



**题目:** **基于GARCH模型的Apple.Inc股票价格**

**姓 名 杜泽宣**

**学 院 理学院**

**专 业 数学与应用数学**

**班 级 2021214101**

**学 号 2021212487**

**指导教师 李晓花**

# 2024年6月10日

**基于GARCH模型的Apple.Inc股票价格**

**摘 要**

本文对Apple.Inc股票价格数据进行分析和预测。数据源自Kaggle，筛选了从2006-01-03到2017-12-29的收盘价。首先，通过加载数据并调整变量类型，进行了数据的清洗和预处理。随后，利用时间序列分析方法，包括趋势分析和平稳性检验，对数据进行了全面的探索。接着，尝试了不同的预测模型，包括线性模型和ARIMA模型,GARCH、TGARCH、EGARCH模型，并对预测结果进行了比较和评估。最终，得到标准T分布假定ARMA(2,2)-GARCH(1,2)的效果最好。在预测中发现了模型在短期内较好，而长期拟合效果非常差。

**关键词** 股票价格，ARIMA模型，GARCH模型，置信区间

**目 录**

[**学 院 理学院** 1](#_Toc168606335)

[**班 级 2021214101** 1](#_Toc168606336)

[**学 号 2021212487** 1](#_Toc168606337)

[2024年6月10日 1](#_Toc168606338)

[**基于GARCH模型的Apple.Inc股票价格** 2](#_Toc168606339)

[**摘 要** 2](#_Toc168606340)

[第一章 引言 1](#_Toc168606341)

[1.1 背景介绍 1](#_Toc168606342)

[1.2 本论文工作简述 1](#_Toc168606343)

[第二章 数据集介绍 2](#_Toc168606344)

[2.1 数据来源 2](#_Toc168606345)

[2.2 变量解释 2](#_Toc168606346)

[第三章 数据分析 2](#_Toc168606347)

[3.1 加载数据，调整变量类型 2](#_Toc168606348)

[3.2 趋势分析 3](#_Toc168606349)

[3.2.1 创建时间序列，画散点图 3](#_Toc168606350)

[3.2.2 检查正态性 4](#_Toc168606351)

[3.3 平稳性检验 4](#_Toc168606352)

[3.3.1 ADF检验 4](#_Toc168606353)

[3.3.2 自相关图与偏自相关图 5](#_Toc168606354)

[第四章 算法预测及结果 5](#_Toc168606355)

[4.1 趋势预测 5](#_Toc168606356)

[4.1.1 线性预测 5](#_Toc168606357)

[4.2 ARIMA模型 5](#_Toc168606358)

[4.2.1 查看ACF和PACF 5](#_Toc168606359)

[4.2.2 扩展自相关函数（EACF） 6](#_Toc168606360)

[4.2.3 最优子集 6](#_Toc168606361)

[4.2.4 预测 6](#_Toc168606362)

[4.3 GARCH模型 7](#_Toc168606363)

[4.3.1 标准GARCH模型 7](#_Toc168606364)

[4.3.2 其他GARCH模型 9](#_Toc168606365)

[4.3.2 预测 9](#_Toc168606366)

[第五章 分析与结论 9](#_Toc168606367)

[参考文献 11](#_Toc168606368)

[附录 11](#_Toc168606369)

# 第一章 引言

## 1.1 背景介绍

股票价格的预测一直是一个复杂而具有挑战性的任务。股票市场的波动性、非线性和不确定性使得预测变得极其困难。然而，对于投资者、交易者以及金融分析师而言，准确地预测股票价格对于做出明智的投资决策至关重要。在这个背景下，统计和计量经济学模型如自回归积分移动平均模型（ARIMA）和广义自回归条件异方差模型（GARCH）应运而生，成为研究市场动态和预测未来走势的重要工具。这里我们以Apple.Inc为研究对象。由于其公司规模非常大，成交量和成交额也比较庞大，可以较大限度的减弱因偶然原因导致的股票波动。

## 1.2 本论文工作简述

首先，我们从Kaggle获取了Apple.Inc开盘价、收盘价、交易量等数据，并筛选出日期及收盘价的相关数据。随后，进行了数据清洗和预处理，包括缺失值处理和变量调整，确保数据的准确性和可用性，并计算每日的对数收益率。

接着，通过时间序列分析方法，包括趋势分析、平稳性检验，对维也纳气温数据进行了全面的探索。我们利用统计工具和可视化技术，揭示了数据的特征和规律，为后续建模和预测奠定了基础。

