



**题目:** **基于ARIMA模型的气温分析**

**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4月25日

**基于ARIMA模型的金融分析**

**摘 要**

本文对维也纳气温数据进行分析和预测。数据源自Kaggle的全球城市每日温度记录，经过筛选和处理得到了维也纳的气温数据。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、季节性验证和平稳性检验，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、余弦模型和ARIMA模型，并对预测结果进行了比较和评估。最终，论文提出了对维也纳气温的预测结果，并对模型的拟合效果进行了讨论和总结。这项研究对理解维也纳气候特征和未来气温变化具有重要意义，为气象预测和城市规划提供了有益参考。

**关键词** ARIMA模型，气温预测，单位根检验，置信区间，ACF，PACF

**目 录**

[**学 院 理学院** 1](#_Toc165802965)

[**班 级 2021214101** 1](#_Toc165802966)

[**学 号 2021212487** 1](#_Toc165802967)

[2024年4月25日 1](#_Toc165802968)

[**摘 要** 2](#_Toc165802969)

[第一章 引言 1](#_Toc165802970)

[1.1 背景介绍 1](#_Toc165802971)

[1.2 本论文工作简述 1](#_Toc165802972)

[第二章 数据集介绍 2](#_Toc165802973)

[2.1 数据来源 2](#_Toc165802974)

[2.2 变量解释 2](#_Toc165802975)

[第三章 数据分析 2](#_Toc165802976)

[3.1 加载数据，调整变量类型 2](#_Toc165802977)

[3.1.1 清洗变量 2](#_Toc165802978)

[3.1.2 提取训练集和测试集 3](#_Toc165802979)

[3.2 趋势分析 3](#_Toc165802980)

[3.2.1 创建时间序列，画散点图 3](#_Toc165802981)

[3.2.2 季节性 3](#_Toc165802982)

[3.3 平稳性检验 4](#_Toc165802983)

[3.3.1 ADF检验 4](#_Toc165802984)

[3.3.2 自相关图与偏自相关图 4](#_Toc165802985)

[第四章 算法预测及结果 4](#_Toc165802986)

[4.1 趋势预测 4](#_Toc165802987)

[4.1.1 线性预测 4](#_Toc165802988)

[4.1.2 余弦模型 5](#_Toc165802989)

[4.2 ARIMA模型 6](#_Toc165802990)

[4.2.1 查看ACF和PACF 6](#_Toc165802991)

[4.2.2 扩展自相关函数（EACF） 7](#_Toc165802992)

[4.2.3 最优子集 7](#_Toc165802993)

[4.2.4 预测 9](#_Toc165802994)

[4.3 季节ARIMA 11](#_Toc165802995)

[第五章 分析与结论 13](#_Toc165802996)

[参考文献 14](#_Toc165802997)

[附录 15](#_Toc165802998)

# 第一章 引言

## 1.1 背景介绍

气候变化是当今世界面临的重要挑战之一，对人类社会、经济和生态系统都产生着深远影响。在这个背景下，对气候数据的分析和预测变得愈发重要，以帮助人们更好地理解气候变化趋势、制定应对措施以及规划未来发展方向。

维也纳作为奥地利的首都和文化中心，其气候变化对城市居民的生活和城市规划产生着重要影响。了解维也纳的气候特征、变化趋势以及未来预测，对于城市居民的生活、城市基础设施建设以及决策者的政策制定都具有重要意义。

本研究基于Kaggle提供的全球城市每日温度数据，选取维也纳的气温数据作为研究对象，旨在通过对这些数据的深入分析和建模，揭示维也纳气候变化的规律性、趋势以及未来的预测情况。通过此项研究，我们可以为维也纳的气象预测、城市规划和社会发展提供科学依据，为适应气候变化、提高城市生活质量做出贡献。

## 1.2 本论文工作简述

首先，我们从Kaggle获取了全球城市每日温度数据，并筛选出维也纳的相关数据。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。

接着，通过时间序列分析方法，包括趋势分析、季节性验证和平稳性检验，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、余弦模型和ARIMA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（[Daily Temperature of Major Cities (kaggle.com)](https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities)），数据原为全球各地区城市的每日温度（1996.9.11-2020.5.13），本论文将其奥地利维也纳的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Region,Country,State,City,Month,Day,Year,AvgTemperature，表示地区（XX洲），国家，洲（省/区），城市，月，日，年，平均气温（日平均气温，单位华氏度），经过处理后保留以下变量：

City 城市名

Month 月

Day 日

Year 年

Celsius\_Temperature 平均气温（摄氏度）

其中Celsius\_Temperature为转化后的摄氏温度，转换公式为：

Celsius\_Temperature=(AvgTemperature-32)/1.8

# 第三章 数据分析

## 3.1 加载数据，调整变量类型

### 3.1.1 清洗变量

从csv中加载数据集，定义存在变量行，查看数据集。

表格

中度可信度描述已自动生成

将年月日合并为一个时间变量：

表格

描述已自动生成

数据提供者在备注中说明：如果存在缺失、遗漏的数据，那么这个数据将标记为-99。在将华氏度转换为摄氏度时，-99变成-72.77777778，提取所有Celsius\_Temperature =-72.77777778的数据，查看：图形用户界面, 表格

