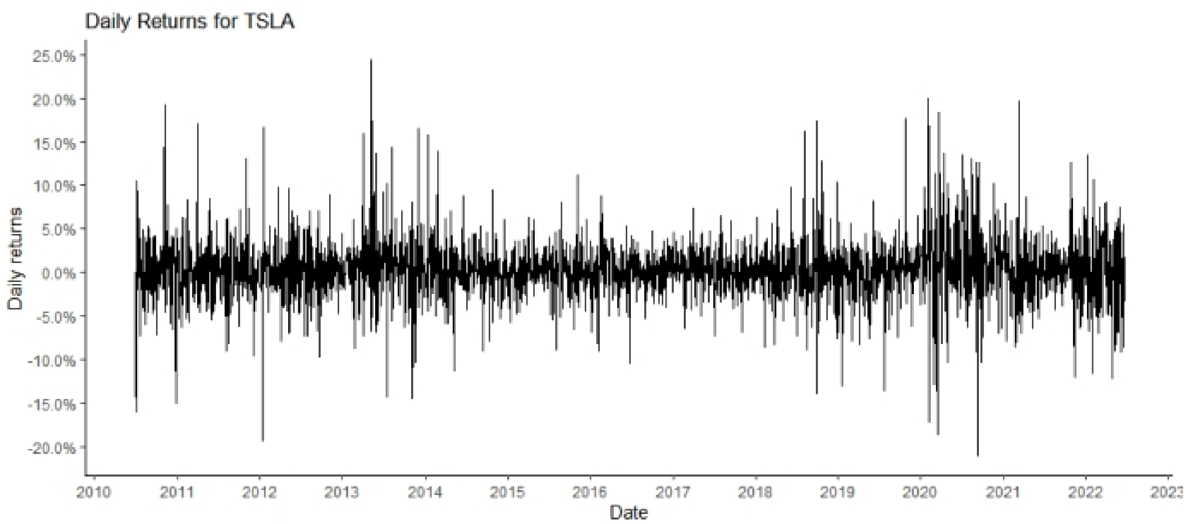
$$Daily\ returns = \frac{Close_t - Close_{t+1}}{Close_{t+1}}$$

我们通过调用`quantmod`包中的`periodReturn`函数并设置参数为`"daily"`来直接计算。

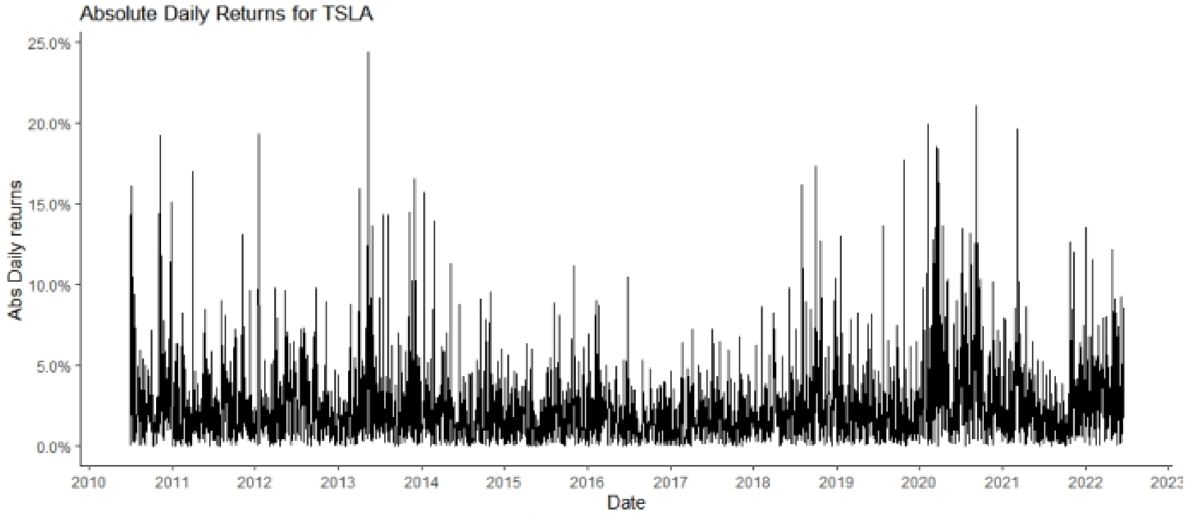
```
# Calculate daily returns

TSLA_daily_returns <- TSLA %>%
  tq_transmute(select = Adj.Close,           # this specifies which column to
select
               mutate_fun = periodReturn,   # This specifies what to do with
that column
               period = "daily",      # This argument calculates Daily returns
               col_rename = "tesla_returns") # renames the column
```

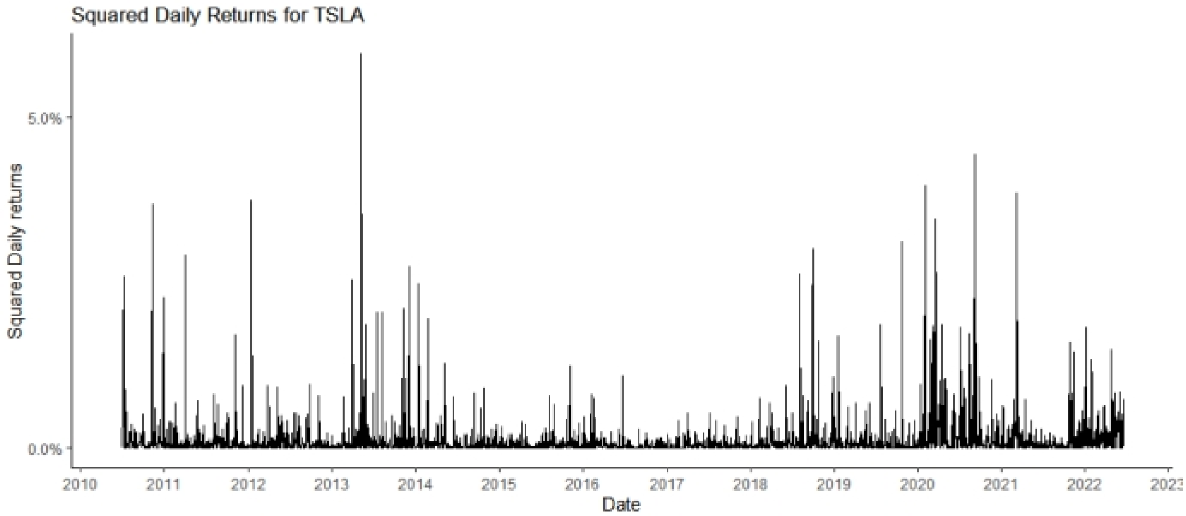
绘制日收益率时序图如下，可以看出，数据存在波动集群的现象，大量持续的平静与波动的数据相交替，**方差在一定时**

段中比较小，而在另一时段中比较大。 所以我们认为日收益率数据可能具有异方差的特点。



这种性质在绝对值日收益率与平方日收益率的时序图中体现更明显。





故，我们考虑先对Daily Returns数据的均值效应建立模型再对其残差进行GARCH模型拟合，提取其中异方差信息。对均值效应建立模型是把样本均值从数据中分离出来,如果样本均值显著地不同于零，可能会需要一个ARMA模型。

1.2 平稳性与随机性检验

由此，首先我们需要确定数据的平稳性与数据并不是白噪声序列两条前提，再进行ARMA模型的拟合。

1. 针对平稳性，进行单位根检验ADF test

单位根检验的原假设为序列有单位根，即此时间序列非平稳。而我们检验得到的p value = 0.01 < 0.05，故有充分理由拒绝原假设，认为日收益率数据为平稳时间序列。