然后，我们尝试了不同的预测模型，包括线性模型、ARIMA模型和不同GARCH模型，并对每种模型的拟合效果进行了评估和比较。通过对比测试集和预测结果，我们验证了模型的准确性和可靠性，并分析了其优缺点。

# 第二章 数据集介绍

## 2.1 数据来源

本数据源于Kaggle([DJIA 30 Stock Time Series (kaggle.com)](https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231))，数据原为全每日各家公司的收盘价、成交量、交易额，本论文将Apple.Inc的收盘价数据筛选出来单独进行研究。

## 2.2 变量解释

原数据有以下变量Date,Open,High,Low,Close,Volume,Name，分别表示交易时间、开盘价、最高价、最低价、收盘价、成交量、股票名，经过处理后保留以下变量：

Date 日期

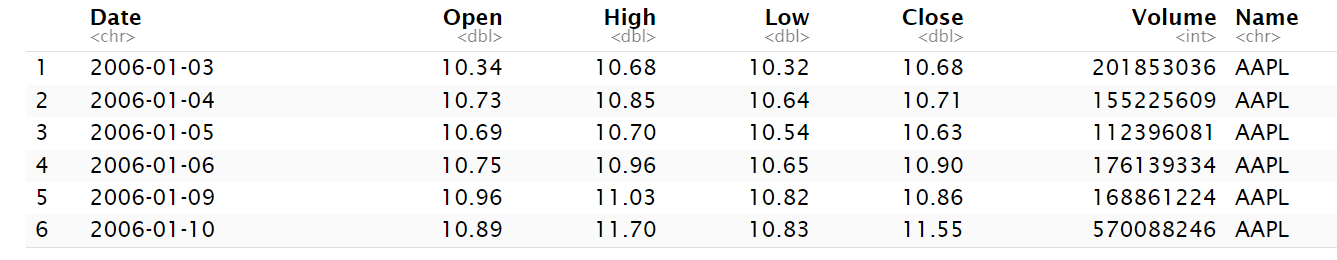
Close 收盘价

原数据中Date变量格式为“2013-04-28”表示UTC标准时间2013年04月28日。

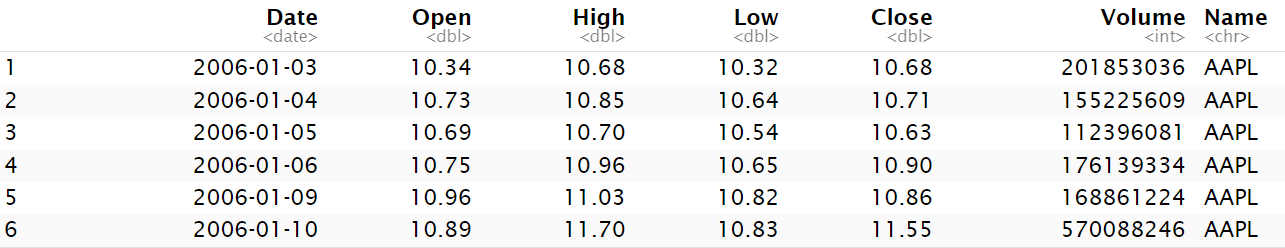
# 第三章 数据分析

## 3.1 加载数据，调整变量类型

从csv中加载数据集，定义存在变量行，查看数据集。



将Date变量转换为date型：



检查数据，数据无误。计算对数收益率，命名为log\_return。

表格

描述已自动生成

## 3.2 趋势分析

### 3.2.1 创建时间序列，画散点图

将时间列转换为时间序列对象，频率为1。

文本

描述已自动生成

画收盘价散点图，对数收益率散点图、时序图。