中度可信度描述已自动生成

共26行数据，错误比较少，假设这些日期气温变化是线性的，用这种办法修补数据，修补后检查：

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

现在数据没有错误了。

### 3.1.2 提取训练集和测试集

创建训练集（2019-01-01之前）和测试集（2019-01-01之后）。显示训练集和测试集的行数：

图示

低可信度描述已自动生成

## 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，频率为一年=365天。



画散点图、时序图。

散点图：<散点图.pdf>

时序图：<时序图.pdf>

### 3.2.2 季节性

温度数据一定具有极强的季节性或周期性，进行季节性分解验证：

季节性分解：<seasonal.pdf>

两种分解方法都可以看出确实存在非常强烈的季节性或周期性。

## 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：<acf.pdf>

PACF：<pacf.pdf>

ACF图与PACF图都表现出了极强的周期性，ACF图

ACF图表明数据具有短期依赖性，即当前值与过去的值之间存在一定的相关性，但这种相关性随着时间的推移而逐渐减弱。

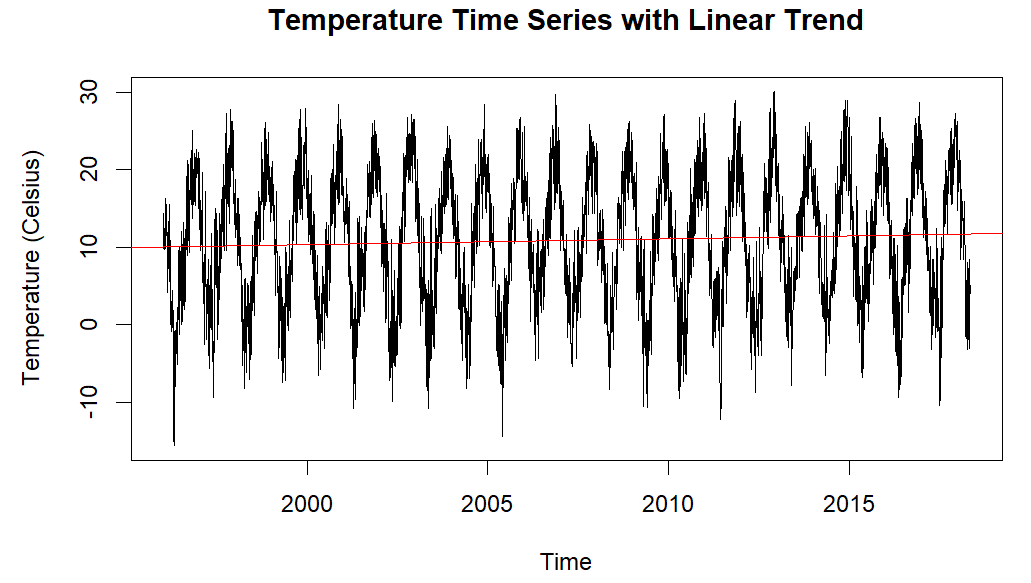
PACF图在lag=2或3以后相关性减弱。

# 第四章 算法预测及结果

## 4.1 趋势预测

### 4.1.1 线性预测

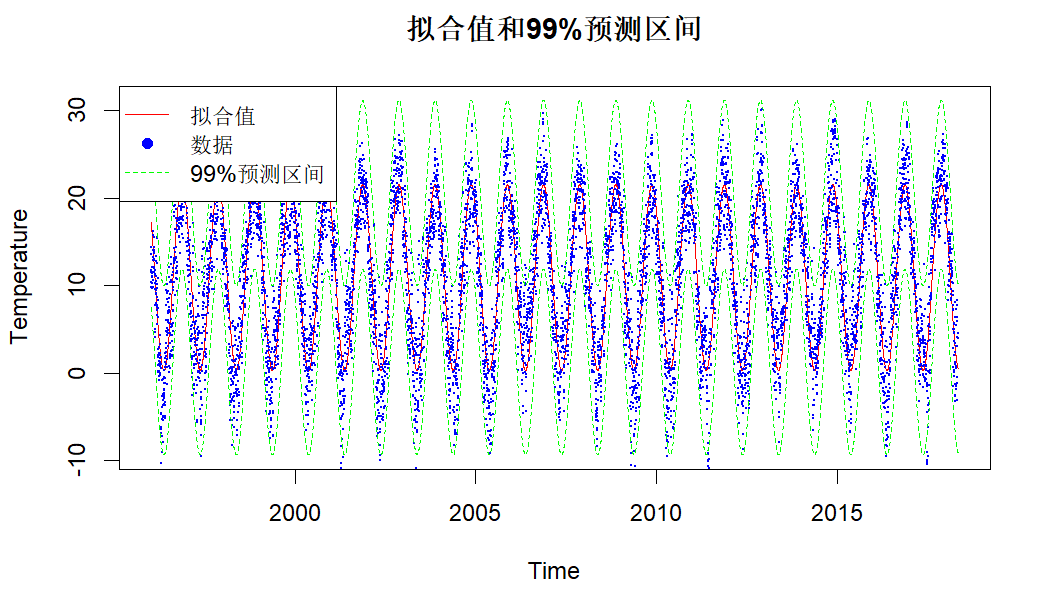
对变量进行线性模拟：



