目录

1 前言

S&P500 指数是记录美国 500 家上市公司的一个股票指数, 与道琼斯指数相比, S&P500 指数包含的公司更多, 因此风险更为分散, 能够反映更广泛的市场变化, 被普遍认为是一种理想的股票指数期货合约的标的. 自 2007 年美国发生金融危机以来, 世界经济秩序都受到了冲击, S&P500 指数更是在 2009 年 3 月 9 日跌至 672.88 点, 随着美国实施启动量化宽松政策、下调利率, 之后的十一年, S&P500 呈稳定的上升趋势并在 2020 年 2 月 19 日达到历史最高点 3393.52 点, 然而之后便在 3 月 9 日、12 日、16 日、18 日经历了惊人的四次熔断, 与最高点时相比接近下降了 1000 点. 在历史的波动点上, 金融预测便更具有显著地实践指导意义, 这也是我们的研究意义所在. 本文选用 2008 年至 2020 年 S&P500 的日收盘价作为研究数据, 采用 ARIMA、ARMA+GARCH、指数相乘模型作为研究工具, 预测 S&P500 指数的变化规律.

2 ARIMA 介绍

ARIMA 模型是最著名的时间序列预测分析方法之一, 全称为差分自回归移动平均模型 (Autoregressive Integrated Moving Average Model, 简记 ARIMA), 于 70 年代初由博克思 (Box) 和詹金斯 (Jenkins) 提出, 所以又称为 box-jenkins 模型、博克思-詹金斯法. ARIMA 模型一般表示为 ARIMA (p, d, q) , 其中 AR 是自回归, p 为自回归项; MA 为滑动平均, q 为滑动平均项数, d 为时间序列成为平稳序列时所做的差分次数.

ARIMA 模型结合了 AR 和 MA 模型的思想, 其一般形式可以表达为

$$(1-\sum_{i=1}^{p}\phi_i L^i)(1-L)^d X_t = (1+\sum_{i=1}^{q}\theta_i L^i)\varepsilon_t, d>0, d\in\mathbb{Z}$$

为了使模型有意义, 一般要求模型中 $\phi_i \neq \theta_i$, 否则模型将退化为一个白噪声序列.

ARIMA 模型是建立在时间序列具有平稳性和方差齐性的基础上的, 使用 ARIMA 模型预测的一般步骤为：

(一) 根据时间序列的散点图、自相关函数和偏自相关函数图检验其方差、趋势及其季节性变化规律, 对序列的平稳性进行识别. 通常来讲, 经济运行的时间序列都不是平稳序列.

(二) 对非平稳序列进行平稳化处理. 如果数据序列是非平稳的, 并存在一定的增长或下降趋势, 则需要使用平滑法或差分法对原始数据进行处理；如果数据存在异方差, 则需对数据进行技术处理, 直到处理后的数据的自相关函数值和偏相关函数值无显著地异于零.

(三) 根据时间序列模型的识别规则, 建立相应的模型. 若平稳序列的偏相关函数是截尾的, 而自相关函数是拖尾的, 可断定序列适合 AR 模型；若平稳序列的偏相关函数是拖尾的, 而自相关函数是截尾的, 则可断定序列适合 MA 模型；若平稳序列的偏相关函数和自相关函数均是拖尾的, 则序列适合 ARMA 模型.

(四) 进行参数估计, 检验是否具有统计意义.

(五) 进行假设检验, 诊断残差序列是否为白噪声.

(六) 利用已通过检验的模型进行预测分析.

在此我们还考虑引入季节性变化对模型的影响. 季节性模型在非季节 ARIMA 模型的基础上加上季节自回归 (SAR)、季节滑动平均 (SMA) 和季节差分 (D), 记为 ARIMA(p,d,q)(P,D,Q). 其表达式分为两部分：

回归部分误差结构方程

$$(1-\phi_1 L-\phi_2 L^2-...-\phi_P L^P)(1-\varphi_1 L^{-1}-...-\varphi_P L^{-1})\mu_t=\varepsilon_t$$

滑动平均部分误差结构方程

$$\mu_t=(1+\theta_1 L+\theta_2 L^2+...+\theta_P L^P)(1+\omega_1 L^{-1}+...+\omega_Q L^{-1})\varepsilon_t$$

两部分合并即成为季节模型的一般表达式, 其中 L 为滞后算子, $L^{-1}\mu_t=\mu_{t-1}$.

3 原始数据分析

首先, 画出 $S\&P500$ 的走势图, 如图 1.

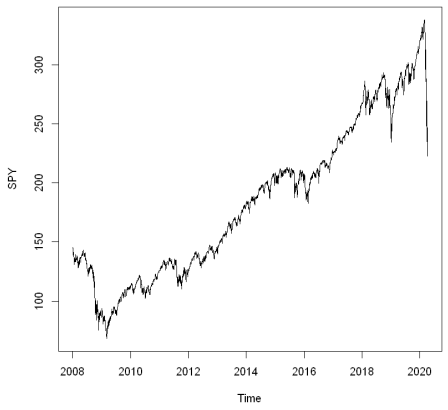

图 1: 08 年后 S&P500 的指数走势.

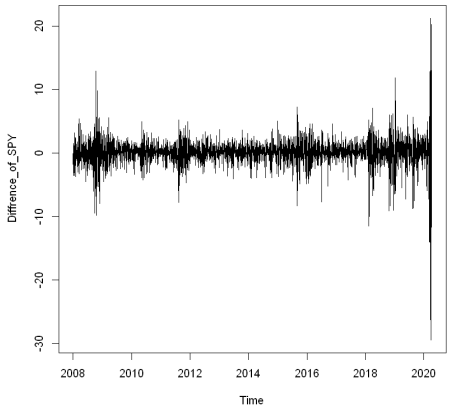

图 2: S&P500 指数差分序列的图示.

从图 1 中, 可以明显的看出它有一定的趋势, 因而并非平稳序列. 故考虑它的一阶差分序列, 将其画出, 如图 2. 从中可以看出, 差分序列基本都是围绕着 0 上下波动的, 因此, 可以看作是弱平稳序列. 于是, 接下来对于一阶差分序列进行分析.

4 初步建模

首先对于差分序列实行 ADF 单位根检验, 结果如下

Dickey-Fuller = -14.196, Lag order = 14, p-value = 0.01.

由此, 可以判断一阶差分序列是稳定的. 因此, 可以进行下一步分析.

4.1 定阶

之后需要确定 ARIMA 模型的阶数, 首先作出 ACF 和 PACF 图, 如图 3 4 所示, 虚线表示的是两倍标准误差的上下界.