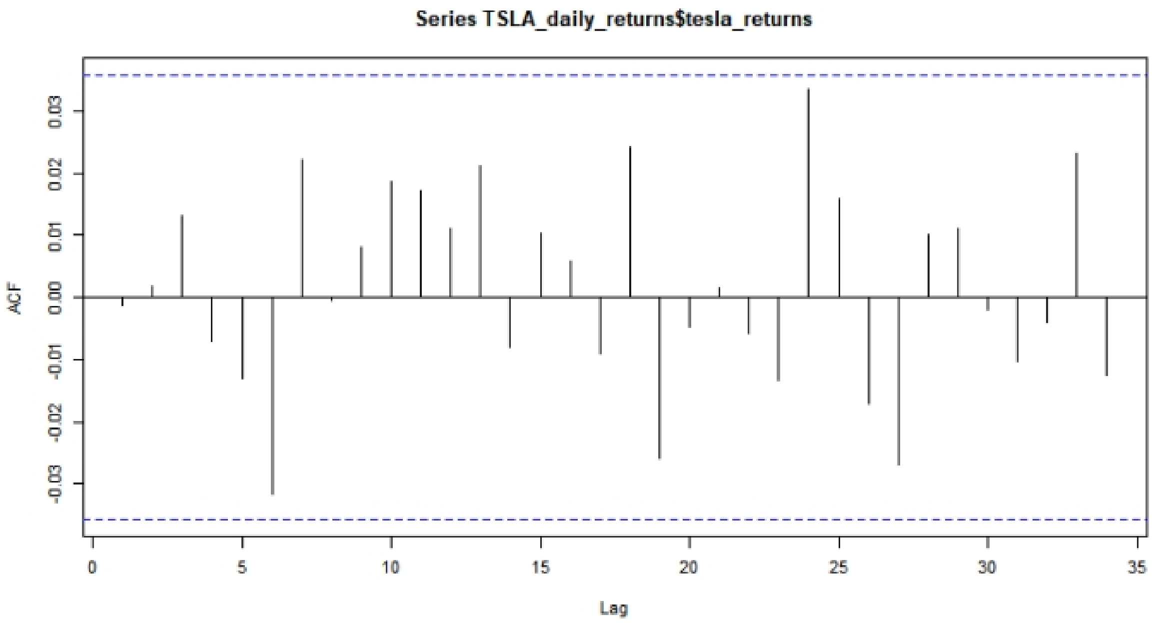
```
> adf.test(TSLA_daily_returns$tesla_returns)

    Augmented Dickey-Fuller Test

data:  TSLA_daily_returns$tesla_returns
Dickey-Fuller = -13.248, Lag order = 14, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(TSLA_daily_returns$tesla_returns) :
  p-value smaller than printed p-value
```

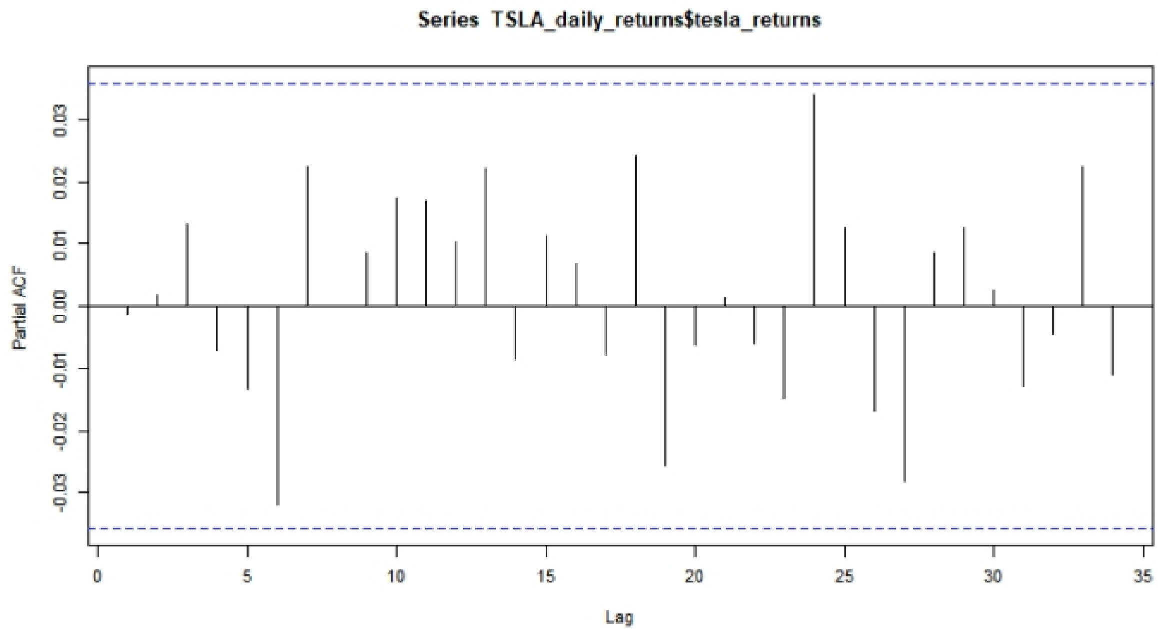
2.针对纯随机性，进行Ljung Box test 检验

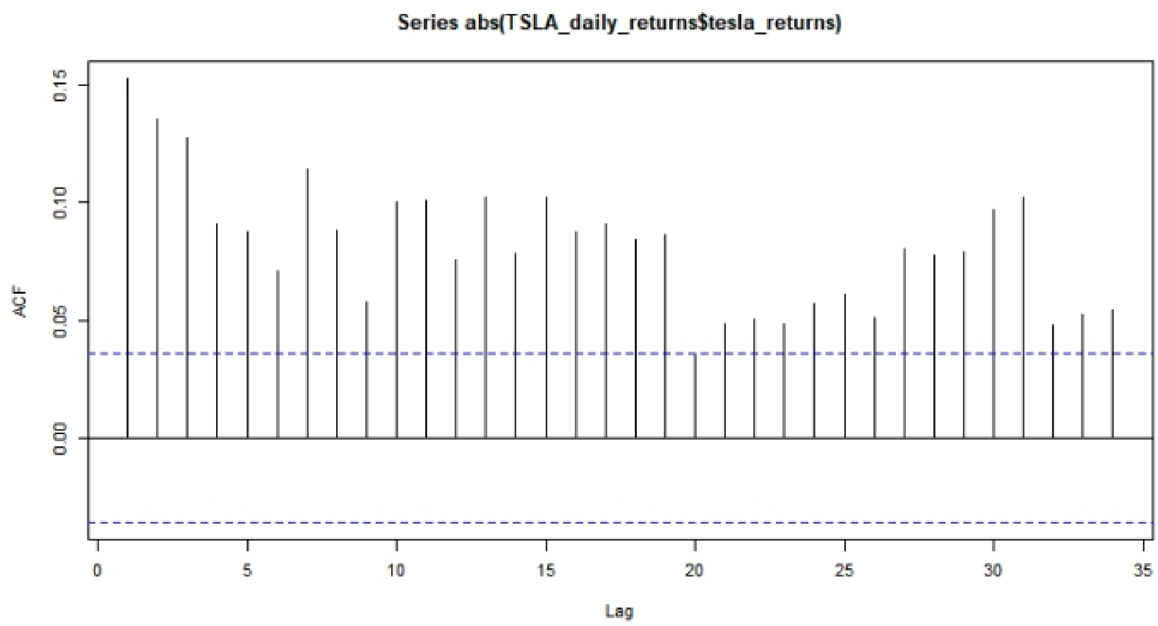
Ljung Box test 的原假设为序列为纯随机序列，检验得到的p value均较大，不能显著拒绝序列为纯随机序列的原假设。