显然，气温数据不能使用线性模型进行拟合。

### 4.1.2 余弦模型

查看散点图可以发现数据呈现出类似于正弦函数或余弦函数的趋势，尝试使用余弦模型进行模拟并给出99%区间：



可以看出，在训练集中，预测情况是比较符合的。接下来查看残差情况

图片包含 图表

描述已自动生成

发现残差波动比较大，部分预测值与原始值存在4℃以上的差距。统计残差数据：图表, 直方图

描述已自动生成

看出残差的分布比较符合正态分布，检查qq图：

图表, 折线图

描述已自动生成

在（-2,2）内，基本成一条直线，比较符合正态分布，可以认为是偶尔出现的极端天气或数据记录错误。但是余弦模型并没有体现出线性拟合种气温曲线逐步上升的趋势。

## 4.2 ARIMA模型

### 4.2.1 查看ACF和PACF

ACF：<acf.pdf>

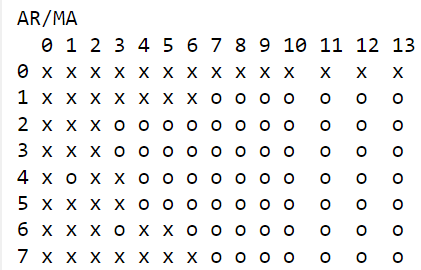
PACF：<pacf.pdf>

观察ACF和PACF确定MA和AR参数：看PACF可以确定AR参数可能为2或3，而ACF拖尾，认为MA(0)。

### 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）：

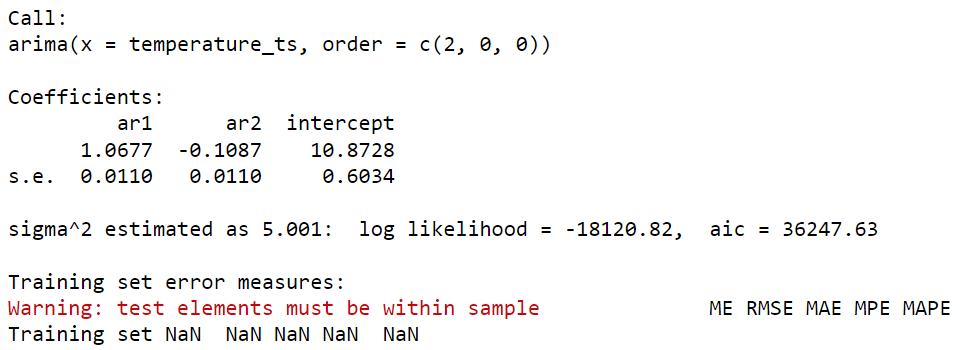
p-value<0.05



参数可能为ARMA(2,3) 。

### 4.2.3 最优子集

ARMA(2,0,0)

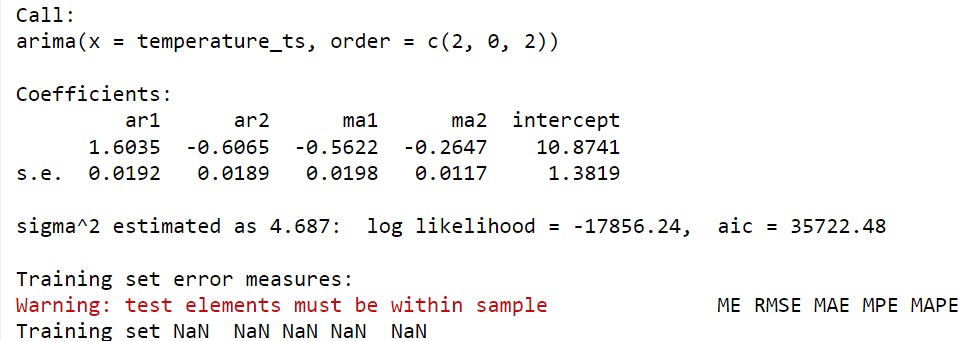


ARMA(2,0,1)

文本

描述已自动生成

ARMA(2,0,2)



ARMA(2,0,3)