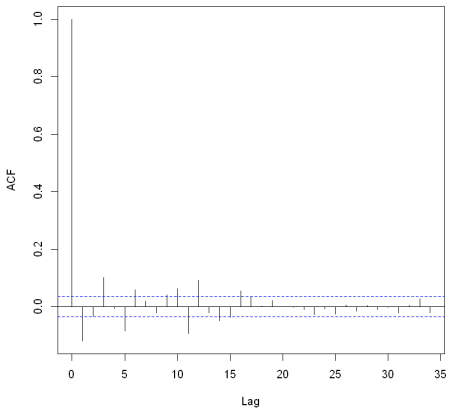

图 3: 差分序列的 ACF 图.

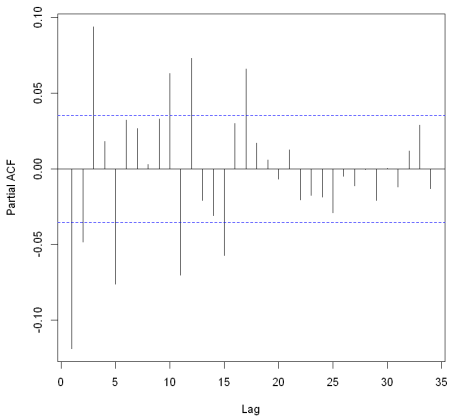

图 4: 差分序列的 $PACF$ 图.

可以看出, 两个图像都存在拖尾现象, $p = 4, q = 4$ 是一组可能的解, 但无法确定确切的最优的 (p,q) 的值, ACF 和 $PACF$ 为识别纯 $AR(p)$ 和 $MA(q)$ 模型提供了有效的工具. 但对于混合 $ARMA$ 模型来说, 它的理论 ACF 和 $PACF$ 有着无限多的非零值, 这使得根据样本 ACF 和 $PACF$ 来识别混合模型非常困难. 于是我们通过更适合大样本的 $EACF$ 来进一步确定.

表 1: EACF 结果

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	X	O	X	O	X	X	O	O	X	X	X	X	O	X
1	X	X	X	O	X	X	X	O	O	X	O	X	O	O
2	X	X	O	O	O	X	O	O	O	X	X	X	X	O
3	X	X	X	O	O	O	O	O	X	O	O	X	O	O
4	X	X	X	X	O	O	O	X	O	O	O	X	X	O
5	X	X	X	X	X	O	X	X	O	O	O	X	O	X
6	X	X	X	X	X	X	O	X	O	O	O	X	O	O
7	X	X	X	O	X	X	X	O	X	X	O	X	X	O

由表1 可知, $p = 2, q = 3; p = 2, q = 6; p = 3q = 4; p = 3, q = 5$ 时的 ARMA 模型可能比较合适.

使用赤池信息准则 (AIC) 对于模型的参数进行筛选, 此准则要求选择使下式最小化的模型

$$AIC = -2\log(MLE) + 2k.$$

通过 R 自带的 $auto.arima()$ 函数 (此函数以 AIC 为准则) 得到的推荐参数选择为 $p = 5, q = 0$. 另外, 还可以使用 Schwarz 贝叶斯信息准则 (BIC) 最来筛选参数, BIC 定义如下的

$$BIC = -2log(MLE) + klog(n).$$

在此准则下各参数的 BIC 如图 5 所示.

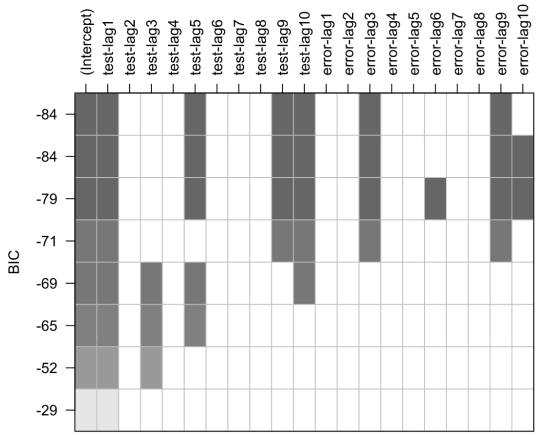

图 5: 最优子集结果.

从图中可以出, 应该使用 $Y_{(t-1)}, Y_{(t-5)}, Y_{(t-9)}, Y_{(t-10)}$ 来建模, 误差部分需要三阶和九阶滞后项. 但这样的话相关项会过多, 且结合 ACF 与 PACF 图, 发现在滞后项 1、5、10 上均比较大, 转而考虑是否存在季节性因素, 即将一周开盘的 5 天视为一个季节, 进行研究.

首先对价格隔五项进行差分, 画出 $ACF, PACF$ 的图像, 如图 6 7.

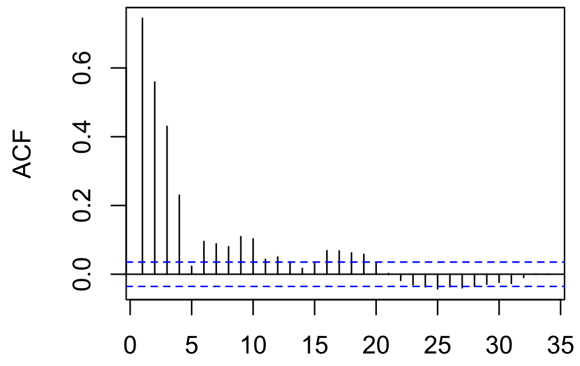

图 6: 差分序列的 ACF 图.

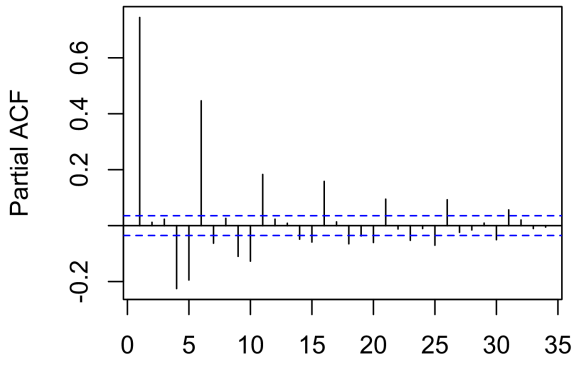

图 7: 差分序列的 $PACF$ 图.

结合 ACF 和 $PACF$ 图判断选择模型即可，为了排除偏差, 使用 $P = 1, P = 2$ 都进行了实验，得到的 AIC 信息准则结果如表 2.

表 2: 各种模型的 AIC 值

p	4	4	5	5	2	2	3	3	3	3	2	2
d	1	1	1	1	1	1	1	1	1	1	1	1
q	4	4	0	0	3	3	4	4	5	5	6	6
P	1	2	1	2	1	2	1	2	1	2	1	2
D	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0
AIC 值 (13000+)	502	497	506	504	512	515	498	494	481	483	494	511

比较 AIC 后, 选择最小的 $ARIMA(3,1,5)(1,0,0)[5]$ 进行建模.

