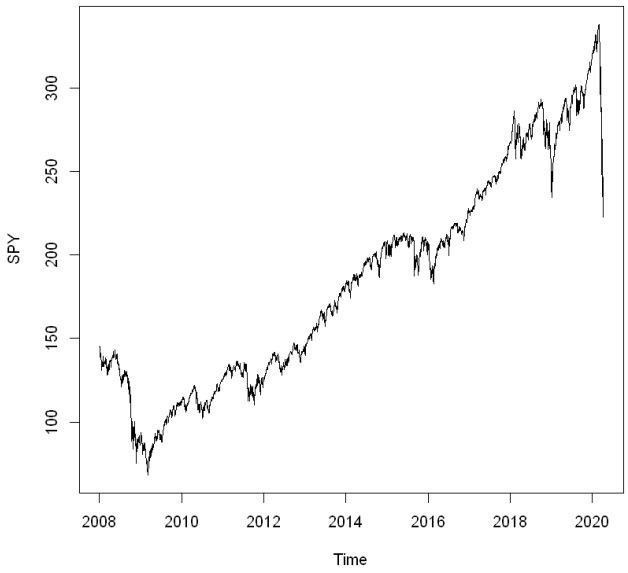
目录

1 目的描述

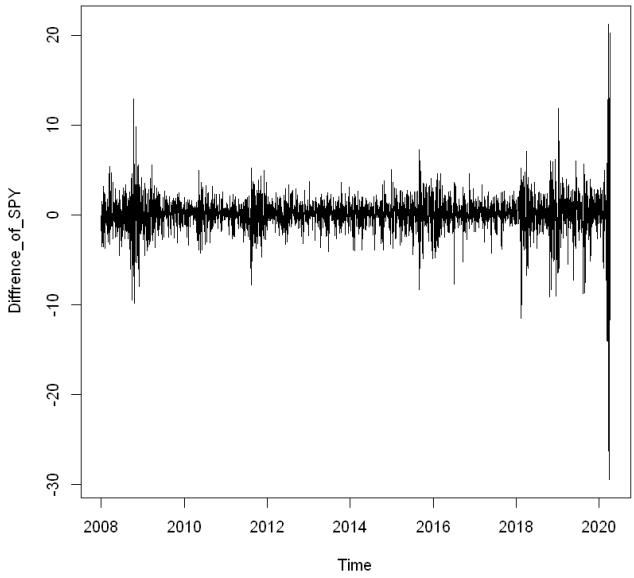
在本次研究报告中, 我们对于 08 年以后的 $S\&P500$ 的指数进行了分析, 并且使用课内以及课外的知识对其进行建模, 企图预测未来的 $S\&P500$ 指数走势.

2 原始数据分析

首先, 画出 $S\&P500$ 的走势图, 如下



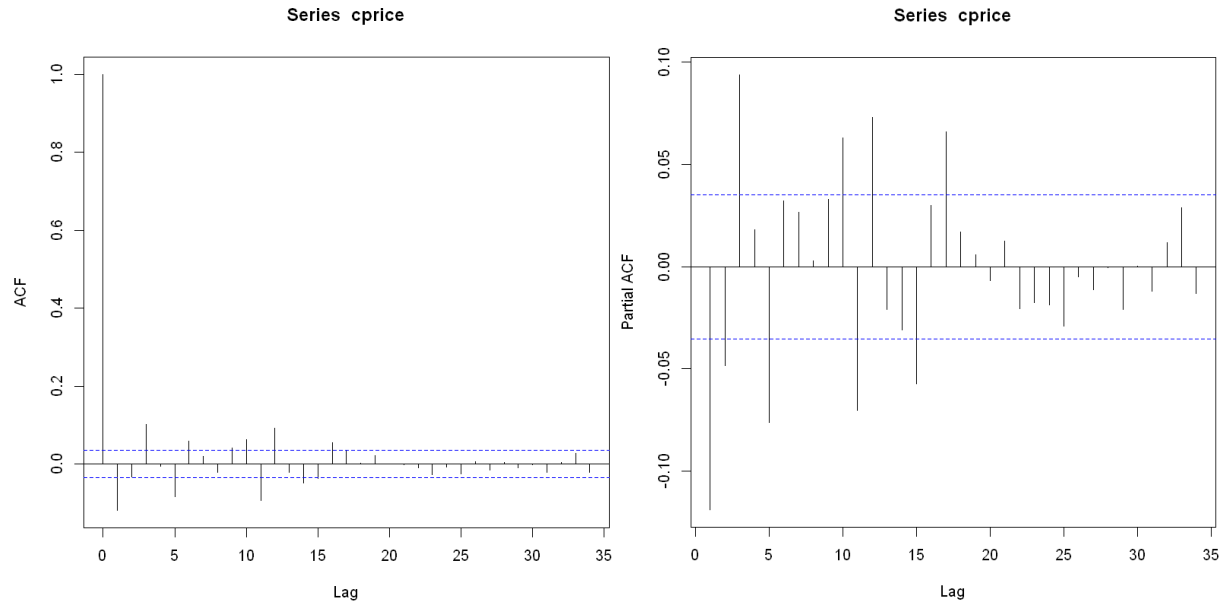
从中, 可以明显的看出它有一定的趋势, 因而并非平稳序列. 故考虑它的一阶差分序列, 将其画出, 如下图



从上图中可以看出, 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.