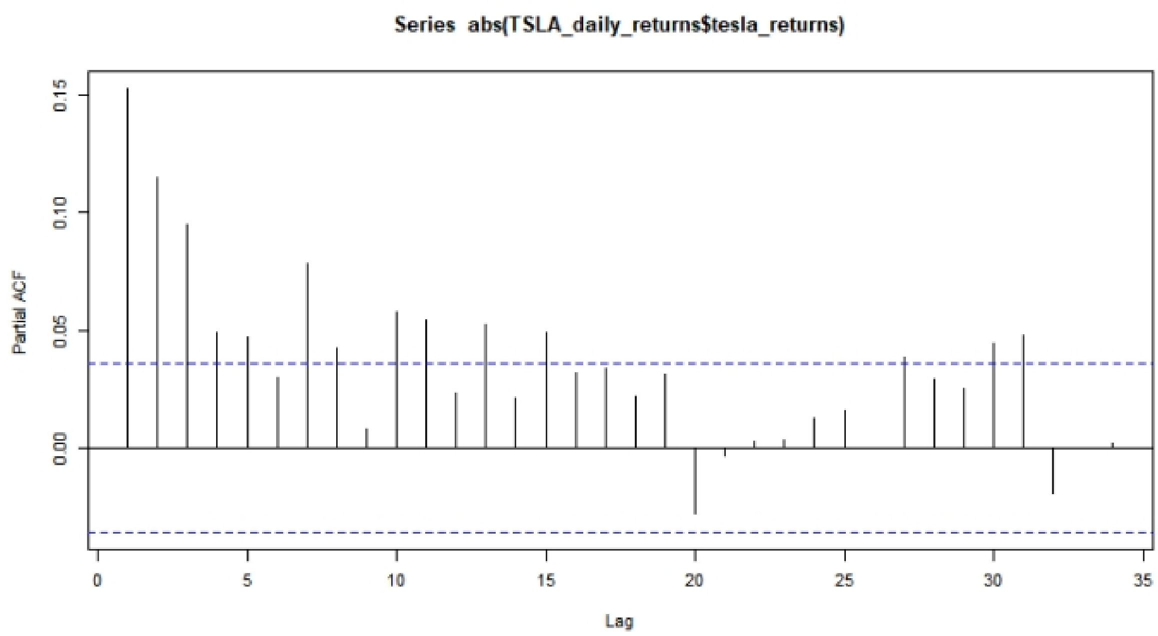
```
> Box.test(TSLA_daily_returns$tesla_returns,type="Ljung-Box",lag=12)

    Box-Ljung test

data:  TSLA_daily_returns$tesla_returns
X-squared = 8.2593, df = 12, p-value = 0.7646

> Box.test(TSLA_daily_returns$tesla_returns,type="Box",lag=12)

    Box-Pierce test

data:  TSLA_daily_returns$tesla_returns
X-squared = 8.2339, df = 12, p-value = 0.7666

> Box.test(TSLA_daily_returns$tesla_returns,type="Ljung-Box",lag=36)

    Box-Ljung test

data:  TSLA_daily_returns$tesla_returns
X-squared = 30.039, df = 36, p-value = 0.7472

> Box.test(TSLA_daily_returns$tesla_returns,type="Box",lag=36)

    Box-Pierce test

data:  TSLA_daily_returns$tesla_returns
X-squared = 29.811, df = 36, p-value = 0.7568
```

但通过ACF与PACF图我们可以看出，自相关性存在，数据并不是白噪声序列。这种自相关性在绝对值的ACF与PACF图中非常明显，几乎所有的值都超过5%的置信区间。




Series TSLA_daily_returns$tesla_returns
Partial ACF
Lag


Series abs(TSLA_daily_returns$tesla_returns)
ACF
Lag


Series abs(TSLA_daily_returns$tesla_returns)
Partial ACF
Lag

我们运用`t.test`对日收益率`TSLA_daily_returns$tesla_returns`进行 t 检验，检验其均值是否显著，检验的p value 小于0.05，否定均值不显著的原假设，故我们可以先将均值提取出来。

检验中同时给出了均值的值与95%的置信区间：

$$mean = 0.002270111$$
$$95\%\ confidence\ interval: \quad [0.0009894132\ ,\ 0.0035508079]$$

```
> t.test(TSLA_daily_returns$tesla_returns)

	One Sample t-test

data:  TSLA_daily_returns$tesla_returns
t = 3.4755, df = 3014, p-value = 0.000517
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.0009894132 0.0035508079
sample estimates:
  mean of x 
0.002270111
```

后续ARMA-GARCH模型的拟合都是针对提取均值后的数据， 我们将其记作 `dailyreturn`

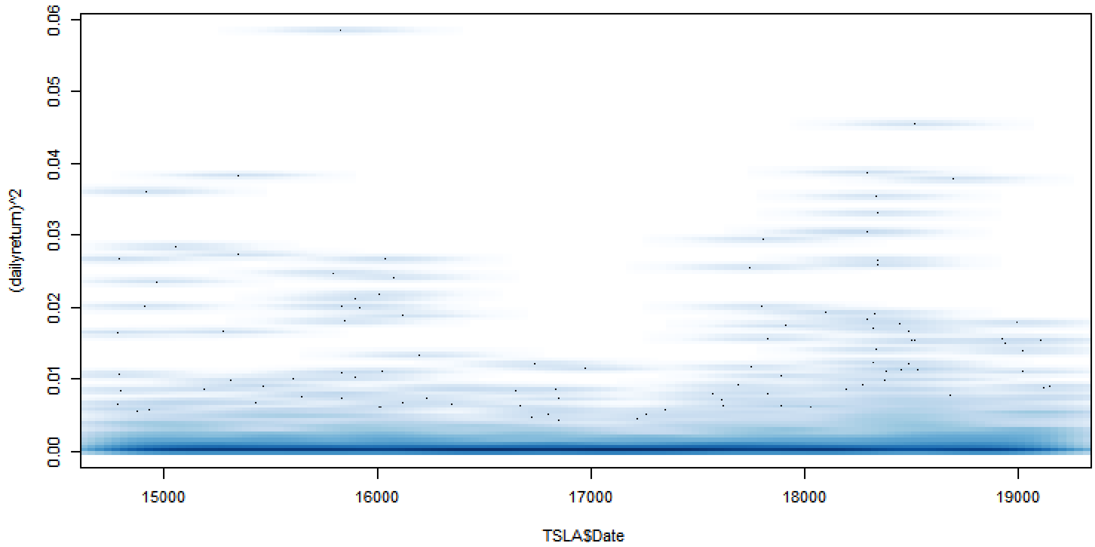
```
> dailyreturn<-TSLA_daily_returns$tesla_returns-
mean(TSLA_daily_returns$tesla_returns)
```

2.异方差检验

下面我们需要判断提取均值方程之后的残差 dailyreturn 是否具有异方差性我们采取两种方法，分别从直观可视的角度和统计量检验的角度来判断异方差性。

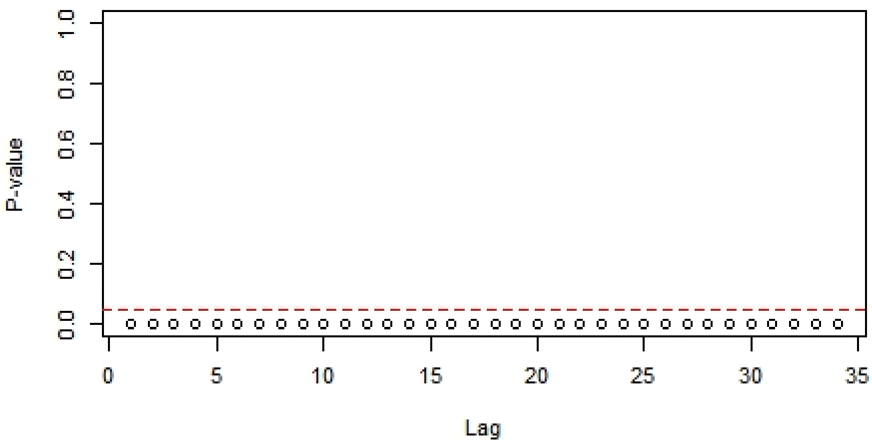
1. 图示法检验

   我们运用`smoothScatter(x=TSLA$Date,y=(dailyreturn)^2)`语句绘制出残差的平方与时间之间的高密度散点图，我们可以看到残差的平方随着时间的推移在发生变化，总体方差变化呈现抛物线式先下降后上升的趋势。所以我们认为，从图中我们判断残差dailyreturn具有异方差性。