4.2 模型建立

使用上文的 $ARIMA(3,1,5)(1,0,0)[5]$ 进行建模, 得到的模型为

$$(Y_t + 0.1151Y_{t-1} + 0.0337Y_{t-2} - 0.0943Y_{t-3})\,(Y_t + 0.0708Y_{t-1} - 0.0625Y_{t-2}) = \varepsilon_t,\ \varepsilon_t \sim N(0,1).$$

并且画出模型的单位根的分布, 如图 8.

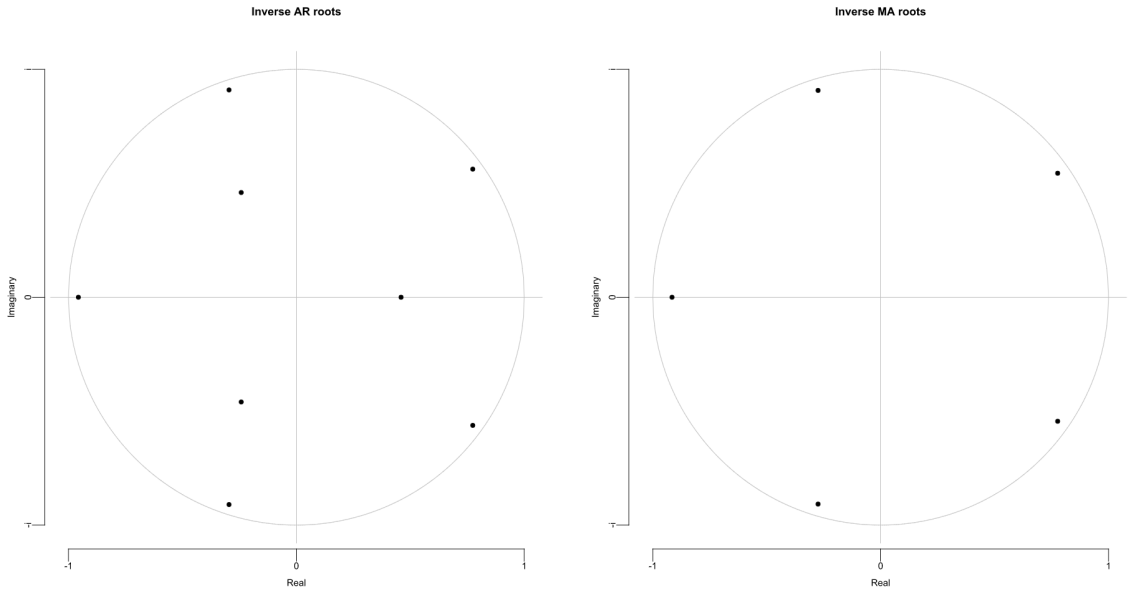

图 8: 模型的单位根分布.

同时, 画出残差的分布, 如图 9.

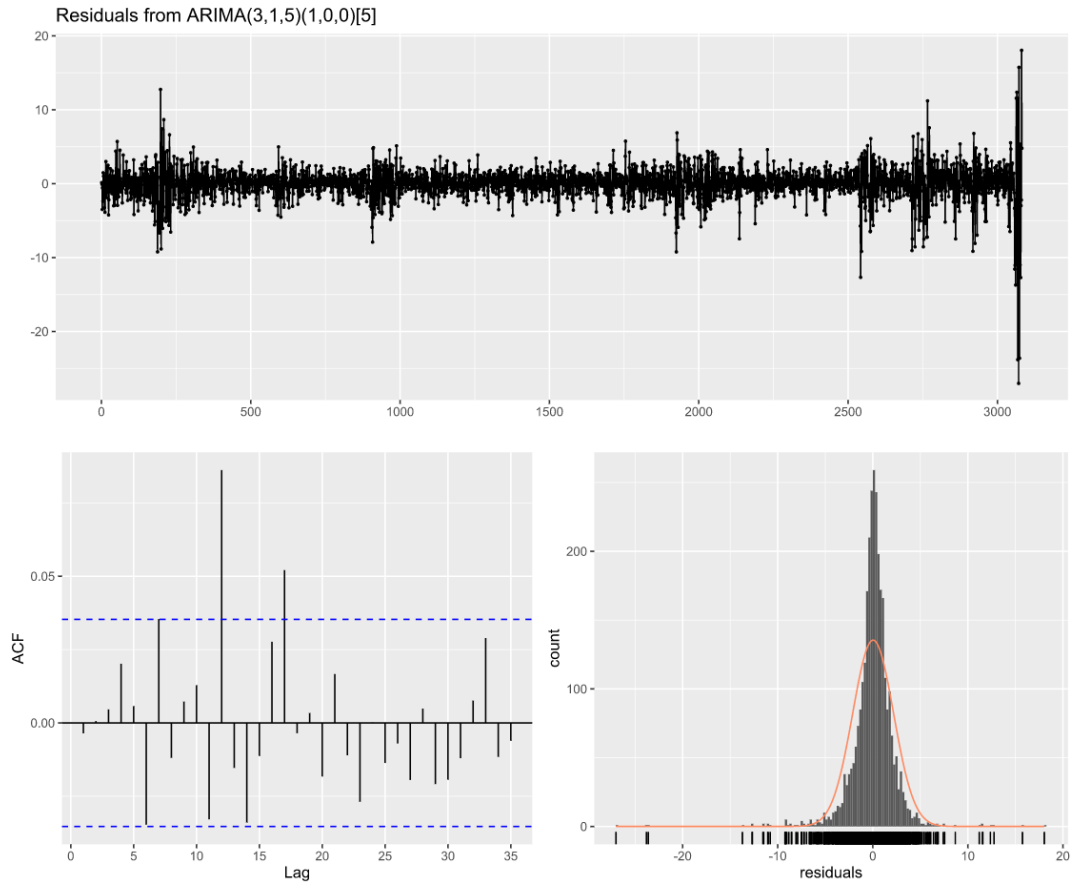

图 9: 残差的分布.

可以看到, 单位根都在单位圆内, 同时, 对与残差进行 Ljung-Box 检验, 结果如下

$$\text{Q}^* = 36.489,\ \text{df} = 3,\ \text{p-value} = 5.902\text{e-}08.$$

p 值接近 0, 因此, 残差的正态性相当好, 但通过残差的 ACF 图可以看出模型仍不是十分理想. 接下来使用此模型对于未来的走势进行预测, 预测结果如图 10.