文本

中度可信度描述已自动生成

ARMA(3,0,0)

文本

描述已自动生成

ARMA(3,0,1)

文本

中度可信度描述已自动生成

ARMA(3,0,2)

文本

描述已自动生成

ARMA(3,0,3)

文本

描述已自动生成

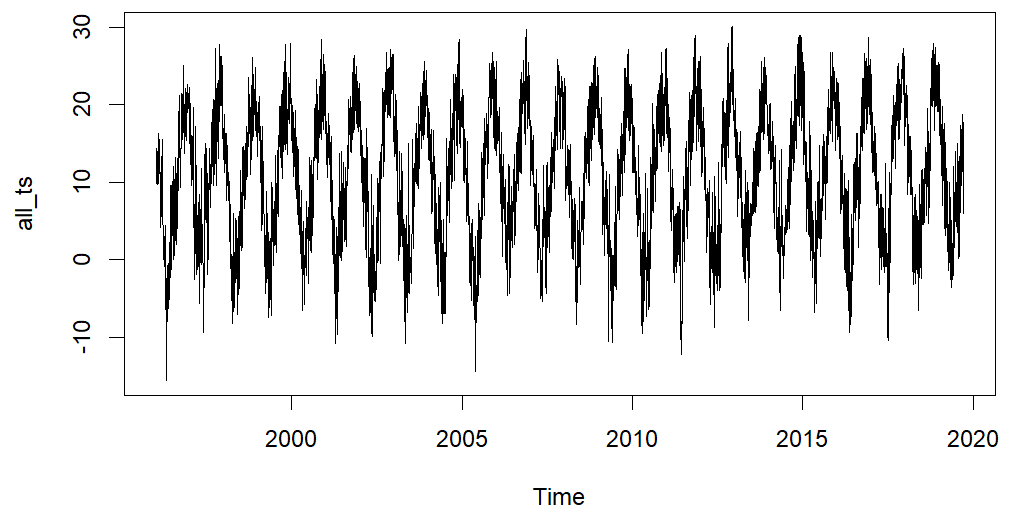
参数可能为ARMIA(2,0,3)

使用auto.arima函数确定：参数为ARIMA(3,0,0)(0,1,0)

### 4.2.4 预测

Forecast：[ARIMA(3,0,0)(0,1,0)[365].pdf](ARIMA(3,0,0)(0,1,0)%5b365%5d.pdf)