3 初步建模

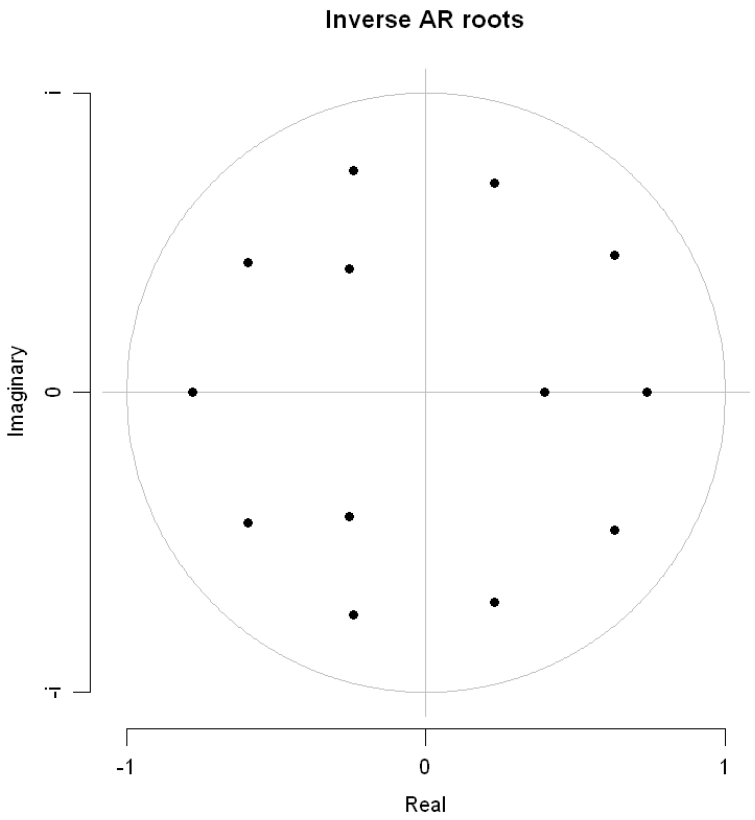
观察差分序列的 ACF 图像, 发现在 3 阶以后都近似在置信区间内, 因此先考虑使用 $AR(3)$ 模型对其近似.



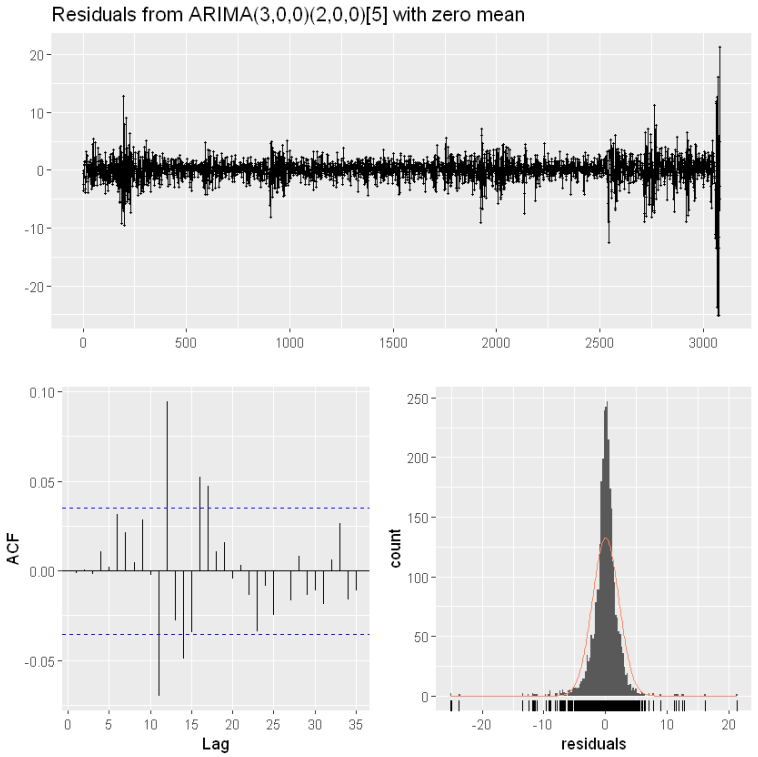
使用 $ARIMA$ 对其进行建模,

方程 AR(3)

并且画出相关图像.

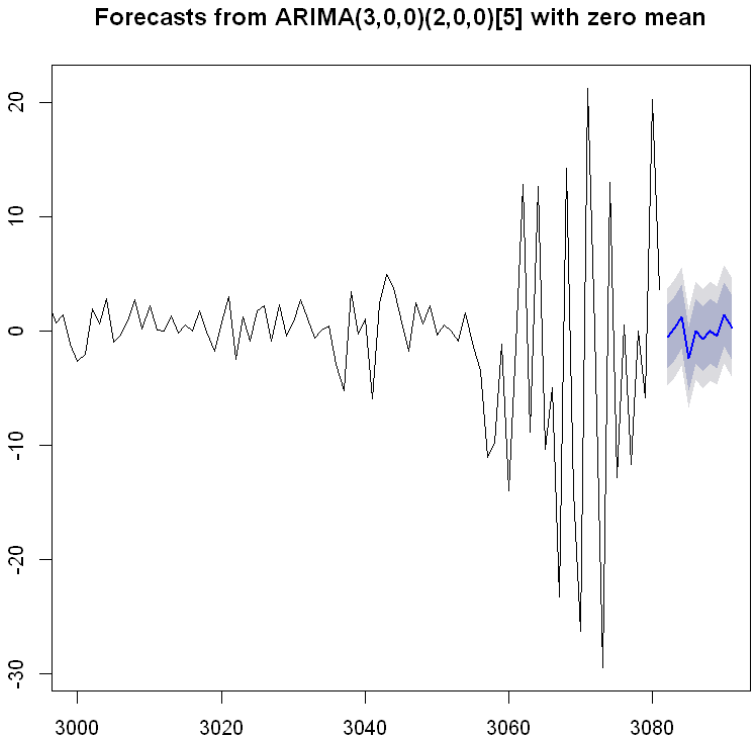


可以看到, 模型特征方程根的倒数都在单位圆, 说明是平稳的, 接下来再对其残差进行分析.