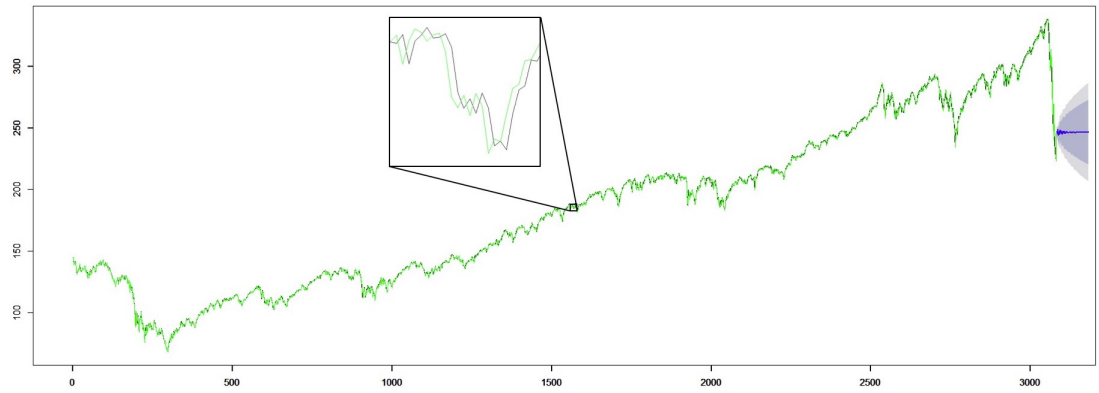

图 10: 模型预测结果.

5 剔除季节因素

5.1 拟合纯季节模型

价格可能会受到季节性的影响, 并且数据为股票市场, 因此可以合理地假设这个周期现象为周模式, 因为一周有五个工作日, 故先对于价格差分序列拟合一个纯季节模型. 其中, 先计算了 $P = 1, 2, 3$ 时的 AIC, 如表 3.

表 3: 纯季节模型的 AIC.

P	1	2	3
AIC	13588.65	13580.55	13579.58

由此, 选择 $P = 3$, 得到的模型 b_t 如下

$$b_t = (1 + 0.0790L^5 - 0.0651L^{10} + 0.0373L^{15})Y_t.$$

同样, 画出它的单位根的分布与残差的图像, 如图 11 12.

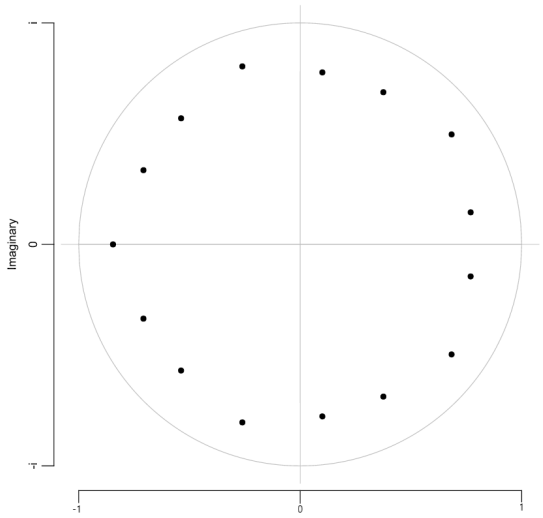

图 11: 单位根的分布.

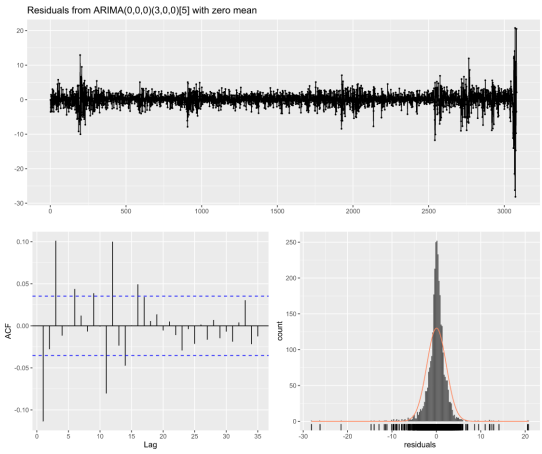

图 12: 模型残差的分布.

可以看到单位根均在单位圆之内, 因此是平稳的. 但从 ACF 序列中也可以看出纯季节性的模型并不理想.

5.2 剔除季节性影响

之后, 对于 Y_t, 结合春季节性模型进行调整

$$Y_t^* = Y_t + 0.0790Y_{t-5} - 0.0651Y_{t-10} + 0.0373L_{t-15}, t = 16, 17, \cdots, 3071.$$

首先对调整后的序列做分析, 画出其自相关与偏自相关图像如图 13 14.

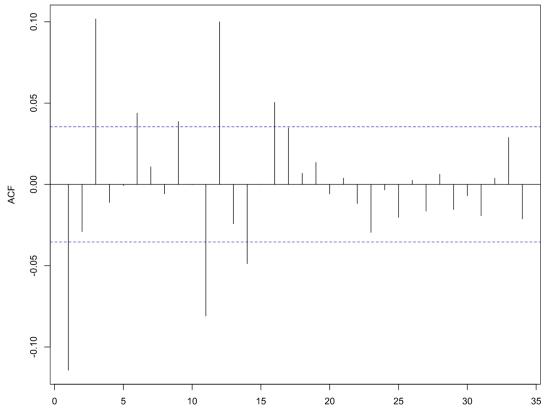

图 13: 调整后的差分序列 ACF 图.

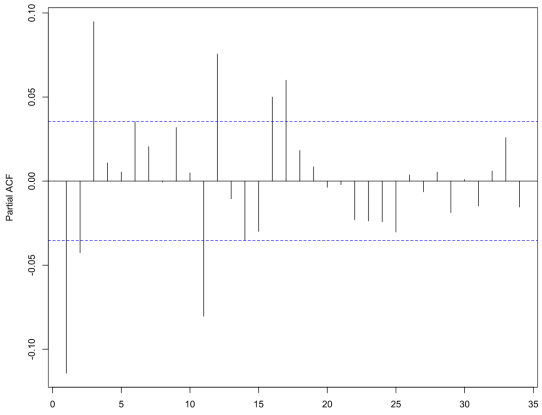

图 14: 调整后的差分序列 $PACF$ 图.

可以看出, 相比于原来的 ACF 与 $PACF$ 图像, 调整后的价格在 5,10 阶上已有大大的下降, 因此, 可以说我们较为成功地剔除了季节性的影响.

6 结合 GARCH 模型

GARCH 模型允许波动率随着时间进行变化, 并且允许波动率的聚集. 即在随机项 ϵ_t 前的系数并非像 ARMA 一样为一个常数, 而是一个与时间相关的函数 σ_t. 因此, 我们需要对这个函数进行建模. GARCH 对这个系数的建模是十分类似于 ARMA 模型对于价格的建模的. 如果对数收益表示为 $r_t = \mu_t + a_t = \mu_t + \sigma_t\epsilon_t$, 那么 GARCH(1,1) 则为

$$\sigma_t^2 = \alpha_0 + \alpha_1\alpha_{t-1}^2 + \beta_1\sigma_{t-1}^2$$

因为这样的波动率从确定的值变成了一个函数, 就可以用来模拟市场大起大落扎堆的现象, 能够对市场的变化响应得比 ARMA 模型更灵敏.