对比测试集：



残差-时间图：

图表, 散点图

描述已自动生成

qq图

图表, 折线图

描述已自动生成

预测效果比较好。

## 4.3 季节ARIMA

对变量进行一次差分，消除增长趋势，使模型简单。

图表

描述已自动生成

做ACF，PACF

ACF：

图表

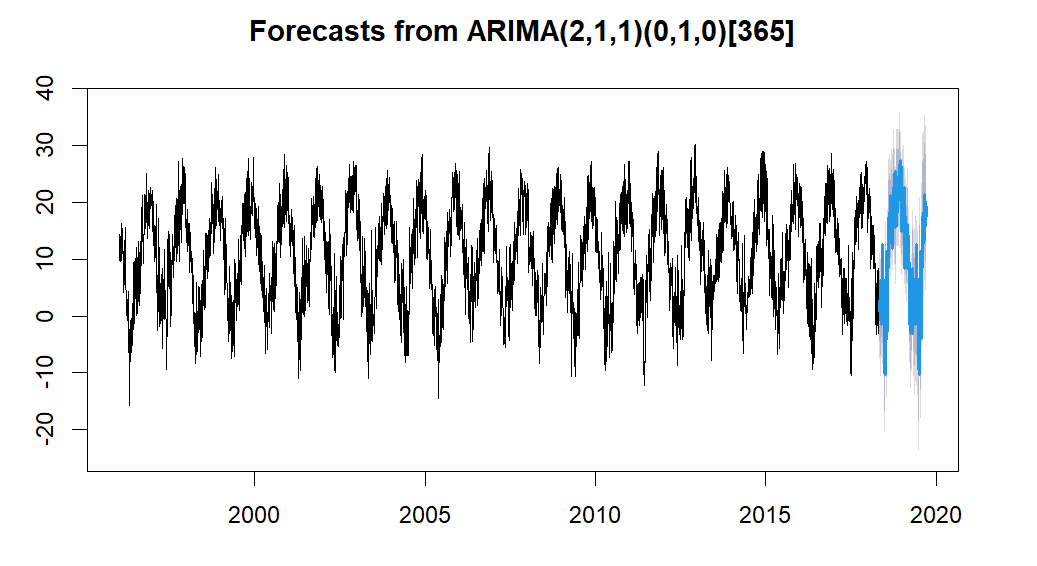
描述已自动生成

PACF：

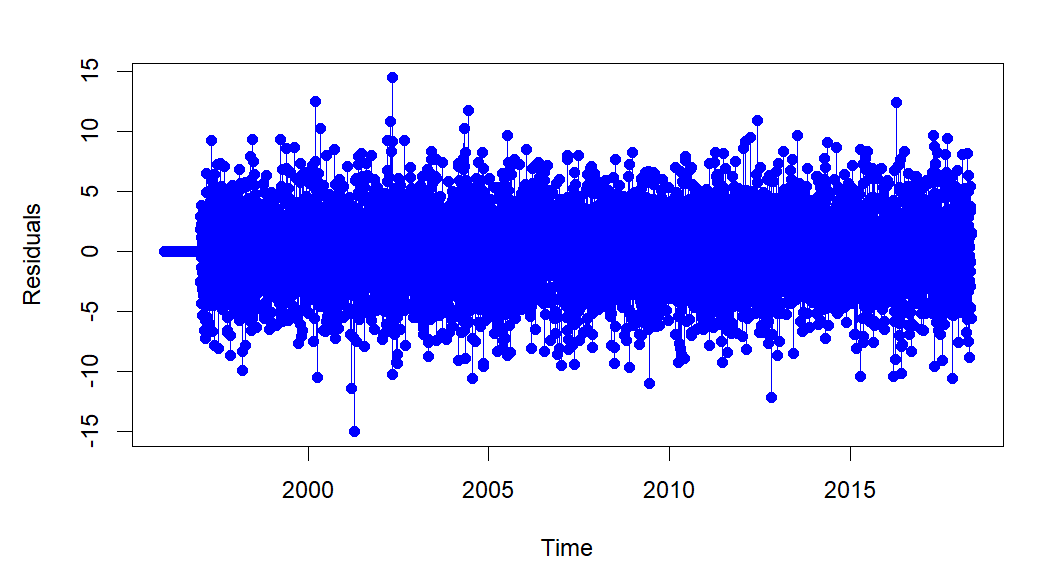
图表

描述已自动生成

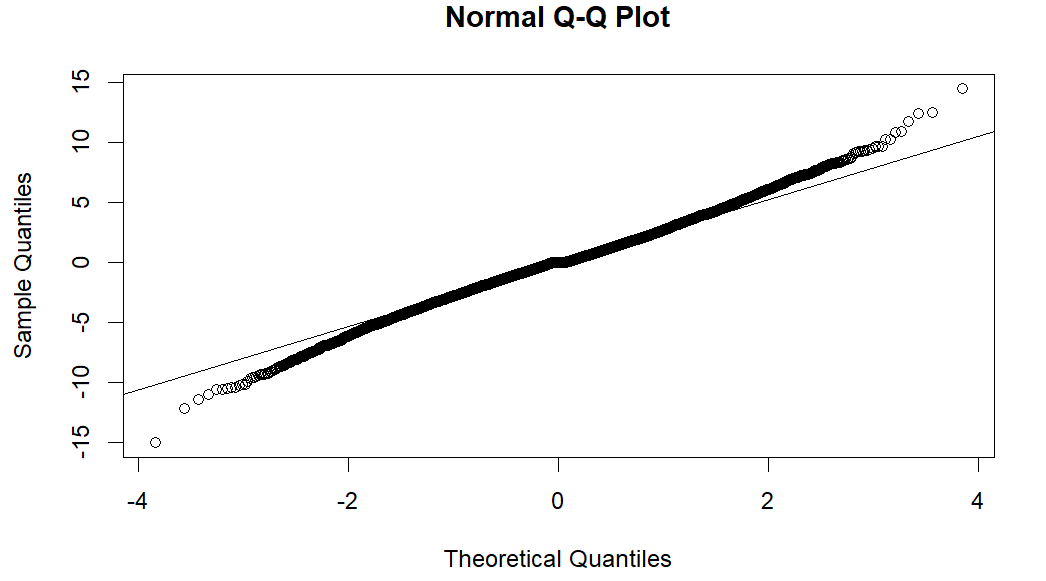
现在变量几乎不具有自相关性。尝试ARMIA（2,1,1）（0,1,0）



查看残差-时间图：



残差值比较大，检查qq图：



可以看出ARMIA模型对气温逐年上升的趋势拟合较好，但可以认为整体拟合效果较差。

# 第五章 分析与结论

通过对维也纳气温数据的全面分析和建模，我们得出了以下几点结论：

1. 气候特征： 维也纳的气温数据呈现出明显的季节性变化，冬季较冷，夏季较热，春秋季节温差较小。同时，数据中还存在一定的年度变化趋势，整体呈现出气温逐年上升的趋势。

2. 建模方法选择： 在预测模型选择方面，线性模型和余弦模型在拟合维也纳气温数据的趋势上效果较差，未能捕捉到数据的复杂特征。相比之下，ARIMA模型能够更好地拟合趋势情况，但数据准确性较差。