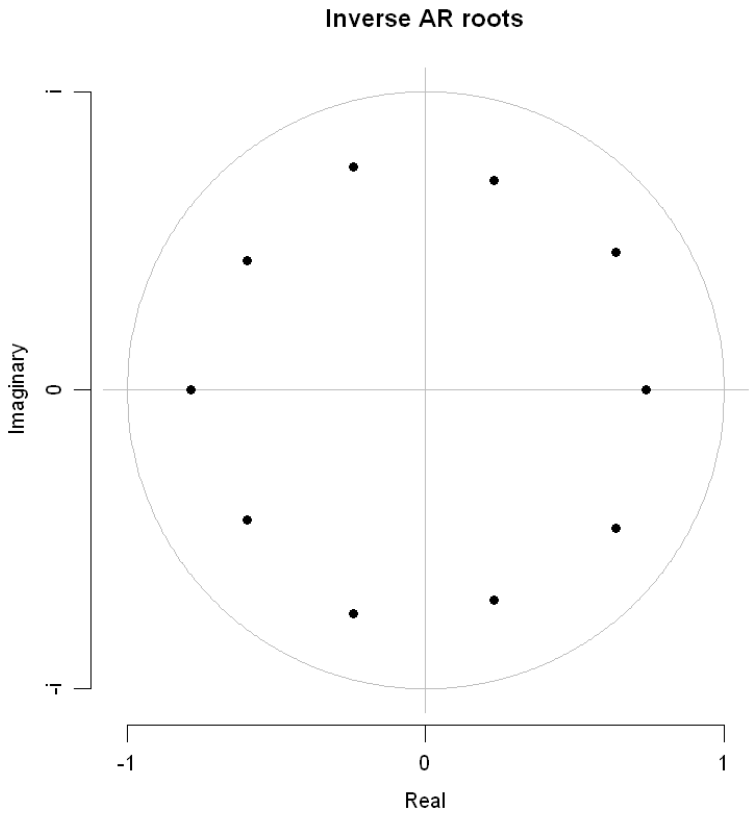
通过残差的 ACF 图, 可以看出, 尽管残差的稳定性和正态性都比较好, 但模型仍不是十分理想.

接下来使用此模型对于未来的走势进行预测.

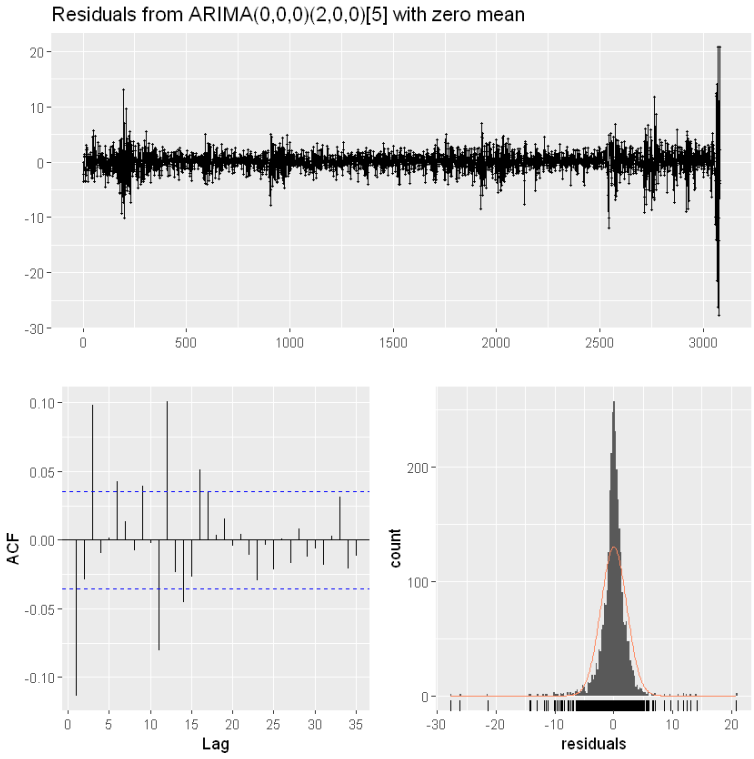


4 剔除季节因素

想到可能价格会受到季节性的影响, 为此, 先对于价格差分序列拟合一个纯季节模型.



同样, 可以看到它是平稳的.