我们猜想数据会受到多方面的外部因素的扰动, 所以对较长时间的数据分析十分的困难. 从数据分析中, 可以看出我们的差分序列波动的最大程度既是价格上涨与下跌最明显的时候. ARMA 模型无法处理这样的波动现象. 于是结合了 GARCH 模型来解释数据, 提高模型的预测能力.

6.1 使用 GARCH 初次建模

首先使用 ARMA(3,0)+GARCH(1,1) 对于调整后的价格序列进行建模. 拟合的模型为

$$Y_t^* = -0.0428Y_{t-1}^* - 0.0062Y_{t-2}^* - 0.0104Y_{t-3}^* + a_t, \quad a_t = \sigma_t\varepsilon_t, \quad \varepsilon_t \sim N(0,1)$$
$$\sigma_t^2 = 0.1035 + 0.1449a_{t-1}^2 + 0.8288\sigma_{t-1}^2$$

同时还发现 $ARMA$ 的第一个系数比较显著.

为了进一步的改进模型, 使用残差的相关图像对模型进行了可视化. 结果如图 15 16.

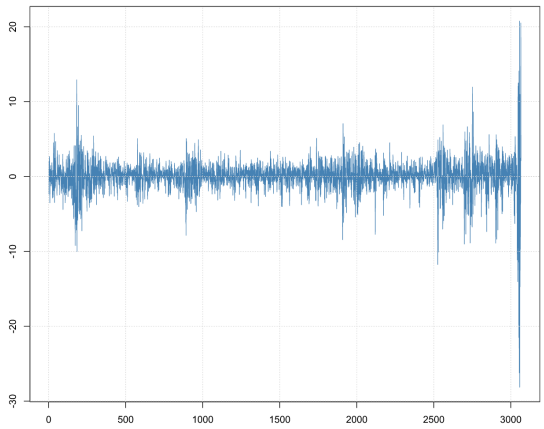

图 15: 模型所拟合的时间序列.

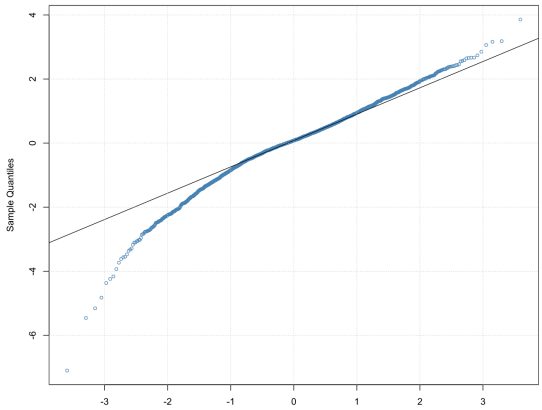

图 16: 残差值与正态分布的 QQ 图.

可以明显地看到, 标准化残差的正态性并不是十分的好, 因此模型还需进一步的调整.

6.2 新息分布更换

改进的模型与上一个模型基本相同, 但是考虑了不同的新息分布. 采用了 t 分布作为新的新息分布. 模型如下

$$Y_t^* = -0.0395Y_{t-1}^* - 0.0047Y_{t-2}^* - 0.0035Y_{t-3}^* + a_t, \quad a_t = \sigma_t\varepsilon_t, \quad \varepsilon_t \sim N(0,1)$$
$$\sigma_t^2 = 0.0739 + 0.1349a_{t-1}^2 + 0.8516\sigma_{t-1}^2$$

从结果可以看到, 依旧只有 $ARMA$ 的第一个系数比较显著, 同时, 对新息的假设也通过了 $L-B$ 检验. 同样的, 我们对于残差的分布进行可视化. 如图 17 18.

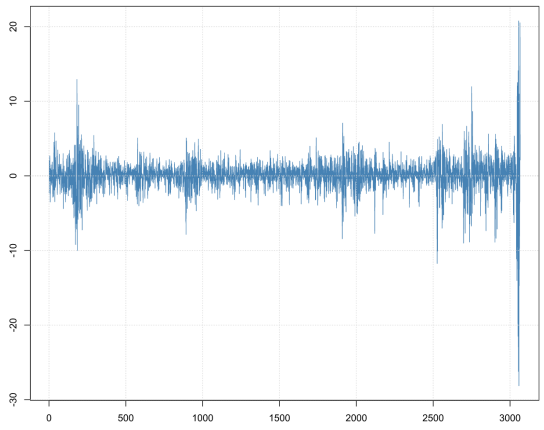

图 17: 模型所拟合的时间序列.

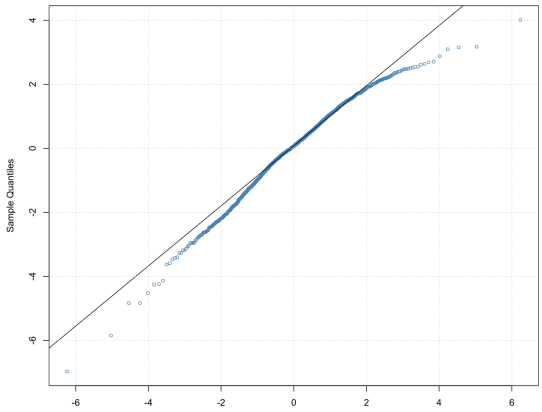

图 18: 残差值与正态分布的 QQ 图.

从图中可以发现, 此时模型的效果反而变差了. 标准化残差的分布依旧左偏, 并且左偏得更加严重. 因此, 需要更新对于新息分布的假设. 自然而然的就考虑了能够处理这样情况的偏 t 分布.

6.3 缩减 $ARMA$ 的系数个数

由于前两次的模型除了 MA 的第五个系数并不显著性. 因此, 考虑将模型更换到 $ARMA(3,4)+GARCH(1,1)$. 并且采用偏 t 分布作为新息分布. 拟合的模型如下

$$Y_t^* = -0.0631Y_{t-1}^* + a_t, \quad a_t = \sigma_t\varepsilon_t, \quad \varepsilon_t \sim N(0,1)$$
$$\sigma_t^2 = 0.0729 + 0.1329a_{t-1}^2 + 0.8554\sigma_{t-1}^2$$

在这个模型中, $ARMA$ 的七个系数的显著性全部十分的高. 残差的分布可视化如图 19 20.