收盘价散点图：

图表, 折线图

描述已自动生成

对数收益率散点图：

图表, 散点图

描述已自动生成

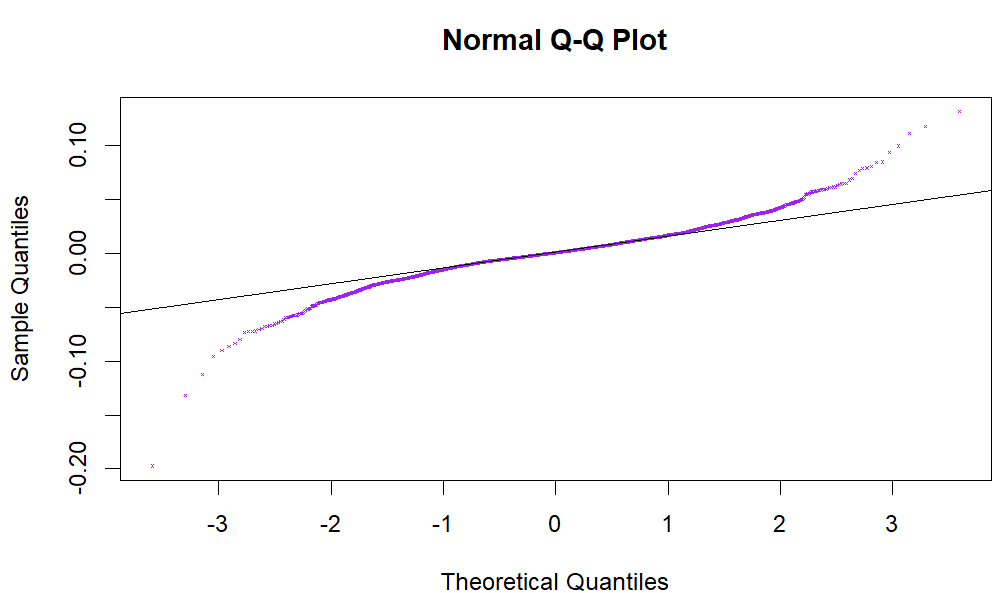
对数收益率时序图：

图表

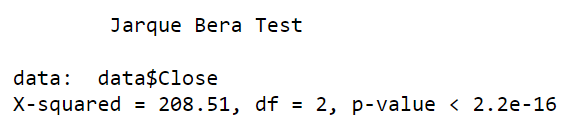
描述已自动生成

### 3.2.2 检查正态性

画log\_xts的QQ图：



J-B检验



显然不符合正态数据特征

## 3.3 平稳性检验

### 3.3.1 ADF检验

检查数据的平稳性：ADF检验是Dickey-Fuller检验的一个特例，通常用于检验单个时间序列是否具有单位根。adf.test函数只能进行常数或常数和趋势的ADF检验，不能进行没有趋势的检验。它返回的检验统计量包括ADF统计量、p值和用于计算p值的临界值。

文本

描述已自动生成

显然，数据是平稳的。

### 3.3.2 自相关图与偏自相关图

画自相关图与偏自相关图：

ACF/PACF

图表

描述已自动生成图表, 箱线图

描述已自动生成

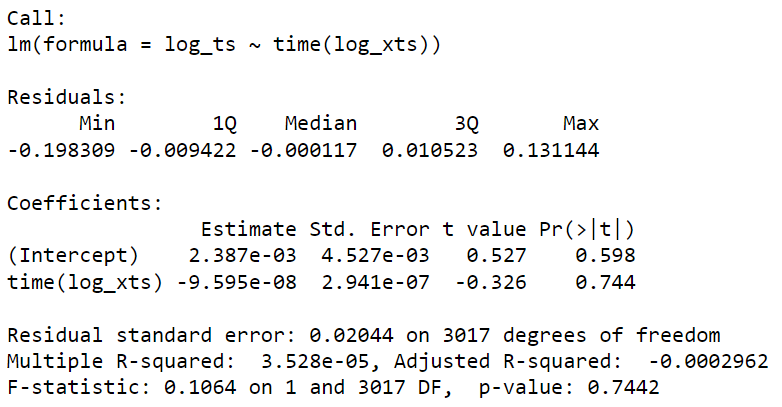
ACF图与PACF图都没有明显的表现出周期性。

# 第四章 算法预测及结果

## 4.1 趋势预测

### 4.1.1 线性预测

对变量进行线性模拟：

日程表

描述已自动生成

显然，收益率波动非常大，数据不能使用线性模型进行拟合。

## 4.2 ARIMA模型

### 4.2.1 查看ACF和PACF

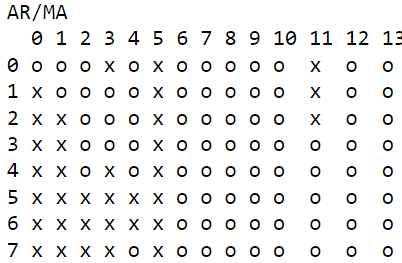
ACF/PACF：[3.3.2 自相关图与偏自相关图](#_3.3.2_自相关图与偏自相关图)