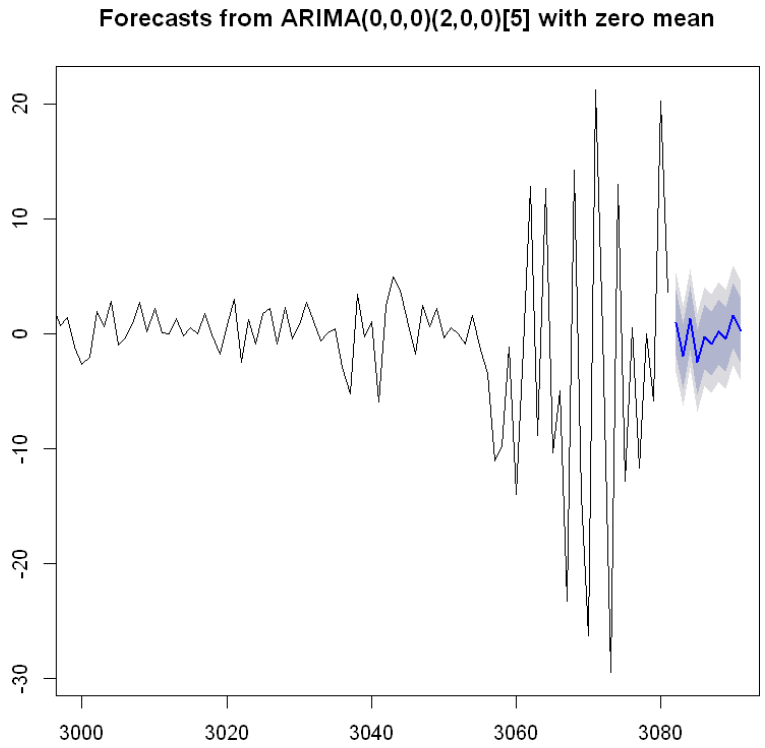


同样的, 这个模型的 ACF 序列也说明春季节性的模型并不理想.

得到的模型如下

$$(1 + 0.0817B^5 - 0.0667B^{10})Y_t = b_t$$

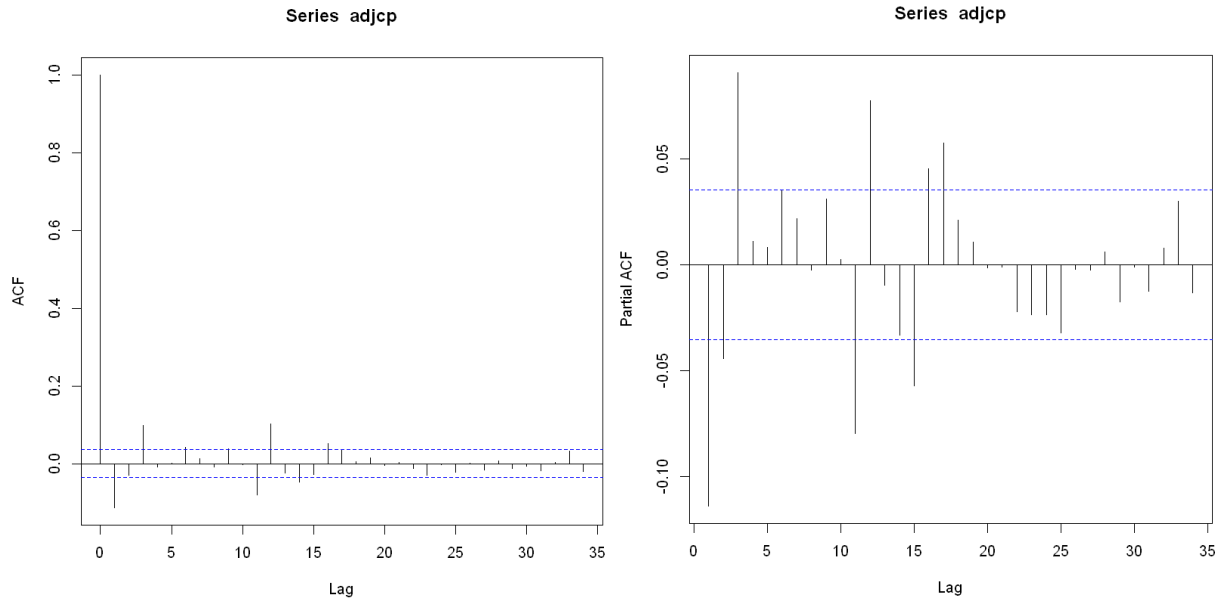
使用此模型进行预测结果如下.



因为数据为股票市场, 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, 对于 Y_t, 结合春季节性模型进行调整

$$Y_t^* = Y_t + 0.0817Y_{t-5} + 0.0667Y_{t-10}, t = 11, 12, 13, \cdots, 3071.$$

首先对调整后的自相关与偏自相关序列做分析.



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, 可以说我们对于季节性的影响或多或少的剔除了一点.

5 结合 GARCH 模型

首先简单了解一下 GARCH 模型

5.1 简单介绍

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.

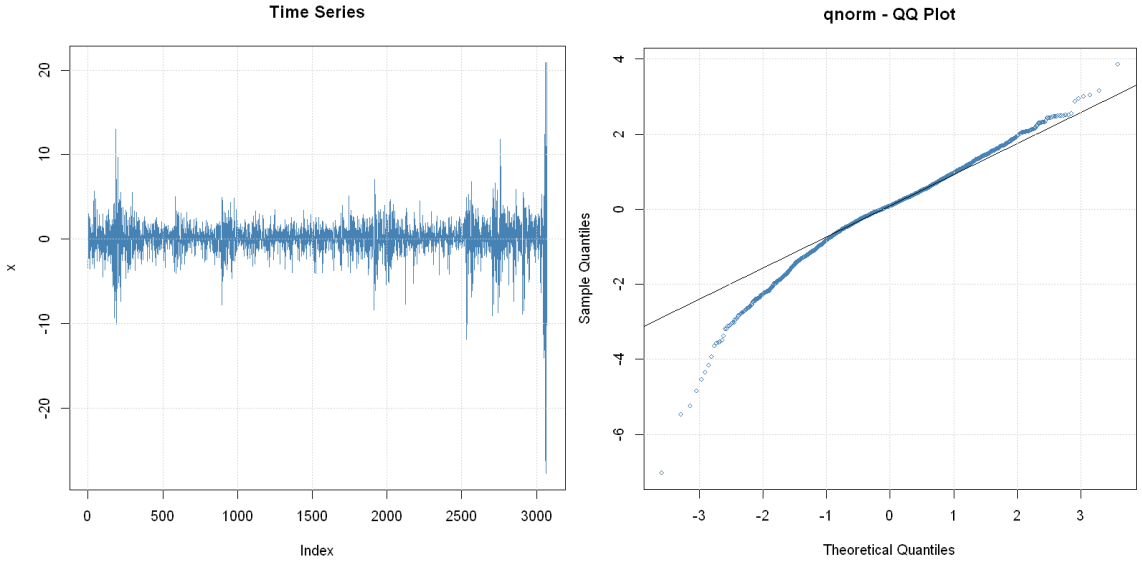
5.2 使用 GARCH 初次建模

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, 我们使用残差的相关图像对模型进行了可视化.