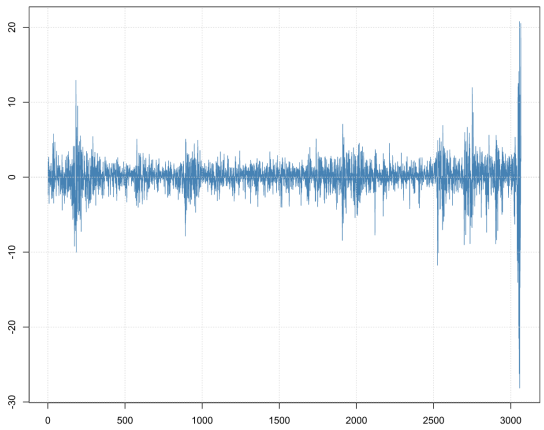

图 19: 模型所拟合的时间序列.

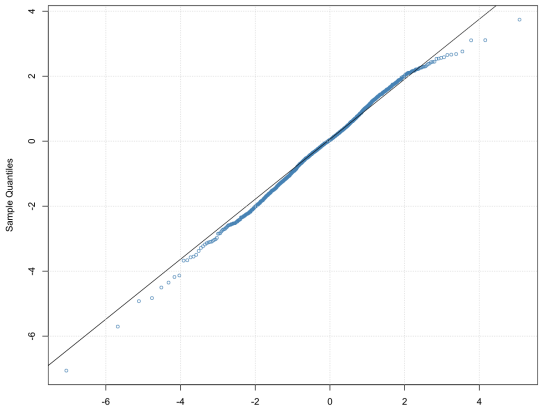

图 20: 残差值与正态分布的 QQ 图.

可以看到, 绝大部分点在该图中都表现出了一条直线的特征. 故可以认为偏 t 分布的假设对于新息而言是十分合理的, 虽然还是有轻微的左偏性.

接下来, 对超前两步的数值进行预测, 结果如下

meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
-2.932209	12.40724	12.40724	-29.91137	19.32762
1.069844	12.36999	12.34532	-25.82832	23.26285

可以看到, 模型预测的 95% 置信区间基本上处于调整后差分序列两个极值附近. 因为采用的是偏 t 分布, 所以区间并未呈现对称性. 反而是差分为正的置信区间更短, 意味着估计的确定性更高, 也就意味着我们对于未来的市场应该保持一个比较积极乐观的态度.

7 指数模型

指数模型是用来预测时序未来值时最常用的一种模型. 经过观察可以发现 2008 ~ 2019 年的 S&P500 指数整体呈上升趋势, 因此可以使用指数模型对其进行拟合.

不同指数模型在建模过程中选用的因子不同. 如单指数模型 (simple exponential model) 拟合的只是常数水平项和时间点 i 处随机项的时间序列, 即认为时间序列不存在趋势项和季节效应; 双指数模型

(double exponential model) 也叫 Holt 指数平滑, 拟合的是有水平项和趋势项的时序; 三指数模型 (triple exponential model) 拟合的是有水平项、趋势项以及季节效应的时序.

R 中的 ets() 函数有自动选取对原始数据拟合优度最高的模型的功能, 因此, 我们使用拟合优度最高的方式, 获得如图 21 所示的结果.

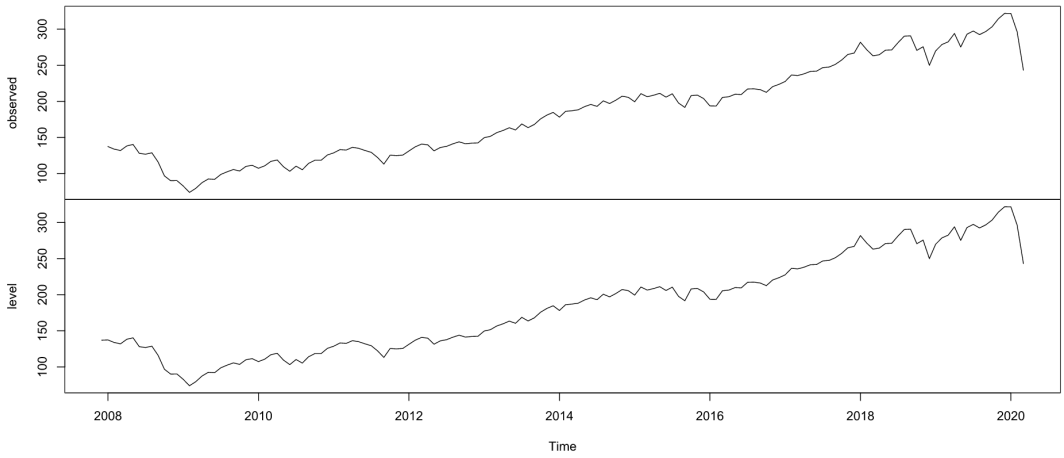

图 21: 指数模型的结果.

由于 ets() 的预测 frequency 的最大选择为 24, 所以原数据直接采用了每月的 S&P500 指数. 可以看到, 最终结果为 $ETS(M,N,N), \alpha = 0.99$, 即代表了单指数相乘模型. 而 α 参数控制权重下降的速度, α 越接近 1, 近期观测值的权重越大; 反之, 越接近于 0, 则历史观测值的权重越大. 也就是

$$Y_t = level + irregular_r.$$

单指数平滑根据现有的时序值的加权平均对未来做短期预测, 图 22 给出了其折线图和以下八个季度的预测.

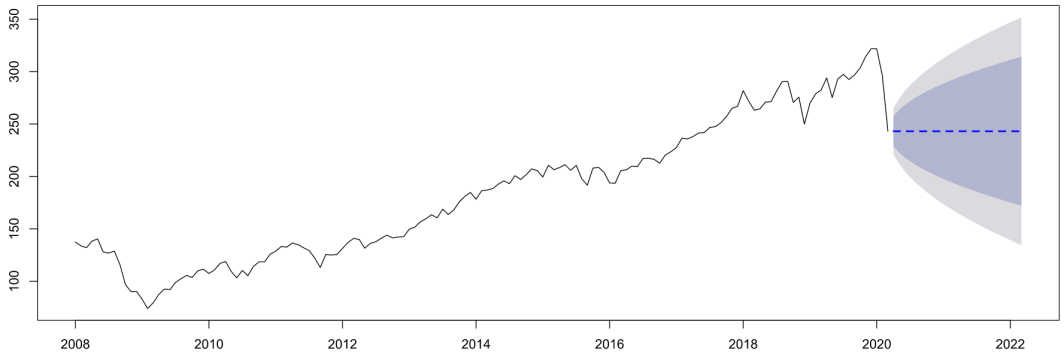

图 22: 可乘的单指数光滑预测, 其中预测值由虚线表示, 80% 和 95% 置信区间分别由淡灰色和深灰色表示.

同时, 模型的各个度量标准如下

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.7217255	8.607849	5.926743	0.2795169	3.445388	0.2628449	0.0787231

一般来说由于平均误差和平均百分比会正向负向相抵消, 所以用处不大, RMSE 给出了平均误差平方和平方根, 由于单指数模型相对简单易行, 即这样的误差水平还算令人满意.