3. ARIMA模型预测： 通过ARIMA模型的预测结果显示，维也纳的气温在未来仍然会呈现出逐年上升的趋势，但变化幅度可能会有所减缓。预测结果对未来气温变化趋势提供了重要参考，为城市规划和气候适应提供了科学依据。

综上，本研究为理解维也纳的气候变化趋势提供了重要的参考，为城市规划和气候适应提供了科学依据，同时也为未来类似研究提供了借鉴和启示。。

# 参考文献

[1] [01tu01/Applied-Time-Series-Analysis-Wangyan: 应用时间序列分析 第五版 王燕 课后习题代码 (github.com)](https://github.com/01tu01/Applied-Time-Series-Analysis-Wangyan)

[2][hehaha68/USTC\_2022Spring\_Time-Series-Analysis-B: 中科大2022春《时间序列分析B》课程资源 (github.com)](https://github.com/hehaha68/USTC_2022Spring_Time-Series-Analysis-B)

[3][【R语言】ARMA时间序列分析上机案例\_时间序列分析arma例题-CSDN博客](https://blog.csdn.net/suansuannainaizi/article/details/124428975?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171388693916800182197290%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=171388693916800182197290&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-1-124428975-null-null.142%5ev100%5econtrol&utm_term=%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E5%88%86%E6%9E%90ARMI&spm=1018.2226.3001.4187)

[4][时间序列预测模型实战案例(六)深入理解机器学习ARIMA包括差分和相关性分析\_时间序列分析arima模型例题-CSDN博客](https://blog.csdn.net/java1314777/article/details/134033087?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E5%88%86%E6%9E%90ARMI&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-134033087.142%5ev100%5econtrol&spm=1018.2226.3001.4187)

[5][时间序列 ACF 和 PACF 理解、代码、可视化-CSDN博客](https://blog.csdn.net/fengdu78/article/details/136093967?ops_request_misc=&request_id=&biz_id=102&utm_term=ACF&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-136093967.142%5ev100%5econtrol&spm=1018.2226.3001.4187)

[6][时间序列分析技巧（一）：根据ACF、PACF进行AR、MA、ARMA模型选择\_acf和pacf拖尾用什么模型-CSDN博客](https://blog.csdn.net/jd1813346972/article/details/136466371?ops_request_misc=&request_id=&biz_id=102&utm_term=ACF&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-6-136466371.142%5ev100%5econtrol&spm=1018.2226.3001.4187)

[7][理解时间序列的ACF与PACF\_acf pacf-CSDN博客](https://blog.csdn.net/SunJW_2017/article/details/126993853?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171393486716800186519793%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171393486716800186519793&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-126993853-null-null.142%5ev100%5econtrol&utm_term=ACF&spm=1018.2226.3001.4187)

[8][R语言--模型的定阶、估计和显著性检验\_tsa包armasubsets函数-CSDN博客](https://blog.csdn.net/qq_67724979/article/details/127938691?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171444309316800225520202%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171444309316800225520202&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-127938691-null-null.142%5ev100%5econtrol&utm_term=%E6%89%A9%E5%B1%95%E8%87%AA%E7%9B%B8%E5%85%B3%E5%87%BD%E6%95%B0&spm=1018.2226.3001.4187)

[9] [时间序列(四) 预测-CSDN博客](https://blog.csdn.net/weixin_30945039/article/details/95268638?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171444309316800225520202%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171444309316800225520202&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-4-95268638-null-null.142%5ev100%5econtrol&utm_term=%E6%89%A9%E5%B1%95%E8%87%AA%E7%9B%B8%E5%85%B3%E5%87%BD%E6%95%B0&spm=1018.2226.3001.4187)

[10][Daily Temperature of Major Cities (kaggle.com)](https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities/code)

[11][时间序列预测(一)-季节性Arima建模 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/595411416)

[12][ARIMA模型确定(p,d,q)的几个规则 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/362962545)

[13][移动平均(Moving Average) - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/151786842)

[14][(13 封私信 / 80 条消息) 如何通过acf和pacf初步判断序列是否平稳？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/594486033/answer/3002575549)

[15][时间序列分析101：序言 | TimeSeriesAnalysis101 (gitbook.io)](https://skywateryang.gitbook.io/timeseriesanalysis101)

# 附录

1. R代码（Temperature.R）

# 加载包

library(TSA)

library(csv)

library(stats)

library(astsa)

library(forecast)

library(tseries)

library(urca)

# 读取CSV文件

temperature\_data <- read.csv("F:\\Desktop\\时间序列分析\\Temperature\_Vienna\_OnlyCelsius.csv")

# 合并 month, day, year 列

temperature\_data$date <- as.Date(paste(temperature\_data$Year,temperature\_data$Month, temperature\_data$Day, sep = "-"))

# 删除原始的 month, day, year 列

temperature\_data <- subset(temperature\_data, select = -c(Month, Day, Year))