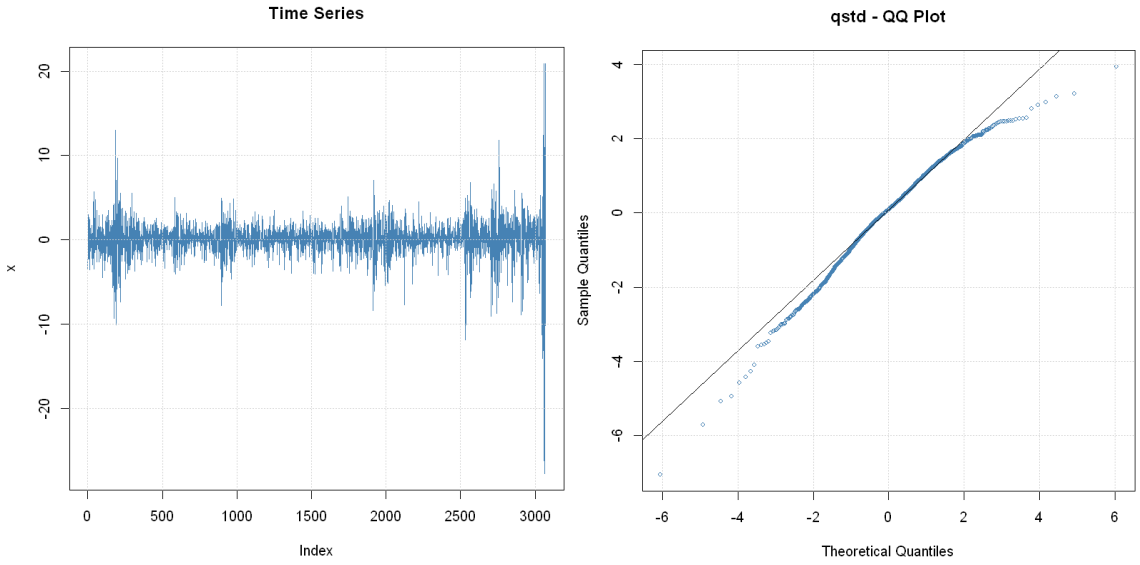
可以明显地看到, 标准化残差的正态性并不是十分的好, 因此模型还需进一步的调整.

5.3 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, 并且我们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.



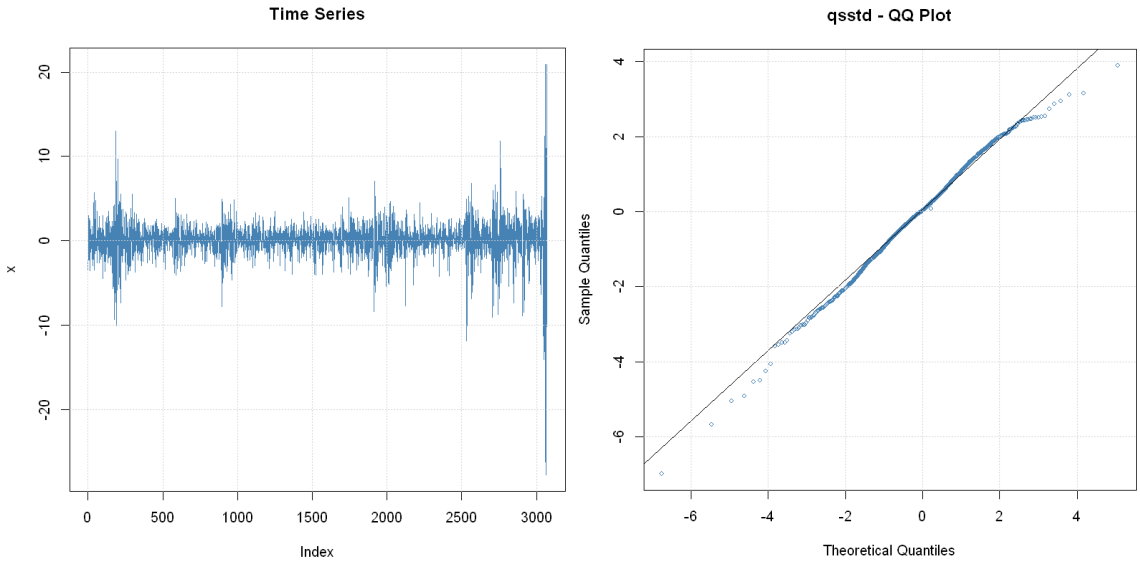
从图中可以发现, 此时模型的效果反而变差了. 标准化残差的分布依旧左偏, 并且左偏得更加严重. 因此, 需要更新对于新息分布的假设. 自然而然的就考虑了能够处理这样情况的偏 t 分布.