8 结语

本文中我们采用了 ARIMA、ARMA+GARCH、指数相乘三个模型对 S&P500 进行研究, 在 ARIMA 与 ARMA+GARCH 中我们都试图通过去除季节性因素使预测更为准确, 并且对残差进行了正态性分析. 而对于新息的预测都是偏向积极, 即目前市场应该会慢慢回温. 在实验过程中, 我们也运用了一些自己并不是很熟悉的知识, 可能会存在使用理解有差或者使用不当的情况, 但这也是我们学习与探索的过程.

参考文献

[1] Ruey S.Tsay. 金融时间序列分析. 人民邮电出版社, 2009.

[2] Ruey S.Tsay. 多元时间序列分析及金融应用: *R* 语言. 机械工业出版社, 2016.

[3] 金融时间序列分析:3. first demo by python. https://blog.csdn.net/matrix_laboratory/article/details/53790549.

[4] 写给你的金融时间序列分析：补完篇. https://zhuanlan.zhihu.com/p/77307871.

[5] 写给你的金融时间序列分析：初级篇. https://zhuanlan.zhihu.com/p/38321845.

[6] 金融时间序列分析：6. ar 模型实例（r 语言）. https://blog.csdn.net/matrix_laboratory/article/details/53924972.

A 完整代码

[1]:
```
# 加载所要用到的包
library(fGarch)
library(tseries)
library(TSA)
library(forecast)
library(scales)
library(plyr)
library(TTR)
```

[2]:
```
# 对原始数据进行分析
Price <- read.csv("./08 年每日.csv")$收盘
ts <- ts(Price, start = c(2008, 1), frequency = 251)
plot(ts)
```

[3]:
```
# 对差分序列进行分析
dPricets <- diff(Price, 1)
plot(dPricets)
adf.test(dPricets)
acf(dPricets)
pacf(dPricets)
eacf(dPricets)
res = armasubsets(y = dPricets, nar = 10, nma = 10, y.name = "test", ar.method = "ols")
plot(res)
```

```
Augmented Dickey-Fuller Test

data:  dPricets
Dickey-Fuller = -14.196, Lag order = 14, p-value = 0.01
alternative hypothesis: stationary
```

[4]:
```
# 对季节差分序列进行分析
seasonprice <- diff(Price, 5)
acf(seasonprice)
pacf(seasonprice)
```

[5]:
```
# 构建首个模型并画出图像
m1 <- Arima(Price, order = c(3, 1, 5), seasonal = list(order = c(1, 0, 0), period = 5))
summary(m1)
plot(m1)
```

```r
checkresiduals(m1)
plot(forecast(m1, h = 100))
pre = forecast(m1, 100)
plot(m1$fitted)
par(new = T)
plot(pre, col = "green", xlim = c(0, 3080))
plot(m1$fitted, xlim = c(1500, 1550), ylim = c(170, 190))
par(new = T)
plot(pre, col = "green", xlim = c(1500, 1550), ylim = c(170, 190))
```

```
Series: Price
ARIMA(3,1,5)(1,0,0)[5]

Coefficients:
          ar1      ar2     ar3      ma1     ma2      ma3      ma4     ma5
      -0.0248  -0.0474  0.1240  -0.0887  0.0262  -0.0374  -0.0176  0.7409
s.e.   0.0273   0.0251  0.0253   0.0206  0.0179   0.0182   0.0134  0.0488
         sar1
      -0.8014
s.e.   0.0469

sigma^2 estimated as 4.648:  log likelihood=-6734.43
AIC=13488.87   AICc=13488.94   BIC=13549.2

Training set error measures:
                     ME     RMSE      MAE         MPE      MAPE     MASE
Training set 0.03472924 2.152525 1.345183 0.009393096 0.8098321 1.010023
                     ACF1
Training set -0.003542397


        Ljung-Box test

data:  Residuals from ARIMA(3,1,5)(1,0,0)[5]
Q* = 36.489, df = 3, p-value = 5.902e-08

Model df: 9.   Total lags used: 12
```

[6]:
```r
# 拟合纯季节模型
m2 <- Arima(dPricets, seasonal = list(order = c(3, 0, 0), period = 5), include.mean = F)
summary(m2)
plot(m2)
checkresiduals(m2)
```

```
Series: dPricets
ARIMA(0,0,0)(3,0,0)[5] with zero mean

Coefficients:
         sar1    sar2     sar3
      -0.0790  0.0651  -0.0373
s.e.   0.0184  0.0191   0.0209

sigma^2 estimated as 4.801:  log likelihood=-6787.06
AIC=13582.12   AICc=13582.14   BIC=13606.26

Training set error measures:
                     ME     RMSE      MAE MPE MAPE      MASE       ACF1
Training set 0.03473708 2.190042 1.339692 NaN  Inf 0.6734637 -0.1135568


        Ljung-Box test

data:  Residuals from ARIMA(0,0,0)(3,0,0)[5] with zero mean
Q* = 85.257, df = 7, p-value = 1.11e-15

Model df: 3.   Total lags used: 10
```

[7]:
```
# 剔除季节性影响
adjp <- dPricets[16:3081] + 0.079 * dPricets[11:3076] - 0.0651 * dPricets[6:3071] +
    0.0373 * dPricets[1:3066]
acf(adjp)
pacf(adjp)
```

[8]:
```
# 加入 GARCH 模型
m3 <- garchFit(~arma(3, 5) + garch(1, 1), data = adjp, trace = F, include.mean = F)
summary(m3)
plot(m3, which = 1)
plot(m3, which = 13)
```

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(3, 5) + garch(1, 1), data = adjp, include.mean = F,
    trace = F)

Mean and Variance Equation:
 data ~ arma(3, 5) + garch(1, 1)
<environment: 0x000000002a7d7f38>
 [data = adjp]

Conditional Distribution:
 norm

Coefficient(s):
       ar1         ar2         ar3         ma1         ma2         ma3
 0.2183850   0.2987791  -0.3715975  -0.2618720  -0.2952942   0.3823865
       ma4         ma5       omega      alpha1       beta1
-0.0098208   0.0606401   0.1040603   0.1468127   0.8269387

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1     0.218385    0.218074    1.001   0.3166
ar2     0.298779    0.180504    1.655   0.0979 .
ar3    -0.371598    0.174067   -2.135   0.0328 *
ma1    -0.261872    0.218860   -1.197   0.2315
ma2    -0.295294    0.181369   -1.628   0.1035
ma3     0.382387    0.180603    2.117   0.0342 *
ma4    -0.009821    0.025452   -0.386   0.6996
ma5     0.060640    0.021181    2.863   0.0042 **
omega   0.104060    0.017073    6.095 1.09e-09 ***
alpha1  0.146813    0.014443   10.165  < 2e-16 ***
beta1   0.826939    0.015739   52.540  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -5766.344    normalized:  -1.880738