通过观察ACF和PACF确定MA和AR参数为（4,4）。

### 4.2.2 扩展自相关函数（EACF）

查看扩展自相关函数（数据为平稳的，不做差分）：

p-value<0.05



参数可能为ARMA(0,0) 。

### 4.2.3 最优子集

ARMA(2,0,2)（以aic为判断依据）

文本

描述已自动生成

### 4.2.4 预测

Forecast：

文本

中度可信度描述已自动生成图表

描述已自动生成

残差-时间图：

图表

描述已自动生成

qq图

图表, 折线图

描述已自动生成

预测效果比较差。

## 4.3 GARCH模型

### 4.3.1 标准GARCH模型

我们对ARMA（2，2）的残差分别进行Mcleod-Li检验。从Mcleod-Li检验的结果图中可以看出，时间序列不适合ARCH或GARCH模型，因此可以对残差序列进行GARCH建模。

图表

低可信度描述已自动生成

使用aic准则对GARCH模型进行定阶：

|  |  |  |
| --- | --- | --- |
| 模型 | AIC | BIC |
| **（1，1）** | 11.39180 | 11.38383 |
| （1，2） | 11.39154 | 11.38158 |
| （2，1） | 11.43689 | 11.42693 |
| （2，2） | 11.43623 | 11.42428 |
| （2，3） | 11.47650 | 11.46256 |
| （3，1） | 11.46275 | 11.45080 |
| （3，2） | 11.46275 | 11.45080 |
| （3，3） | 11.45965 | 11.44372 |

从表中可以看出GARCH(1,2)比较合适，表达式为

我们针对数据特点考虑不同的GARCH模型，当我们在GARCH中对残差使用t分布，此时aic= -11.36228 ,bic=-11.20874。

当我们在GARCH中对残差使用GED分布，Aic=-11.55164 bic=-11.42137

我们可以认为此时ARMA(2,2)-GARCH(1,2)-t效果最好。

### 4.3.2 其他GARCH模型

我们同时建立ARMA（2,2）-EGARCH（1,2）-std模型和ARMA（2,2）-TGARCH（1,2）-std模型并和标准的ARMA（2,2）-GARCH（1,2）-std模型进行对比，得出下表：

|  |  |  |
| --- | --- | --- |
| 分布 | AIC | BIC |
| **GARCH** | 11.36228 | 11.20874 |
| TGARCH | 11.46255 | 11.40253 |
| EGARCH | 11.43397 | 11.39915 |

从表中可以看出，标准ARMA（4，3）-GARCH（1，1）-std模型拟合效果最好。

### 4.3.2 预测

根据我们上得到的结论进行预测：标准ARMA（4，3）-GARCH（1，1）-std

表格

低可信度描述已自动生成文本

描述已自动生成

当然，R不能处理休息日不开盘的情况，我们手动对应每天的情况，例如2017/12/30对应2018/1/2等等，对比纽交所给出的数据计算得到的对数收益率，发现前四天预测效果比较好，但接下来的数据效果非常差，可能是因为某些政策对数据的影响比数据自身大。、、

# 第五章 分析与结论

通过对Apple.Inc股票价格的分析，使用ARIMA模型拟合对数收益率得到的模型为ARIMA(2,0,2)。同时发现数据残差序列存在异方差性，可以建立GARCH模型。经过GARCH、GARCH-t、GARCH-ged建模，发现模型的残差序列分布不服从正态分布。因此我们考虑EGARCH、TGARCH不同模型进行分析，发现标准T分布假定ARMA(2,2)-GARCH(1,2)的效果最好。

同时，我们发现现实的金融数据收到非常多方面的影响，包括政策，经济环境，上下游产业链，股民心态等各个方面的影响。单从数据本身寻找相关性得到预测数据的行为是片面的，分析得到的数据可能在短期内拟合效果较好，但长期效果会非常差。

# 参考文献

[1] [纽 交 所 (nyse.com)](https://www.nyse.com/quote/XNGS:AAPL)

[2] [DJIA 30 Stock Time Series (kaggle.com)](https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231)

[3] [时间序列分析中的GARCH模型与应用-CSDN博客](https://blog.csdn.net/universsky2015/article/details/135808133?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171768469116800226575244%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171768469116800226575244&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-2-135808133-null-null.142%5ev100%5econtrol&utm_term=GARCH&spm=1018.2226.3001.4187)

[4] [时间序列分析波动性预测GARCH模型-CSDN博客](https://blog.csdn.net/Aresiii/article/details/131630834?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171768469116800226575244%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171768469116800226575244&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-3-131630834-null-null.142%5ev100%5econtrol&utm_term=GARCH&spm=1018.2226.3001.4187)

[5] [时间序列分析：ARCH、GARCH、TGARCH与DCC-GARCH波动率模型（含代码）\_建立波动率用什么模型-CSDN博客](https://blog.csdn.net/celiaweiwei/article/details/135762880?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171768472216800211516261%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171768472216800211516261&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-135762880-null-null.142%5ev100%5econtrol&utm_term=TGARCH&spm=1018.2226.3001.4187)