5.4 缩减 $ARMA$ 的系数个数

由于前两次的模型除了 $ARMA$ 的第一个系数以外, 其他系数都没有任何的统计显著性. 因此, 考虑将模型更换到 $ARMA(1,0)+GARCH(1,1)$. 并且采用偏 t 分布作为新息分布. 拟合的模型如下

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

在这个模型中, $ARMA$ 的第一个系数的显著性依旧十分的高. 残差的分布可视化如下

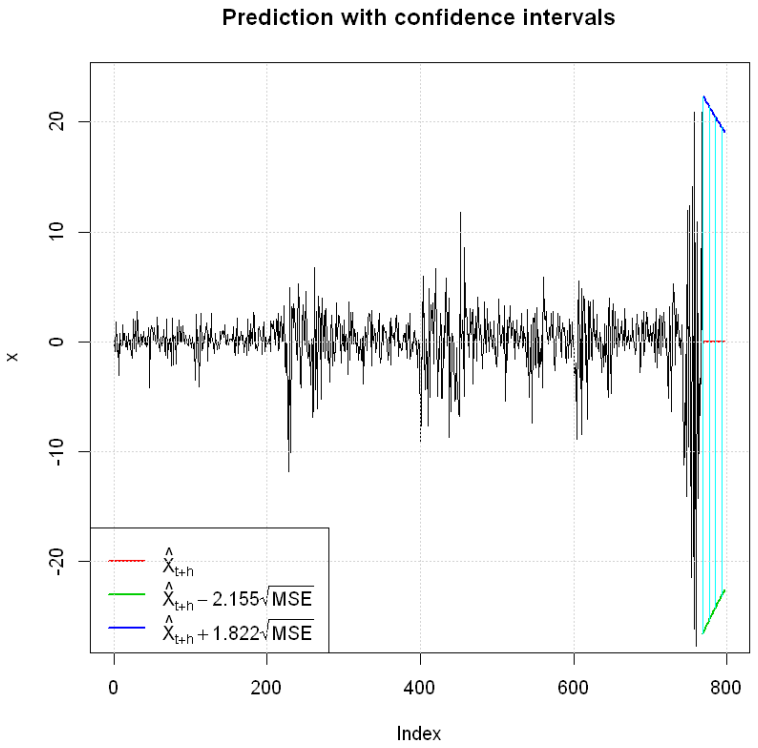


可以看到, 绝大部分点在该图中都表现出了一条直线的特征. 故可以认为偏 t 分布的假设对于新息而言是十分合理的, 虽然还是有轻微的左偏性.

接下来, 对超前五步的数值进行预测, 结果如下

meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
-1.430017e-01	12.28936	12.28936	-26.62170	22.24396
9.020667e-03	12.24464	12.22007	-26.37331	22.31450
-5.690311e-04	12.17573	12.15121	-26.23443	22.17939
3.589496e-05	12.10715	12.08276	-26.08606	22.05506
-2.264284e-06	12.03899	12.01474	-25.93924	21.93086

将其图示化, 如下图



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.

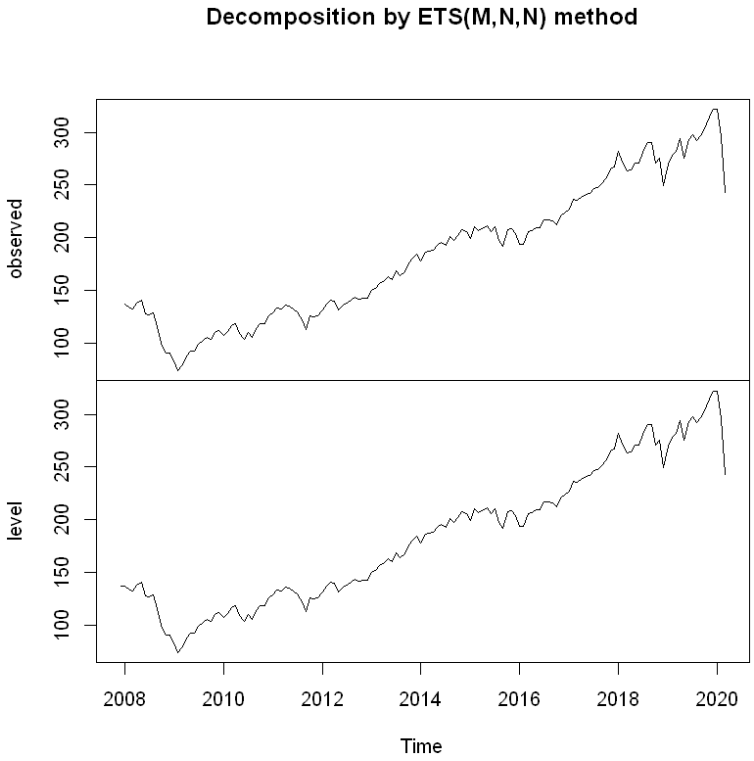
6 指数模型

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, 获得如下结果.