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  877.8818  0
 Shapiro-Wilk Test  R    W      0.9742828 0
 Ljung-Box Test     R    Q(10)  7.272825  0.6994619
 Ljung-Box Test     R    Q(15)  10.76221  0.7692618
 Ljung-Box Test     R    Q(20)  19.26747  0.5045065
 Ljung-Box Test     R^2  Q(10)  17.54842  0.0630759
 Ljung-Box Test     R^2  Q(15)  23.29068  0.07816009
 Ljung-Box Test     R^2  Q(20)  24.74355  0.211419
 LM Arch Test       R    TR^2   19.65619  0.07387571

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
```

```
3.768652 3.790280 3.768627 3.776423
```

[9]:
```
# 更换新息分布
m4 <- garchFit(~arma(3, 5) + garch(1, 1), data = adjp, trace = F, include.mean = F,
    cond.dist = "std")
summary(m4)
plot(m4, which = 1)
plot(m4, which = 13)
```

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(3, 5) + garch(1, 1), data = adjp, cond.dist = "std",
    include.mean = F, trace = F)

Mean and Variance Equation:
 data ~ arma(3, 5) + garch(1, 1)
<environment: 0x00000000277696f8>
 [data = adjp]

Conditional Distribution:
 std

Coefficient(s):
       ar1         ar2         ar3         ma1         ma2         ma3
 0.6076789  -0.7878460   0.6311511  -0.6460218   0.8145114  -0.6672652
       ma4         ma5       omega      alpha1       beta1       shape
 0.0638112   0.0075509   0.0782826   0.1403160   0.8464154   5.6295839

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1     0.607679    0.142061    4.278 1.89e-05 ***
ar2    -0.787846    0.060724  -12.974  < 2e-16 ***
ar3     0.631151    0.139193    4.534 5.78e-06 ***
ma1    -0.646022    0.142938   -4.520 6.20e-06 ***
ma2     0.814511    0.065073   12.517  < 2e-16 ***
ma3    -0.667265    0.142763   -4.674 2.95e-06 ***
ma4     0.063811    0.023718    2.690  0.00714 **
ma5     0.007551    0.021109    0.358  0.72056
omega   0.078283    0.018418    4.250 2.13e-05 ***
alpha1  0.140316    0.017177    8.169 2.22e-16 ***
beta1   0.846415    0.017253   49.058  < 2e-16 ***
shape   5.629584    0.611584    9.205  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -5683.127    normalized:  -1.853597

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  934.9298  0
 Shapiro-Wilk Test  R    W      0.9734008 0
 Ljung-Box Test     R    Q(10)  10.4691   0.4003451
 Ljung-Box Test     R    Q(15)  11.42327  0.7220529
 Ljung-Box Test     R    Q(20)  19.36198  0.4984213
 Ljung-Box Test     R^2  Q(10)  12.04808  0.2818518
 Ljung-Box Test     R^2  Q(15)  18.77382  0.2241668
 Ljung-Box Test     R^2  Q(20)  20.53837  0.4247363
 LM Arch Test       R    TR^2   14.94122  0.2446668

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
3.715021 3.738614 3.714990 3.723498
```

[10]:
```r
# 缩减系数个数
m5 <- garchFit(~arma(3, 4) + garch(1, 1), data = adjp, trace = F, include.mean = F,
    cond.dist = "sstd")
summary(m5)
plot(m5, which = 1)
plot(m5, which = 13)
predict(m5, n.ahead = 2, plot = TRUE)
```

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(3, 4) + garch(1, 1), data = adjp, cond.dist = "sstd",
    include.mean = F, trace = F)

Mean and Variance Equation:
 data ~ arma(3, 4) + garch(1, 1)
<environment: 0x00000000255ea038>
 [data = adjp]

Conditional Distribution:
 sstd

Coefficient(s):
      ar1        ar2        ar3        ma1        ma2        ma3        ma4
-1.000000  -1.000000  -0.658369   0.937063   0.890059   0.541669  -0.110916
    omega     alpha1      beta1       skew      shape
 0.072631   0.131563   0.858010   0.838907   5.806889

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1     -1.00000     0.11909   -8.397  < 2e-16 ***
ar2     -1.00000     0.07114  -14.056  < 2e-16 ***
ar3     -0.65837     0.11574   -5.688 1.28e-08 ***
ma1      0.93706     0.11981    7.821 5.33e-15 ***
ma2      0.89006     0.07455   11.939  < 2e-16 ***
ma3      0.54167     0.12108    4.473 7.70e-06 ***
ma4     -0.11092     0.01984   -5.591 2.25e-08 ***
omega    0.07263     0.01704    4.261 2.03e-05 ***
alpha1   0.13156     0.01585    8.300  < 2e-16 ***
beta1    0.85801     0.01578   54.377  < 2e-16 ***
skew     0.83891     0.02020   41.523  < 2e-16 ***
shape    5.80689     0.67391    8.617  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -5656.916    normalized:  -1.845048

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  1084.698  0
 Shapiro-Wilk Test  R    W      0.9699741 0
 Ljung-Box Test     R    Q(10)  16.63696  0.08279324
 Ljung-Box Test     R    Q(15)  17.12291  0.3115699
 Ljung-Box Test     R    Q(20)  25.54736  0.1812866
 Ljung-Box Test     R^2  Q(10)  15.00136  0.1320124
 Ljung-Box Test     R^2  Q(15)  21.68922  0.1162157
 Ljung-Box Test     R^2  Q(20)  23.81214  0.2506945
 LM Arch Test       R    TR^2   18.33108  0.1060013

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
3.697923 3.721516 3.697892 3.706400
```

meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
-2.932209	12.40724	12.40724	-29.91137	19.32762
1.069844	12.36999	12.34532	-25.82832	23.26285

[11]:
```r
# 指数模型
monthly_2008 <- read.csv("08 年每月.csv")
Monthly_Price <- monthly_2008$收盘
Monthly_Pricets <- ts(Monthly_Price, frequency = 12, start = c(2008, 1))
fit_ets <- ets(Monthly_Pricets)
summary(fit_ets)
plot(fit_ets)
plot(forecast(fit_ets), xlab = "Time", flty = 2)
accuracy(fit_ets)
```

```
ETS(M,N,N)

Call:
 ets(y = Monthly_Pricets)

  Smoothing parameters:
    alpha = 0.9999

  Initial states:
    l = 137.0724

  sigma:  0.046

     AIC     AICc      BIC
1345.616 1345.784 1354.587

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set 0.7217255 8.607849 5.926743 0.2795169 3.445388 0.2628449 0.0787231
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.7217255	8.607849	5.926743	0.2795169	3.445388	0.2628449	0.0787231