[6] [msperlin/GARCH-RAC: Repository for GARCH tutorial paper in RAC (github.com)](https://github.com/msperlin/GARCH-RAC)

[7] [R语言： GARCH模型股票交易量的研究道琼斯股票市场指数\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1Aw4m1v7Pk/?spm_id_from=333.337.search-card.all.click)

# 附录

1. R代码（APPL.R）

library(fpp)

library(TSA)

library(xts)

library(FinTS)

library(rugarch)

library(rmgarch)

library(Rmpfr)

library(quantmod)

library(DataCombine)

library(tseries)

library(urca)

library(dplyr)

library(fGarch)

APPL = "APPL.csv"

data <- read.csv(APPL)

head(data)

#data$Date <- as.POSIXct(data$Date)

data$Date <- as.Date(data$Date, format = "%Y-%m-%d")

head(data)

# 计算对数收益率并添加为新列

data <- data %>%

  mutate(log\_return = log(Close / lag(Close, default = first(Close))))

# 使用first(Price)处理第一天没有前一日价格的情况

#data <- data %>%

#  mutate(log\_return = data$Price)

# 查看结果

head(data)

# 提取起始日期

start\_date <- as.Date(min(data$Date))

# 计算总的观测天数

n\_days <- nrow(df)

# 转换为ts对象，这里假设数据是每日的，所以frequency为365

log\_xts <- xts(data$log\_return, order.by = data$Date)

summary(log\_xts)

plot(data$Close,pch =16 , col = "RED", cex = 0.2)

plot(data$log\_return,pch =16 , col = "RED", cex = 0.2)

plot(log\_xts)

#pdf("QQ.pdf",width =5,height = 5)

qqnorm(log\_xts,pch =4 , col = "RED", cex = 0.2);qqline(log\_xts)

jb\_test <- jarque.bera.test(data$Close)

print(jb\_test)

adf=adf.test(log\_ts)

print(adf)

acf(log\_xts)

pacf(log\_xts)

trend\_lm <- lm(log\_ts ~ time(log\_ts))

summary(trend\_lm)

plot(log\_ts, main = "BTC Time Series with Linear Trend", ylab = "Price")

abline(trend\_lm, col = "red")

eacf(log\_xts)

#log\_xts\_1 <- na.omit(log\_xts)

arima\_model <- Arima(log\_ts, order = c(2,0,2))

summary(arima\_model)

forecast\_values <- forecast(arima\_model,h=30)

plot(forecast\_values)

#lines(data$log\_return, col = "red")

residuals <- residuals(arima\_model)

# 绘制残差-时间图

plot(residuals, xlab = "Time", ylab = "Residuals", type = "o", pch = 32, col = "blue")

abline(h = 0, col = "red", lty = 2)

qqnorm(residuals)

qqline(residuals)

McLeod.Li.test(y=log\_xts)

#正常

m1<-auto.arima(log\_xts)

# 拟合GARCH模型

arma\_residuals <- residuals(m1)

garch\_fit <- garchFit(formula=~garch(1,2),data = arma\_residuals^2, trace = FALSE)

summary(garch\_fit)

#残差检验

shapiro.test(residuals(garch\_fit, standardize = TRUE))

# t分布？

m1<-auto.arima(log\_xts)

# 拟合GARCH模型

arma\_residuals <- residuals(m1)

garch\_fit\_t <- garchFit(formula=~garch(1,2),data = arma\_residuals^2, trace = FALSE,distribution.model = "std")

summary(garch\_fit)

#TGARCH

tgarch\_spec <- ugarchspec(variance.model = list(model = "fGARCH",

                                                submodel = "TGARCH",

                                                garchOrder = c(1, 1)),

                          mean.model = list(armaOrder = c(2,2), include.mean = TRUE),

                          distribution.model = "ged")

# 拟合 TGARCH 模型

tgarch\_fit <- ugarchfit(spec = tgarch\_spec, data = log\_xts)

# 显示 TGARCH 模型拟合结果

print(tgarch\_fit)

#EGARCH

egarch\_spec <- ugarchspec(variance.model = list(model = "eGARCH", garchOrder = c(1, 1)),

                          mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),

                          distribution.model = "ged")

# 拟合 EGARCH 模型

egarch\_fit <- ugarchfit(spec = egarch\_spec, data = log\_xts)

# 显示 EGARCH 模型拟合结果

print(egarch\_fit)

#预测

forecast\_values <- ugarchforecast(garch\_fit\_t,h=10)