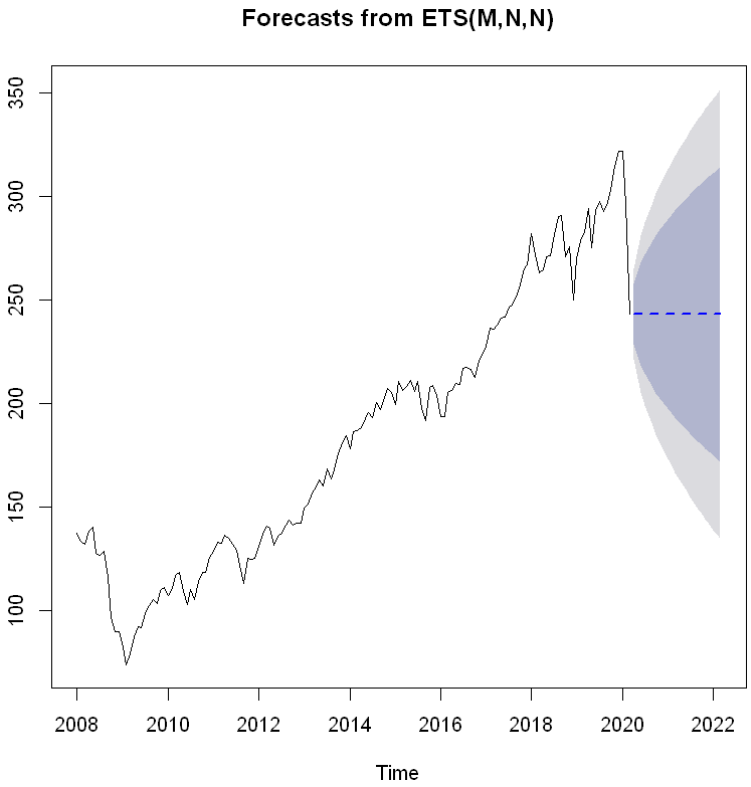


由于 ets() 的预测 frequency 的最大选择为 24, 所以原数据直接采用了每月的 S&P500 指数. 可以看到, 最终结果为 $ETS(M,N,N)$, 即代表了单指数相乘模型. 也就是

$$Y_t = level + irregular_r.$$

而 α 参数控制权数下降的速度, 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.

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, 其中权图中给出了其折线图和以下八个季度的预测.



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.

参考文献

[1] 周志华. 机器学习. 清华大学出版社, 2016.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[3] Iris data set - uci machine learning repository. https://archive.ics.uci.edu/ml/datasets/Iris, 1988.

[4] 集 成 学 习. https://www.jiqizhixin.com/graph/technologies/29722de0-8501-4b01-9b73-189141b9eefd.

[5] Adaboost classifier in python. https://www.datacamp.com/community/tutorials/adaboost-classifier-python.

A 完整代码

[1]:
```
library(fGARCH)
library(ggplot2)
library(forecast)
library(tseries)
library(scales)
library(plyr)
library(TTR)
set.seed(2020)
daily_2008 <- read.csv("./Data/SP500/08 年每日.csv")
SPY <- ts(daily_2008$收盘, frequency = 251, start = c(2008, 1))
Diffrence_of_SPY <- ts(diff(SPY), frequency = 251, start = c(2008, 1))
plot(SPY)
plot(Diffrence_of_SPY)
cprice <- diff(daily_2008$收盘)
acf(cprice)
pacf(cprice)
m2 <- arima(cprice, order = c(3, 0, 0),
            seasonal = list(order = c(2, 0, 0), period = 5), include.mean = F)
summary(m2)
```

```
Call:
arima(x = cprice, order = c(3, 0, 0), seasonal = list(order = c(2, 0, 0), period
= 5),
    include.mean = F)

Coefficients:
          ar1      ar2     ar3     sar1    sar2
      -0.1151  -0.0337  0.0943  -0.0708  0.0625
s.e.   0.0180   0.0186  0.0182   0.0186  0.0192

sigma^2 estimated as 4.69:  log likelihood = -6752.41,  aic = 13516.81

Training set error measures:
                     ME     RMSE      MAE MPE MAPE      MASE         ACF1
Training set 0.0347671 2.165543 1.344143 NaN  Inf 0.6757011 -0.001454335
```

[2]:
```
plot(m2)
checkresiduals(m2)
```

```
        Ljung-Box test

data:  Residuals from ARIMA(3,0,0)(2,0,0)[5] with zero mean
Q* = 7.5057, df = 5, p-value = 0.1857

Model df: 5.   Total lags used: 10
```

[3]:
```
plot(forecast(m2,h=10),xlim=c(3000,3090))
m3 <- arima(cprice, seasonal = list(order = c(2, 0, 0), period = 5), include.mean = F)
summary(m3)
```

```
Call:
arima(x = cprice, seasonal = list(order = c(2, 0, 0), period = 5), include.mean
= F)

Coefficients:
         sar1    sar2
      -0.0817  0.0667
s.e.   0.0183  0.0191

sigma^2 estimated as 4.801:  log likelihood = -6788.66,  aic = 13583.32

Training set error measures:
                     ME     RMSE      MAE MPE MAPE      MASE       ACF1
Training set 0.03290312 2.191185 1.340943 NaN  Inf 0.6740924 -0.1135116
```

[4]:
```
plot(m3)
checkresiduals(m3)
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,0,0)(2,0,0)[5] with zero mean
Q* = 83.375, df = 8, p-value = 1.021e-14

Model df: 2.   Total lags used: 10
```

[5]:
```
plot(forecast(m3,h=10),xlim=c(3000,3090))
adjcp <- cprice[11:3081] + 0.0817 * cprice[6:3076] - 0.0667 * cprice[1:3071]
par(mfrow = c(2, 1))
acf(adjcp)
pacf(adjcp)
m4 <- GARCHFit(~arma(3, 0) + GARCH(1, 1),
               data = adjcp, trace = F, include.mean = F)
summary(m4)
```

```
Title:
 GARCH Modelling

Call:
 GARCHFit(formula = ~arma(3, 0) + GARCH(1, 1), data = adjcp, include.mean = F,
    trace = F)

Mean and Variance Equation:
 data ~ arma(3, 0) + GARCH(1, 1)
<environment: 0x0000000020098838>
 [data = adjcp]

Conditional Distribution:
 norm

Coefficient(s):
       ar1         ar2         ar3       omega      alpha1       beta1
-0.0428183  -0.0062084  -0.0104306   0.1035492   0.1448658   0.8288358

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1    -0.042818    0.019885   -2.153   0.0313 *
ar2    -0.006208    0.019714   -0.315   0.7528
ar3    -0.010431    0.019380   -0.538   0.5904
omega   0.103549    0.016815    6.158 7.36e-10 ***
alpha1  0.144866    0.013950   10.385  < 2e-16 ***
beta1   0.828836    0.015207   54.502  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -5785.103    normalized:  -1.883785

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  897.1599  0
 Shapiro-Wilk Test  R    W      0.9735453 0
 Ljung-Box Test     R    Q(10)  14.04683  0.1708662
 Ljung-Box Test     R    Q(15)  20.6747   0.1475486
 Ljung-Box Test     R    Q(20)  29.32515  0.08155163
 Ljung-Box Test     R^2  Q(10)  16.13787  0.09575306
 Ljung-Box Test     R^2  Q(15)  21.61548  0.1182901
 Ljung-Box Test     R^2  Q(20)  23.78189  0.2520496
 LM Arch Test       R    TR^2   18.69182  0.09624218

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
```

```
3.771477 3.783258 3.771470 3.775710
```