2. 统计量检验

首先，我们采用 McLeod-Li test 来检验，检验的原假设为在考虑的滞后中没有自回归条件异方差性（autoregressive conditional heteroskedasticity ARCH）。通过绘制一系列滞后阶数的 p value 图，我们可以看到，在考虑的所有滞后阶数中，p value 均小于0.05，我们有充分的理由拒绝原假设，认定 dailyreturn 存在异方差性。红线标示着5%的显著水平。



我们也采用 LM 统计量检验，原假设与上面相同，检验所得的 p value 远小于0.05，与我们上面做出的判断相吻合。

```
> ArchTest(dailyreturn)

    ARCH LM-test; Null hypothesis: no ARCH effects

data:  dailyreturn
Chi-squared = 160.05, df = 12, p-value < 2.2e-16
```

我们还可以采用F统计量检验，检验的结果也符合我们的预期。

```
> source("archTest.R")  # R script available on the book web site.
> archTest(dailyreturn,12)
```

```

Call:
lm(formula = atsq ~ x)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.008635 -0.000971 -0.000708 -0.000106  0.056609 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.776e-04  8.232e-05   8.231 2.73e-16 ***
x1           1.239e-01  1.828e-02   6.776 1.48e-11 ***
x2           9.576e-02  1.841e-02   5.202 2.10e-07 ***
x3           5.959e-02  1.847e-02   3.226  0.00127 ** 
x4           3.758e-02  1.850e-02   2.032  0.04226 *  
x5           2.431e-02  1.850e-02   1.314  0.18893    
x6          -9.522e-03  1.849e-02  -0.515  0.60662    
x7           1.916e-02  1.848e-02   1.036  0.30010    
x8           2.861e-02  1.834e-02   1.560  0.11885    
x9          -2.329e-02  1.830e-02  -1.273  0.20328    
x10          4.401e-02  1.828e-02   2.408  0.01609 *  
x11          3.269e-02  1.822e-02   1.794  0.07285 .  
x12          3.088e-02  1.807e-02   1.708  0.08765 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.003372 on 2990 degrees of freedom
Multiple R-squared:  0.0533,    Adjusted R-squared:  0.0495 
F-statistic: 14.03 on 12 and 2990 DF,  p-value: < 2.2e-16
```

由此我们确认了数据的自回归条件异方差性。

3.GARCH模型定阶与比较

为了确定最优的模型阶数，我们定义函数`find_best_arch_model`用于寻求最优的GARCH模型，判断标准为最小AIC准则与最小BIC准则。

函数中设置的参数含义如下

- x 拟合数据
- type_models 用于拟合的GARCH模型种类，如sGARCH（标准GARCH），eGARCH（指数GARCH）等
- dist_to_use 模型中扰动项分布假设，如norm（正态分布），std（student 分布，即t分布）等
- max_lag_AR\MA\ARCH\GARCH 循环中AR\MA\ARCH\GARCH模型最大阶数

```
find_best_arch_model <- function(x, 
                                 type_models, 
                                 dist_to_use,
                                 max_lag_AR,
                                 max_lag_MA,
                                 max_lag_ARCH,
                                 max_lag_GARCH) {

  require(tidyr)

```

```
  df_grid <- expand_grid(type_models = type_models,
                         dist_to_use = dist_to_use,
                         arma_lag = 0:max_lag_AR,
                         ma_lag = 0:max_lag_MA,
                         arch_lag = 1:max_lag_ARCH,
                         garch_lag = 1:max_lag_GARCH)


  l_out <- pmap(.l = list(x = rep(list(x), nrow(df_grid)),
                          type_model = df_grid$type_models,
                          type_dist = df_grid$dist_to_use,
                          lag_ar = df_grid$arma_lag,
                          lag_ma = df_grid$ma_lag,
                          lag_arch = df_grid$arch_lag,
                          lag_garch  = df_grid$garch_lag),
                do_single_garch)

  tab_out <- bind_rows(l_out)

  # find by AIC
  idx <- which.min(tab_out$AIC)
  best_aic <- tab_out[idx, ]

  # find by BIC
  idx <- which.min(tab_out$BIC)
  best_bic <- tab_out[idx, ]

  l_out <- list(best_aic = best_aic,
                best_bic = best_bic,
                tab_out = tab_out)

  return(l_out)
}
```

其中调用的`do_single_garch`函数定义如下

```
do_single_garch <- function(x,
                            type_model,
                            type_dist,
                            lag_ar,
                            lag_ma,
                            lag_arch,
                            lag_garch) {
  require(rugarch)


  spec = ugarchspec(variance.model = list(model =  type_model,
                                          garchOrder = c(lag_arch, lag_garch)),
                    mean.model = list(armaOrder = c(lag_ar, lag_ma)),
                    distribution = type_dist)

  message('Estimating ARMA(',lag_ar, ',', lag_ma,')-',
          type_model, '(', lag_arch, ',', lag_garch, ')',
          ' dist = ', type_dist,
```

```
          appendLF = FALSE)

  try({
    my_rugarch <- list()
    my_rugarch <- ugarchfit(spec = spec, data = x)
  })

  if (!is.null(coef(my_rugarch))) {
    message('\tDone')

    AIC <- rugarch::infocriteria(my_rugarch)[1]
    BIC <- rugarch::infocriteria(my_rugarch)[2]
  } else {
    message('\tEstimation failed..')

    AIC <- NA
    BIC <- NA
  }

  est_tab <- tibble(lag_ar,
                    lag_ma,
                    lag_arch,
                    lag_garch,
                    AIC =  AIC,
                    BIC = BIC,
                    type_model = type_model,
                    type_dist,
                    model_name = paste0('ARMA(', lag_ar, ',', lag_ma, ')+',
                                        type_model, '(', lag_arch, ',',
lag_garch, ') ',
                                        type_dist) )

  return(est_tab)
}
```

在应用此寻求最优模型函数时，我们对参数的设定为

```
max_lag_AR <- 1
max_lag_MA <- 1
max_lag_ARCH <- 2
max_lag_GARCH <- 1
dist_to_use <- c('norm', 'std')
models_to_estimate <- c('sGARCH', 'eGARCH', 'gjrGARCH')
```

拟合结果初步为表格的形式，查看前6行结果，我们发现一些模型拟合失败，AIC与BIC为缺失值NA。

```
out <- find_best_arch_model(x = dailyreturn,
                            type_models = models_to_estimate,
                            dist_to_use = dist_to_use,
                            max_lag_AR = max_lag_AR,
                            max_lag_MA = max_lag_MA,
                            max_lag_ARCH = max_lag_ARCH,
                            max_lag_GARCH = max_lag_GARCH)

# get table with estimation results
tab_out <- out$tab_out
```

```
> head(tab_out)
# A tibble: 6 x 9
  lag_ar lag_ma lag_arch lag_garch   AIC   BIC type_model type_dist model_name

   <int>  <int>    <int>     <int> <dbl> <dbl> <chr>      <chr>     <chr>

1      0      0        1         1 -3.53 -3.53 sGARCH     norm
ARMA(0,0)+sGARCH(1,1) norm
2      0      0        2         1 -3.50 -3.49 sGARCH     norm
ARMA(0,0)+sGARCH(2,1) norm
3      0      1        1         1 NA    NA    sGARCH     norm
ARMA(0,1)+sGARCH(1,1) norm
4      0      1        2         1 NA    NA    sGARCH     norm
ARMA(0,1)+sGARCH(2,1) norm
5      1      0        1         1 NA    NA    sGARCH     norm
ARMA(1,0)+sGARCH(1,1) norm
6      1      0        2         1 NA    NA    sGARCH     norm
ARMA(1,0)+sGARCH(2,1) norm
```