# 显示更新后的数据集的前几行

head(temperature\_data)

# 筛选出所有值为-99的行

filtered\_data\_1 <- temperature\_data[temperature\_data$Celsius\_Temperature == -72.77777778, ]

# 显示筛选后的数据集

print(filtered\_data\_1)

# 26条缺失值，手动补数据

temperature\_data[835, "Celsius\_Temperature"] =-2.740740741

temperature\_data[836, "Celsius\_Temperature"] =-4.481481481

temperature\_data[841, "Celsius\_Temperature"]=-3.888888889

temperature\_data[842, "Celsius\_Temperature"]=-5.166666667

temperature\_data[852, "Celsius\_Temperature"]=3.138888889

temperature\_data[2107, "Celsius\_Temperature"]=23.1

temperature\_data[2108, "Celsius\_Temperature"]=23.4222222

temperature\_data[2109, "Celsius\_Temperature"]=23.7444444

temperature\_data[2110, "Celsius\_Temperature"]=24.0666667

temperature\_data[4004, "Celsius\_Temperature"]=19.1388889

temperature\_data[4397, "Celsius\_Temperature"]=11.66666667

temperature\_data[4594, "Celsius\_Temperature"]=15.83333333

temperature\_data[6358, "Celsius\_Temperature"]=1.083333333

temperature\_data[7051, "Celsius\_Temperature"]=-0.407407407

temperature\_data[7052, "Celsius\_Temperature"]=-1.314814815

temperature\_data[7178, "Celsius\_Temperature"]=12.5

temperature\_data[7317, "Celsius\_Temperature"]=13.72222222

temperature\_data[7368, "Celsius\_Temperature"]=2.444444444

temperature\_data[7787, "Celsius\_Temperature"]=5

temperature\_data[8101, "Celsius\_Temperature"]=7.166666667

temperature\_data[8102, "Celsius\_Temperature"]=5

temperature\_data[8262, "Celsius\_Temperature"]=14.97222222

temperature\_data[8284, "Celsius\_Temperature"]=8.32222222

temperature\_data[8285, "Celsius\_Temperature"]=9.97777778

temperature\_data[8286, "Celsius\_Temperature"]=11.63333333

temperature\_data[8287, "Celsius\_Temperature"]=13.28888889

filtered\_data\_2 <- temperature\_data[temperature\_data$Celsius\_Temperature == -72.77777778, ]

# 显示筛选后的数据集

print(filtered\_data\_2)

###至此，数据无错误

# 分训练集，测试集

# 创建训练集（2019-01-01之前）和测试集（2019-01-01之后）

train\_data <- subset(temperature\_data, date < "2019-01-01")

test\_data <- subset(temperature\_data, date >= "2019-01-01")

# 显示训练集和测试集的行数

cat("训练集观测数量：", nrow(train\_data), "\n")

cat("测试集观测数量：", nrow(test\_data), "\n")

#画原始数据

plot(train\_data$date,train\_data$Celsius\_Temperature ,pch =16 , col = "RED", cex = 0.2)

# 将数据集转换为时间序列对象

temperature\_ts <- ts(train\_data$Celsius\_Temperature,start = c(1996,9,11),frequency = 365)

# 显示时间序列对象的摘要信息

print(summary(temperature\_ts))

# 季节性分解

seasonal\_decompose\_m <- decompose(temperature\_ts, type = c( "multiplicative"))

seasonal\_decompose\_a <- decompose(temperature\_ts, type = c("additive"))

# 绘制季节分解图

pdf("seasonal.pdf",width = 32, height = 4)

plot(seasonal\_decompose\_m)

plot(seasonal\_decompose\_a)

#明显的周期性

# 进行ADF检验

adf\_test <- adf.test(temperature\_ts)

print(adf\_test)

# p值远远小于0.05，表示拒绝原假设，即时间序列是平稳的

# 绘制自相关函数（ACF）图

pdf("acf\_365.pdf", width = 6, height = 8)

acf(temperature\_ts, lag = 365 \* 20, main = "Autocorrelation Plot")

#明显的周期性.

# 绘制偏自相关函数（PACF）图

pdf("pacf\_365.pdf", width = 32\*4, height = 8)

#图要展开画，要不然就是一坨黑

pacf(temperature\_ts, lag = 365 \* 20, main = "Partial Autocorrelation Plot")

#明显的周期性.