[6]:
```
plot(m4, which = 1)
plot(m4, which = 13)
m5 <- GARCHFit(~arma(3, 0) + GARCH(1, 1),
               data = adjcp, trace = F, include.mean = F, cond.dist = "std")
summary(m5)
```

```
Title:
 GARCH Modelling

Call:
 GARCHFit(formula = ~arma(3, 0) + GARCH(1, 1), data = adjcp, cond.dist = "std",
    include.mean = F, trace = F)

Mean and Variance Equation:
 data ~ arma(3, 0) + GARCH(1, 1)
<environment: 0x0000000015033cf8>
 [data = adjcp]

Conditional Distribution:
 std

Coefficient(s):
       ar1         ar2         ar3       omega      alpha1       beta1
-0.0395068  -0.0046808  -0.0035299   0.0739401   0.1348823   0.8515826
     shape
 5.9149506

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1    -0.039507    0.018535   -2.132    0.033 *
ar2    -0.004681    0.018732   -0.250    0.803
ar3    -0.003530    0.018477   -0.191    0.848
omega   0.073940    0.017372    4.256 2.08e-05 ***
alpha1  0.134882    0.016180    8.337  < 2e-16 ***
beta1   0.851583    0.016356   52.065  < 2e-16 ***
shape   5.914951    0.652243    9.069  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -5705.55    normalized:  -1.85788

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  976.3192  0
 Shapiro-Wilk Test  R    W      0.9727265 0
 Ljung-Box Test     R    Q(10)  13.80983  0.1818439
 Ljung-Box Test     R    Q(15)  20.30748  0.1604519
 Ljung-Box Test     R    Q(20)  28.68535  0.09414238
 Ljung-Box Test     R^2  Q(10)  14.30046  0.1597227
 Ljung-Box Test     R^2  Q(15)  20.81853  0.1427284
 Ljung-Box Test     R^2  Q(20)  23.03286  0.2871838
 LM Arch Test       R    TR^2   17.54599  0.1301872

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
3.720319 3.734063 3.720308 3.725257
```

[7]:
```
plot(m5, which = 1)
plot(m5, which = 13)
m6 <- GARCHFit(~arma(1, 0) + GARCH(1, 1),
               data = adjcp, trace = F, include.mean = F, cond.dist = "sstd")
summary(m6)
```

```
Title:
 GARCH Modelling

Call:
 GARCHFit(formula = ~arma(1, 0) + GARCH(1, 1), data = adjcp, cond.dist = "sstd",
    include.mean = F, trace = F)

Mean and Variance Equation:
 data ~ arma(1, 0) + GARCH(1, 1)
<environment: 0x000000001e801b20>
 [data = adjcp]

Conditional Distribution:
 sstd

Coefficient(s):
      ar1      omega     alpha1      beta1       skew      shape
-0.063081   0.072857   0.132903   0.855370   0.856473   6.084229

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1     -0.06308     0.01861   -3.390 0.000699 ***
omega    0.07286     0.01681    4.335 1.46e-05 ***
alpha1   0.13290     0.01568    8.475  < 2e-16 ***
beta1    0.85537     0.01565   54.655  < 2e-16 ***
skew     0.85647     0.01979   43.284  < 2e-16 ***
shape    6.08423     0.71690    8.487  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -5682.663    normalized:  -1.850428

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  980.1835  0
 Shapiro-Wilk Test  R    W      0.9725485 0
 Ljung-Box Test     R    Q(10)  15.80819  0.1052563
 Ljung-Box Test     R    Q(15)  22.69048  0.0909363
 Ljung-Box Test     R    Q(20)  31.27833  0.05162165
 Ljung-Box Test     R^2  Q(10)  15.33821  0.1202093
 Ljung-Box Test     R^2  Q(15)  21.69158  0.1161498
 Ljung-Box Test     R^2  Q(20)  24.03101  0.2410402
 LM Arch Test       R    TR^2   18.63672  0.09768077

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
3.704763 3.716543 3.704755 3.708995
```

[8]:
```r
plot(m6, which = 1)
plot(m6, which = 13)
predict(m6, n.ahead = 5, plot = TRUE)
```

meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
-1.430017e-01	12.28936	12.28936	-26.62170	22.24396
9.020667e-03	12.24464	12.22007	-26.37331	22.31450
-5.690311e-04	12.17573	12.15121	-26.23443	22.17939
3.589496e-05	12.10715	12.08276	-26.08606	22.05506
-2.264284e-06	12.03899	12.01474	-25.93924	21.93086

[9]:
```r
monthly_2008 <- read.csv("./Data/SP500/08 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)
```

```
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
```

[10]:
```
plot(fit_ets)
plot(forecast(fit_ets), xlab = "Time", flty = 2)
accuracy(fit_ets)
```

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