选取一个失败的拟合ARMA(0,1)+sGARCH(1,1)单独进行拟合，可以看出，单独拟合时存在如下报错 `solver failer to converge`，通过查阅资料（https://stackoverflow.com/questions/43710302/solver-failer-in-garch-model），我们发现报错原因为模型中求解器失效，我们更换求解器为 `hybrid` 即可解决。

```
 failed_spec = ugarchspec(variance.model=list(model="sGARCH",
garchOrder=c(1,1)),
+                          mean.model=list(armaOrder=c(0,1), include.mean=TRUE),
+                          distribution.model="norm", fixed.pars=list(omega=0))
> failedfit = ugarchfit(spec = failed_spec, data = dailyreturn)
Warning message:
In .sgarchfit(spec = spec, data = data, out.sample = out.sample,  :
ugarchfit-->warning: solver failer to converge.
> failedfit

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(0,0,1)
```

```
Distribution    : norm

Convergence Problem:
Solver Message:
```

```
failedfit = ugarchfit(spec = failed_spec, data = dailyreturn,solver = "hybrid")
> failedfit

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(0,0,1)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu       0.00000    0.000585   0.000000  1.00000
ma1      0.00006    0.019664   0.003046  0.99757
omega    0.00000          NA         NA       NA
alpha1   0.04686    0.003142  14.916055  0.00000
beta1    0.95954    0.002196 436.968985  0.00000

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu       0.00000    0.000651   0.000000  1.00000
ma1      0.00006    0.020649   0.002901  0.99769
omega    0.00000          NA         NA       NA
alpha1   0.04686    0.003878  12.082840  0.00000
beta1    0.95954    0.002765 347.045604  0.00000

LogLikelihood : 5909.891

Information Criteria
------------------------------------

Akaike       -3.9177
Bayes        -3.9097
Shibata      -3.9177
Hannan-Quinn -3.9148

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2590  0.6108
Lag[2*(p+q)+(p+q)-1][2]    0.3112  0.9934
Lag[4*(p+q)+(p+q)-1][5]    0.4282  0.9962
d.o.f=1
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
```

```
------------------------------------
                        statistic p-value
Lag[1]                      3.492 0.06167
Lag[2*(p+q)+(p+q)-1][5]     5.170 0.14010
Lag[4*(p+q)+(p+q)-1][9]     6.500 0.24473
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]  0.004158 0.500 2.000  0.9486
ARCH Lag[5]  0.191124 1.440 1.667  0.9676
ARCH Lag[7]  1.213091 2.315 1.543  0.8766

Nyblom stability test
------------------------------------
Joint Statistic:  3.0991
Individual Statistics:
mu     0.46839
ma1    0.10139
alpha1 0.15171
beta1  0.02511

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.07 1.24 1.6
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                     t-value   prob sig
Sign Bias          0.0007619 0.9994
Negative Sign Bias 1.2156467 0.2242
Positive Sign Bias 0.8789156 0.3795
Joint Effect       4.2158439 0.2391


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     249.6    3.723e-42
2    30     275.0    6.802e-42
3    40     280.5    2.656e-38
4    50     299.6    1.093e-37


Elapsed time : 0.8495491
```

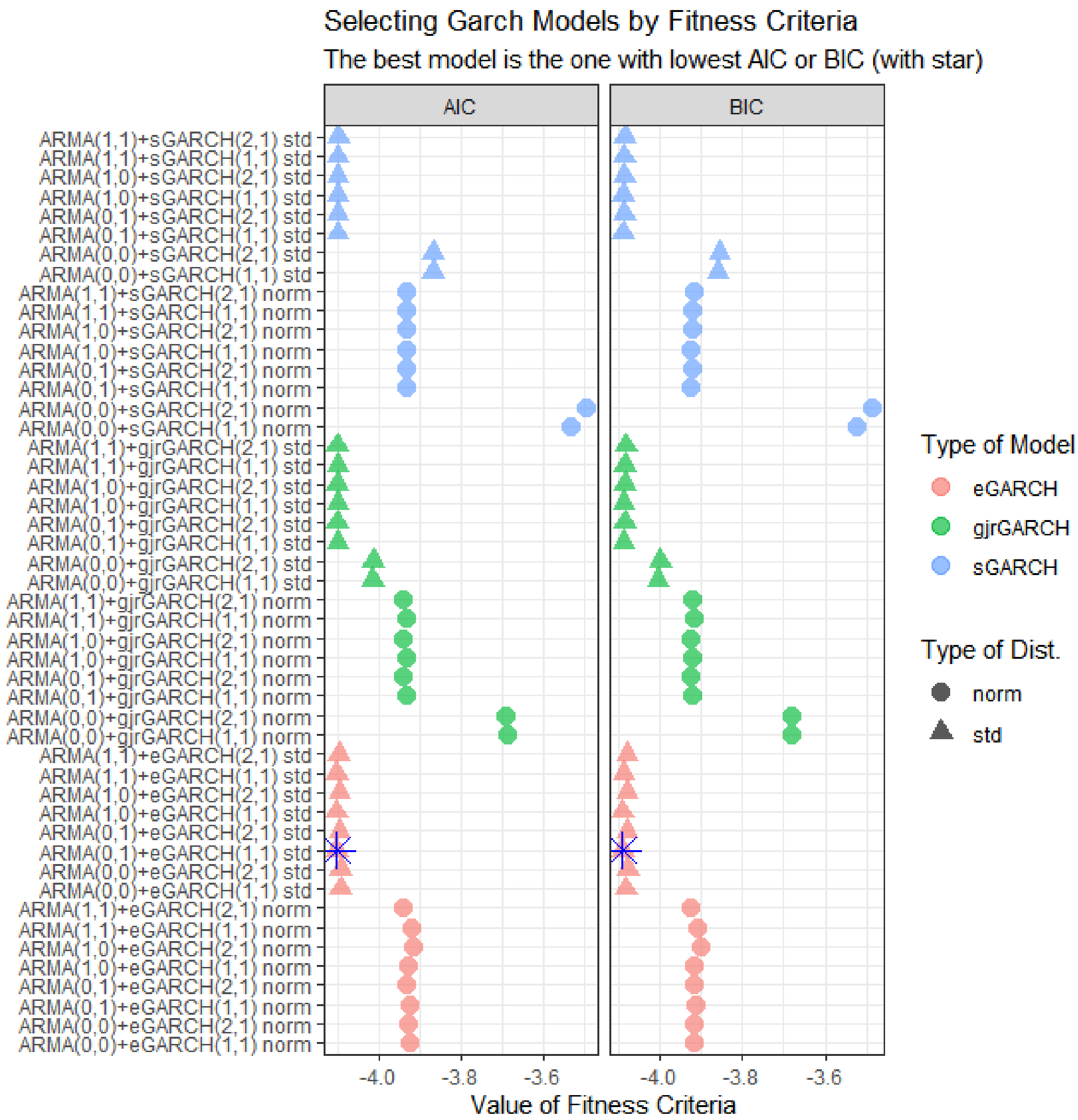
同样，我们在寻求最优模型函数中调用的`do_single_garch`函数中也进行修正，可以得到新的循环后模型表格。现在所有的模型都能成功拟合。

```
head(tab_out)
# A tibble: 6 x 9
  lag_ar lag_ma lag_arch lag_garch   AIC   BIC type_model type_dist model_name


   <int>  <int>    <int>     <int> <dbl> <dbl> <chr>      <chr>     <chr>


1      0      0        1         1 -3.53 -3.53 sGARCH     norm
ARMA(0,0)+sGARCH(1,1) norm
2      0      0        2         1 -3.50 -3.49 sGARCH     norm
ARMA(0,0)+sGARCH(2,1) norm
3      0      1        1         1 -3.93 -3.92 sGARCH     norm
ARMA(0,1)+sGARCH(1,1) norm
4      0      1        2         1 -3.93 -3.92 sGARCH     norm
ARMA(0,1)+sGARCH(2,1) norm
5      1      0        1         1 -3.93 -3.92 sGARCH     norm
ARMA(1,0)+sGARCH(1,1) norm
6      1      0        2         1 -3.93 -3.92 sGARCH     norm
ARMA(1,0)+sGARCH(2,1) norm
>
```