#数据在ADF检验时已经确认平稳，试试一次差分什么情况

# 对温度时间序列数据进行一次差分

diff\_temperature <- diff(temperature\_ts)

# 绘制差分后的时间序列图

plot(diff\_temperature, main = "First Difference of Temperature Time Series", ylab = "Differenced Temperature")

# 进行ADF检验

adf\_test <- adf.test(diff\_temperature)

print(adf\_test)

# 绘制自相关函数（ACF）图

acf(diff\_temperature, lag = 365 \* 20, main = "Autocorrelation Plot")

# 绘制偏自相关函数（PACF）图

pacf(diff\_temperature, main = "Partial Autocorrelation Plot")

#明显周期性

#看看趋势如何

#线性趋势

# 执行线性趋势分析

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

# 显示回归模型的摘要信息

summary(trend\_lm)

# 绘制数据和趋势线

plot(temperature\_ts, main = "Temperature Time Series with Linear Trend", ylab = "Temperature (Celsius)")

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

#余弦趋势模拟？

library(TSA)

har=harmonic(temperature\_ts,1)

ml=lm(temperature\_ts~har)

summary(ml)

pdf("余弦趋势模拟.pdf", width = 32, height = 4)

plot (ts (fitted(ml), freq=365,start=c(1996,9,11)),

ylab= 'Temperature', type='l',

ylim=range (c(fitted (ml) , temperature\_ts)),col="red" ) ; points (temperature\_ts ,pch =16 , col = "blue", cex = 0.2)

#效果还行？

#残差-时间图

pdf("残差-时间图.pdf", width = 32, height = 4)

plot (y=rstudent (ml), x=as.vector (time (temperature\_ts) ) ,

xlab= 'Time', ylab='Standardized Residuals', type='o',pch =16 , col = "blue", cex = 0.2)

#±4°C的差距还是比较大

#残差是否符合正态

hist(rstudent(ml),xlab="标准残差")

qqnorm(rstudent(ml))

#拟合一下

plot(forecast(ml))

#ARMA

# 绘制自相关函数（ACF）图

pdf("acf\_365.pdf", width = 6, height = 8)

acf(temperature\_ts, lag = 365 \* 20, main = "Autocorrelation Plot")

#明显的周期性.

# 绘制偏自相关函数（PACF）图

pdf("pacf\_365.pdf", width = 32\*4, height = 8)

#图要展开画，要不然就是一坨黑

pacf(temperature\_ts, lag = 365 \* 20, main = "Partial Autocorrelation Plot")

#明显的周期性.

#AR(3),diff(0),MA(0)

#扩展自相关

eacf(temperature\_ts)

#看不出来啊

#最优子集(这个函数运行太慢了，我手动进行一下)

library(forecast)

best\_aic <- Inf

best\_order <- c(0, 0, 0)

# 迭代尝试各种参数组合

for (p in 0:3) {

  for (d in 0:2) {

    for (q in 0:2) {

      # 拟合ARIMA模型

      model <- Arima(temperature\_ts, order = c(p, d, q), seasonal = list(order = c(0, 1, 0)))

      # 计算AIC

      aic <- AIC(model)

      # 更新最佳参数和AIC

      if (aic < best\_aic) {

        best\_aic <- aic

        best\_order <- c(p, d, q)

      }

    }

  }

}

# 打印最佳参数

print(best\_order)

#自动arima，aic最小认为合适

#library(forecast)

best\_fit <- auto.arima(temperature\_ts, ic = "aic", seasonal = TRUE)

summary(best\_fit)

#手动操作，d=0已经确定，acf说ar应该在2,3附近

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

summary(arima\_model)

arima\_model <- arima(temperature\_ts, order = c(2,0,1))

summary(arima\_model)

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

summary(arima\_model)

arima\_model <- arima(temperature\_ts, order = c(2,0,3))

summary(arima\_model)

arima\_model <- arima(temperature\_ts, order = c(3,0,0))

summary(arima\_model)

arima\_model <- arima(temperature\_ts, order = c(3,0,1))

summary(arima\_model)

arima\_model <- arima(temperature\_ts, order = c(3,0,2))

summary(arima\_model)

arima\_model <- arima(temperature\_ts, order = c(3,0,3))

summary(arima\_model)

#看下来是（2,0,3）

#季节ARIMA

#一次差分

temperature\_ts\_diff <- diff(temperature\_ts)

# 打印差分后的前几个观测值

print(head(temperature\_ts\_diff))

plot(ts\_diff)

acf(ts\_diff,lag=365\*20)

pacf(ts\_diff,lag=365\*20)

arima\_model <- arima(temperature\_ts, order = c(2,1,1),seasonal=list(c(0,1,0)))

# 使用拟合的ARIMA模型进行未来499个值的预测

forecast\_values <- forecast(arima\_model, h = 499)

# 加载 test\_data 数据集

# 请确保你已经加载了 test\_data 数据集，以便进行比较

# 绘制预测值和测试数据的对比图

plot(forecast\_values)

lines(test\_data, col = "red")

# 计算残差

residuals <- residuals(arima\_model)

# 绘制残差-时间图

plot(residuals, xlab = "Time", ylab = "Residuals", type = "o", pch = 16, col = "blue")

# 绘制QQ图

qqnorm(residuals)

qqline(residuals)