通过绘制拟合模型基于最小AIC与最小BIC准则进行比较的图，可以直观地看出，对于求解器调整后的循环，模型ARMA(0,1)+eGARCH(1,1)std为最优的模型。

通过图像，我们认为选取以下3个模型进行比较较为合适

1. （简称做最优拟合模型 **my_best_garch**）ARMA(0,1)+eGARCH(1,1)std
2. （备选模型1 **m1fit**）ARMA(0,1)+sGARCH(1,1)std
3. （备选模型2 **m2fit**）ARMA(1,0)+sGARCH(1,1)std



附 GARCH-M

我们可以通过以下代码拟合GARCH-M模型，但模型无法收敛。

```
#Fit a GARCH-M Model with Normal Distribution
myspec1=ugarchspec(variance.model=list(model="fGARCH",garchOrder=c(1,1),submodel
="GARCH"),mean.model=list(armaOrder=c(1,0),include.mean=T,archm=T,archpow=2),dis
tribution.model="norm")
gm_garchm=ugarchfit(spec=myspec1,data=RV.xts,solver="hybrid")
gm_garchm
```

4.模型拟合残差检验

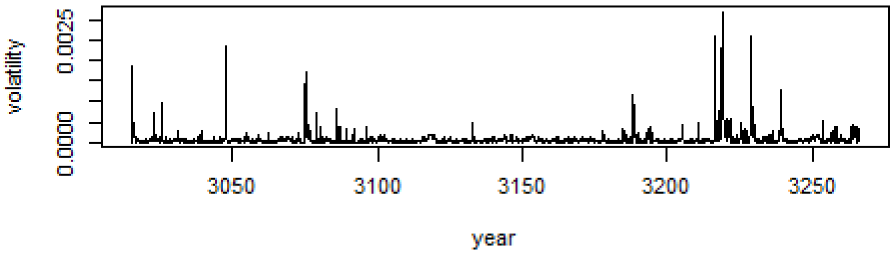
检查最优拟合模型的信息，可以看出

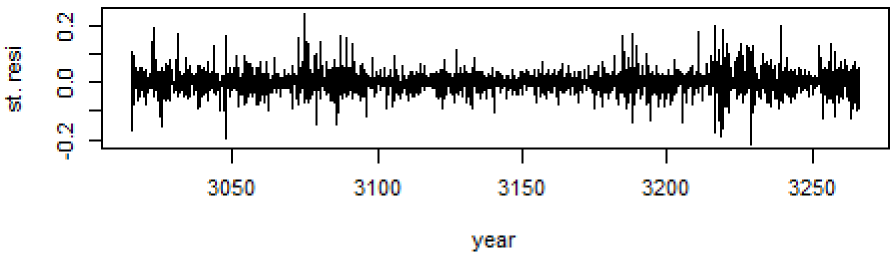
1. 模型所有的参数均显著，均大于或接近于其标准差
2. 模型进行了关于残差及残差平方的 Ljung-Box 检验， p value 均远大于于0.05，不能拒绝残差无自相关性的原假设
3. 模型进行了关于残差是否仍具有ARCH效应的 LM 检验，p value 均远大于于0.05，不能拒绝无ARCH效应的原假设

```
> my_best_garch <- ugarchfit(spec = best_spec,
+                            data = dailyreturn,solver = 'hybrid')
> my_best_garch

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : eGARCH(1,1)
Mean Model  : ARFIMA(0,0,1)
Distribution    : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.000000    0.000546   0.00000  1.00000
ma1    -0.010899    0.017745  -0.61421  0.53907
omega  -0.117876    0.009730 -12.11416  0.00000
alpha1 -0.016127    0.011702  -1.37819  0.16814
beta1   0.982478    0.001397 703.27645  0.00000
gamma1  0.145033    0.019949   7.27012  0.00000
shape   3.690339    0.261837  14.09404  0.00000

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000000    0.000643    0.00000  1.00000
ma1    -0.010899    0.017240   -0.63221  0.52725
omega  -0.117876    0.005013  -23.51475  0.00000
alpha1 -0.016127    0.014554   -1.10805  0.26784
beta1   0.982478    0.000610 1609.67953  0.00000
gamma1  0.145033    0.022051    6.57724  0.00000
shape   3.690339    0.262474   14.05985  0.00000

LogLikelihood : 6193.566

Information Criteria
------------------------------------

Akaike       -4.1039
Bayes        -4.0899
Shibata      -4.1039
Hannan-Quinn -4.0988

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.7203  0.3960
Lag[2*(p+q)+(p+q)-1][2]    0.7673  0.8674
Lag[4*(p+q)+(p+q)-1][5]    1.0059  0.9460
d.o.f=1
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
```

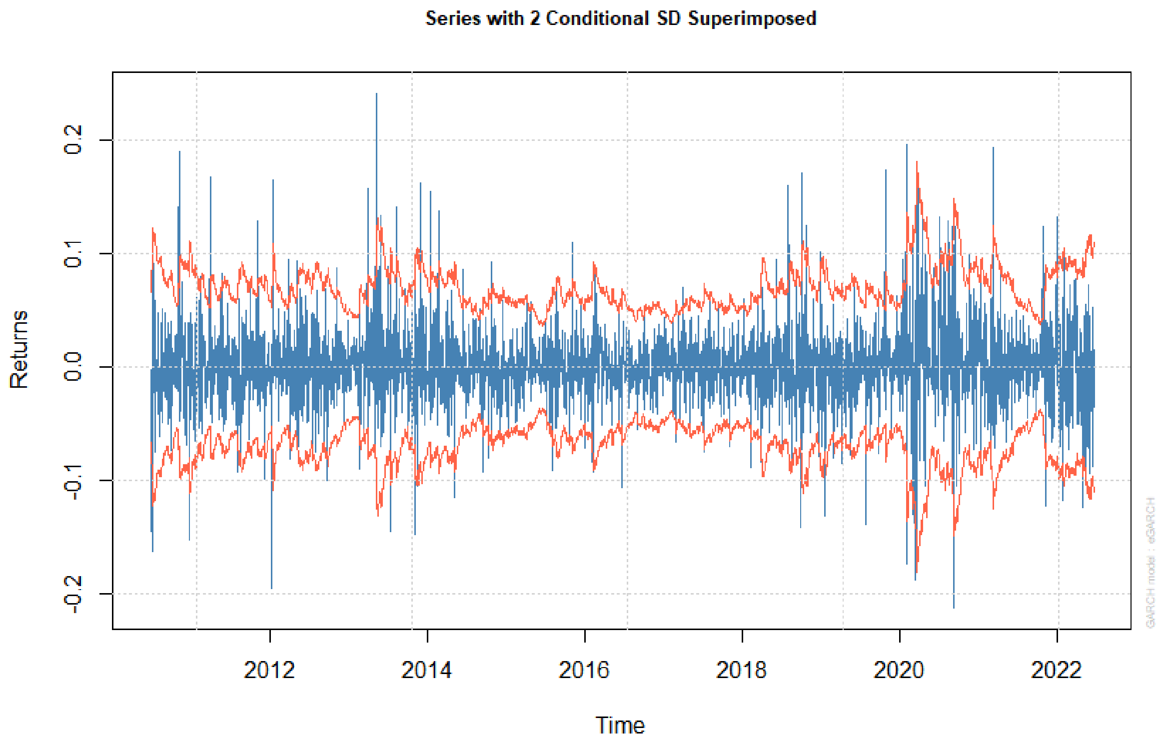
```
------------------------------------
                        statistic  p-value
Lag[1]                      8.518 0.003516
Lag[2*(p+q)+(p+q)-1][5]    10.494 0.006872
Lag[4*(p+q)+(p+q)-1][9]    12.210 0.016245
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3] 0.0009018 0.500 2.000  0.9760
ARCH Lag[5] 0.4301238 1.440 1.667  0.9040
ARCH Lag[7] 1.7852082 2.315 1.543  0.7627

Nyblom stability test
------------------------------------
Joint Statistic:  2.6368
Individual Statistics:
mu     1.46991
ma1    0.08143
omega  0.39630
alpha1 0.09528
beta1  0.35497
gamma1 0.22627
shape  0.39571

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.69 1.9 2.35
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias          0.43913 0.6606
Negative Sign Bias 1.24608 0.2128
Positive Sign Bias 0.01885 0.9850
Joint Effect       3.65834 0.3008


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     23.60     0.211932
2    30     51.34     0.006455
3    40     51.57     0.085653
4    50     69.06     0.030943


Elapsed time : 1.439855
```

由模型拟合残差图也可以看出，残差波动率中虽仍可能存在某种趋势，但相较于原始数据，异方差现象已经大大改善。波动率图也如下。





进一步绘制条件异方差和无条件方差比较示意图，当序列大幅度波动时，条件异方差置信区间大于无条件方差置信区间；当序列小范围波动时，条件异方差置信区间又小于无条件方差置信区间。显然条件异方差更好地拟合了序列的集群波动特征，更接近序列的真实波动情况，对序列的预测也将会更加准确。



下面我们从三个方面来检验残差的正态性:

1. Jarque-Bera test

   Jarque-Bera test的原假设为数据符合正态分布，检验 p value 均小于0.05，拒绝原假设，残差为非正态分布。

```
> r<-my_best_garch@fit$residuals
> r<-na.omit(r)
> jarque.bera.test(r)

	Jarque Bera Test

data:  r
X-squared = 3731.5, df = 2, p-value < 2.2e-16

> shapiro.test(r)

	Shapiro-Wilk normality test

data:  r
W = 0.92907, p-value < 2.2e-16
```

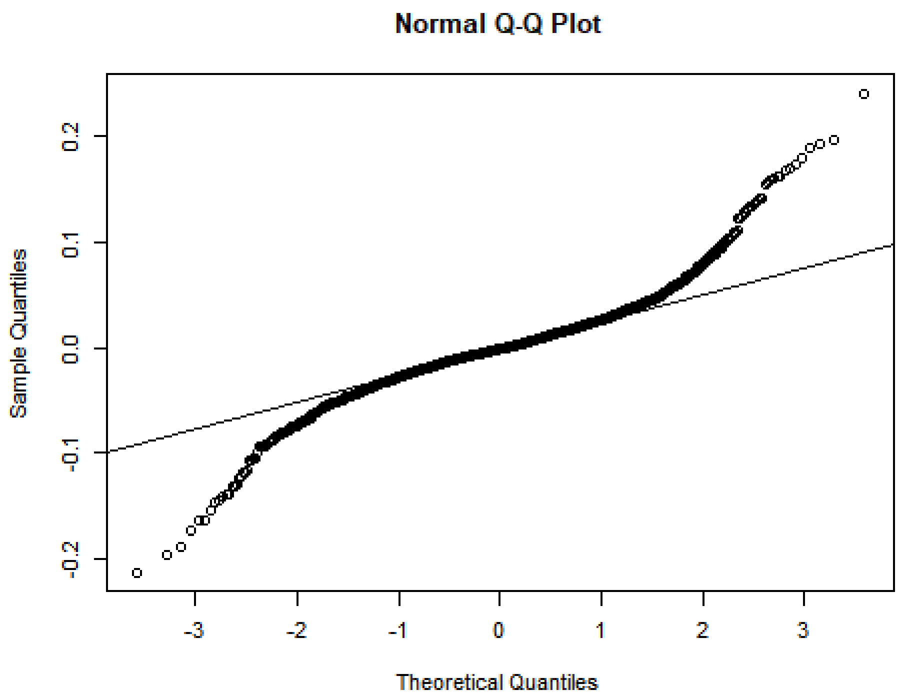
2. Shapiro-Wilk test

   该检验的原假设与结果均同上。

3. QQ图

   由QQ图我们也能直观地看出，残差并不满足正态分布。

   事实上，残差呈现长尾分布，比标准正态分布有更多偏离的数据。



下面我们来检验残差的平稳性，显然，残差为平稳序列。

```
> adf.test(r)

	Augmented Dickey-Fuller Test

data:  r
Dickey-Fuller = -13.237, Lag order = 14, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(r) : p-value smaller than printed p-value
```

用类似的方法检验另外两个模型，可以得到模型比较表如下。我们仍认为最优拟合模型**my_best_garch**为我们最优的选择。

model_name	AR阶数	MA阶数	ARCH阶数	GARCH阶数	GARCH模型类型	参数是否显著	残差正态性	残差平稳性	残差是否消除异方差	AIC	BIC
my_best_garch	0	1	1	1	eGARCH	√	×	√	√	-4.103858	-4.089901
m1fit	0	1	1	1	sGARCH	√	×	√	√	-4.095015	-4.085046
m2fit	1	0	1	1	sGARCH	√	×	√	√	-4.095012	-4.085042

5.回测检验与预测

5.1 回测检验

下面我们利用`rugarch`包中的`garchroll`函数对三种模型进行回测检验。我们采用回测检验的平均绝对差误 (MAFE) 来度量模型预测能力。我们可以看到最优拟合模型的误差最大，备选模型2的误差最小，但它们之间的差别较小。所以我们认为三种模型的预测能力在同一水平。

```
> garchroll<-ugarchroll(best_spec, data = RV.xts,n.start =500,
+                       refit.window="moving", refit.every =200)
> garchroll

*-------------------------------------*
*              GARCH Roll             *
*-------------------------------------*
No.Refits        : 13
Refit Horizon    : 200
No.Forecasts     : 2515
GARCH Model      : eGARCH(1,1)
Distribution     : std

Forecast Density:
                Mu  Sigma Skew  Shape Shape(GIG) Realized
2012-06-21  0.0007 0.0364    0 3.7256          0  -0.0493
2012-06-22 -0.0033 0.0433    0 3.7256          0   0.0474
2012-06-25  0.0007 0.0423    0 3.7256          0  -0.0224
2012-06-26 -0.0022 0.0409    0 3.7256          0  -0.0476
2012-06-27 -0.0031 0.0447    0 3.7256          0   0.0088
2012-06-28 -0.0008 0.0392    0 3.7256          0  -0.0195
```

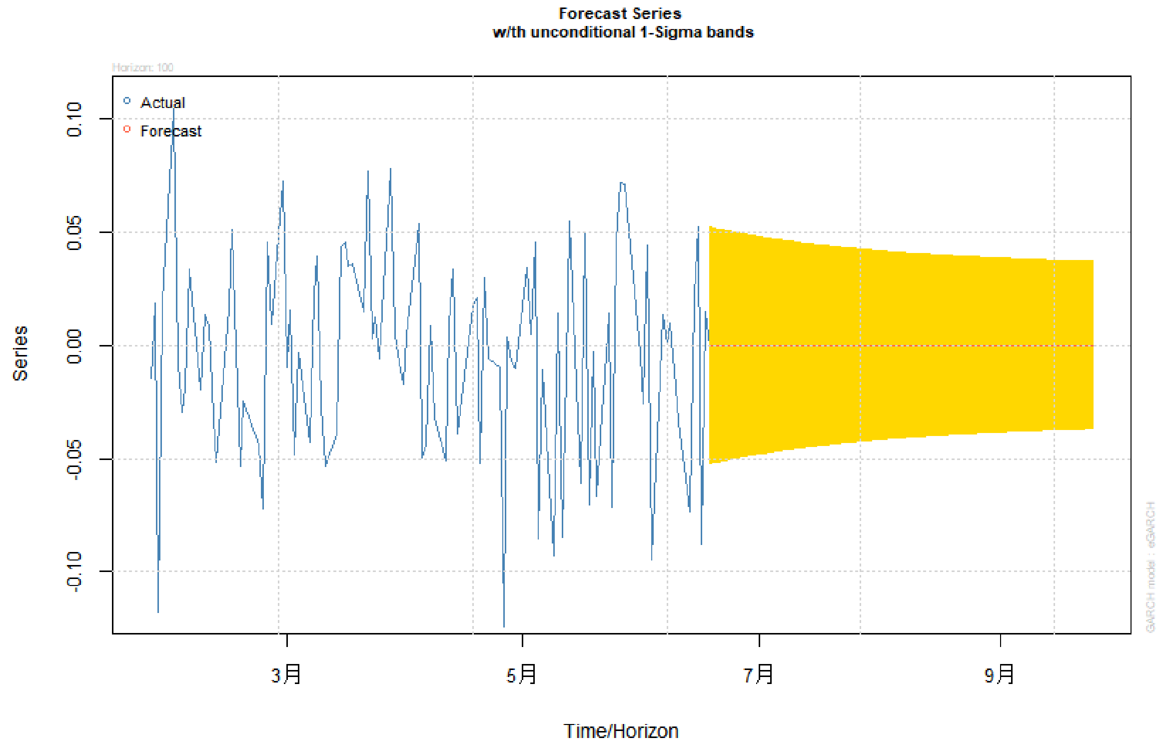
```
..........................
                Mu  Sigma Skew  Shape Shape(GIG) Realized
2022-06-10  0.0024 0.0463    0 4.1844          0  -0.0335
2022-06-13  0.0033 0.0468    0 4.1844          0  -0.0733
2022-06-14  0.0050 0.0533    0 4.1844          0   0.0216
2022-06-15  0.0012 0.0502    0 4.1844          0   0.0526
2022-06-16 -0.0002 0.0521    0 4.1844          0  -0.0877
2022-06-17  0.0054 0.0595    0 4.1844          0   0.0149

Elapsed: 6.047919 secs

> preds<-as.data.frame(garchroll)
> #prediction error
> e=(preds$Realized-preds$Mu)/preds$Realized
> mean(abs(e))
[1] 1.267818
```

model name	MAFE
my_best_garch	1.267818
m1_fit	1.261211
m2_fit	1.260036

5.2 预测

GARCH模型的预测有两种，一种为直接预测，一种为滚动预测，用第1天到第100天真实数据预测第101天数据，用第2天到第101天真实数据预测第102天数据，用第3天到第102天真实数据预测第103天数据……

我们采用直接预测，向后预测100天，并标记出95%置信区间。



6.异常值处理

对于GARCH类模型异常值的处理，我们采用Doornik与Ooms于2002年提出的方法[1]，其基本步骤为

Step 1

先采用GARCH模型拟合，得到对数似然函数值lb（log-likelihood）和残差（residuals）,再找到最大的（绝对值）标准化残差，记出现的时刻为t=s，构造矩阵dt。

Step 2

采用扩展的GARCH模型（将dt作为外生变量加入模型）拟合，这就得到了新的对数似然函数值lm。

如果2(lm-lb)<C，则终止：不再有异常值出现。这里的C=5.66+1.88log(T)，T是观察值的数量（在本例中为3015）。

代码实现如下

```
### Outlier Detection in GARCH(1,1) by Doornik & Ooms 2002
#Preparation
T<- 3015 #length of data
Ct<- 5.66+1.88*log10(T)
specgarch0 <- ugarchspec()
mod0<- ugarchfit()
lb<-c()
mod0.resSt<-c()
mod0.res.abs<-c()
a<-c()
dt<-matrix()
dt1<-matrix()
specgarch<-ugarchspec()
mod<-ugarchfit()
lm<-c()
C<-c()
```

```
mod.resSt<-c()
mod.res.abs<-c()
loc<-matrix()
outliers<-matrix()
critval<-c("FALSE")
no<-c()
k<-c()
outliers<-c()
```

```
### Outlier Detection in GARCH(1,1) by Doornik & Ooms 2002
## Step 1
# Estimate baseline GARCH model to obtain log-likelihood and residuals

mod0<- ugarchfit(data=RV.xts, spec=best_spec,solver = 'hybrid')
lb<-likelihood(mod0)

## Step 2
# Find largest absolute standardized residual
mod0.resSt<-residuals(mod0, standardize=TRUE)
mod0.res.abs<-abs(mod0.resSt)
a<-which.max(mod0.res.abs)

# Estimate the extended GARCH model with dummy dt in mean and dt-1 in variance
# Dummies
dt<-matrix(0,T)
dt[a]<-1
dt1<-matrix(0,T)
dt1[a-1]<-1
# Extended GARCH model
# If C < Ct then terminate: no further outliers are present!
while (critval == "FALSE") {
  specgarch <- ugarchspec(variance.model=list(model="eGARCH", garchOrder=c(1,1),
external.regressors= dt1), mean.model=list(armaOrder=c(0,1), external.regressors=
dt), distribution="std")
  mod<- ugarchfit(data=RV.xts, solver = 'hybrid', spec=specgarch)
  lm<-likelihood(mod)
  C<- 2*(lm-lb)
  mod.resSt<-residuals(mod, standardize=TRUE)
  mod.res.abs<-abs(mod.resSt)
  a<-which.max(mod.res.abs)
  dt[a]<-1
  dt1[a-1]<-1
  critval<- C < Ct}

outliers<-cbind(RV.xts[which(dt==1)])

> print(outliers)
                 [,1]
2012-01-13 -0.1955444
2019-10-24  0.1744222
```

求得的异常值为`2012-01-13`日与`2019-10-24`日的数据，我们通过将其赋值为`NA`，再利用`na.omit`函数剔除的方法获得了去掉异常值的数据，并对其进行重新拟合。

```
dates <- as.Date(c("2012-01-13","2019-10-24"))
RV.xts[dates] <- NA
RV.xts<-na.omit(RV.xts)

my_best_garch_outlier <- ugarchfit(spec = best_spec, data = RV.xts,solver =
'hybrid')
```

```
> rugarch::infocriteria(my_best_garch_outlier)[1]
[1] -4.114612
> rugarch::infocriteria(my_best_garch_outlier)[2]
[1] -4.100647
```

可以看到，剔除异常值后的拟合模型的AIC与BIC均有所下降，说明我们对异常值的处理是有意义的。

model name	AIC	BIC
my_best_garch	-4.103858	-4.089901
my_best_garch_outlier	-4.114612	-4.100647

参考文献

[1]Doornik, Jurgen A., and Marius Ooms. *Outlier detection in GARCH models*. No. 05-092/4. Tinbergen Institute Discussion Paper, 2005.

[2]Perlin, Marcelo Scherer, et al. "A garch tutorial with r." *Revista de Administração Contemporânea* 25 (2020).