华东理工大学 2023 –2024 学年第 2 学期

《大数据与金融计算》实验报告

实验名称　中国股市收益率分布的实证分析

实验目的/要求
1、掌握 Python 基本知识（变量、编程、Numpy、Matplotlib 和 Pandas）。 2、掌握中国股市收益率的分布特征，会利用 Python 分析和计算收益率分布。
实验内容
1. 认真阅读三篇文献资料（自己也可以下载相关文献进行阅读），了解股票市场收益率分布的重要性和形态特征，重点关注收益率分布的研究思路和研究方法。 2. 在阅读文献的基础上，结合提供的上证指数分钟数据，尝试完成如下计算： （1）根据上证指数分钟数据，分布计算不同时间尺度（1 分钟、5 分钟、10 分钟、30 分钟、60 分钟、120 分钟、240 分钟）的收益率序列； （2）计算不同尺度收益率的经验概率密度，并画在同一幅图上，比较不同尺度收益率分布的区别； （3）计算不同尺度收益率的正态拟合分布，比较他们对正态分布偏离程度； （4）考察不同尺度收益率的尾部分布特征，是否满足幂律分布？是否服从“负三次方定律”？ **提示一：经验概率分布计算：将分析样本覆盖的范围分成若干区间，即$[r_0, r_1)\cup[r_1, r_2)\cup[r_2, r_3)\cup[r_3, r_5)\cup\ldots\cup[r_{i-1}, r_i)\cup\ldots\cup[r_{n-1}, r_n]$，再分别计算每个区间内的样本个数 y_i，进而计算概率 $p_i=y_i/(r_i-r_{i-1})/m$，其中 m 为样本总个数。画图时，区间概率 p_i 对应 r_i 可以用区间的中点表示。为了使经验概率分布图形美观，若画图使用线性坐标（x 轴），可以线性均分样本范围；若画图使用对数坐标（x 轴），可对数等分样本范围。分割的份数自主选择，一般以画图美观为准。** **提示二：所有图形按照画图标准流程（画图、修饰、输出）进行。**
实验总结
请提供对本次实验结果的讨论分析，以及实验的心得和体会。包括对知识点的掌

握，算法的理解，以及对理论课程和实验课程改进的建议。（不少于 500 字）

本文收集了上证指数 1min，5min，10min，30min，60min，120min，240min 这 7 个不同时间尺度的指数数据，对其对数收益率进行研究。经过实证发现

（1）各个时间尺度收益率特征可知随着时间尺度增大(从 1 分钟到 240 分钟)，其分布的峰逐渐变缓，而正负尾逐渐变厚。从该特征中可以推断随着时间尺度增大，极端高(低)收益的出现频率更高。

（2）不同时间尺度下收益率数据均偏离与正态分布。与正态分布相比，每个数据都呈现了不同程度的尖峰厚尾特征，时间尺度越小，该特征就越明显。

（3）在不同时间尺度下的收益率正负尾都有一段分布都近似于直线，即满足幂律分布，并且满足负三次方定律。

经过这个实验，我掌握了

（1）python 编程知识，包括数据处理常用的 numpy 和 matplotlib 库的基本操作

（2）python 有关收益率的计算：包括使用 np.log()矩阵相减法计算对数收益率、如何划分等差区间计算经验概率密度和如何对正态性和幂律分布以及负三次方定律进行检验的知识。

（3）正态分布和幂律分布的数学知识。

实验的心得与体会：

（1）掌握了 Numpy 数组筛选的新方法，numpy 的条件判断语句，比如 data[data>0]等使用起来比循环更加灵活便捷。

（2）增长了对于收益率分布的认知，过去金融知识常以收益率正态分布作为假设，而经过这个实验让我直观的看到了收益率的分布情况以及特征。

课程改进建议：

（1）可以对 python 知识讲解的更详细一点，比如展示一个函数不同情况下的用法。

教师批阅：　　　　　　　　　　　　　　　　　　　　实验成绩：   　　　　　　　　　　　　　　　　　　　教师签名:　　　　　　日期：

实验报告正文：

（每次实验报告均为一篇小论文，因此，统一按照学术论文的要求完成实验报告正文，应包括：题目、摘要、文献综述、模型和方法、结果和讨论、参考文献、附录，具体格式如下：

中国股市收益率的分布特征

摘要：股市收益率分布特征长期以来都是金融领域的热门研究议题，许多金融理论基于收益率正态分布的假设，但是越来越多实证研究证明收益率并非正态分布。为了验证收益率的统计分布问题，本文收集了上证指数 1min，5min，10min，30min，60min，120min，240min 这 7 个不同时间尺度的指数数据，对其对数收益率进行研究。经过实证分析发现不同时间尺度下的收益率分布均偏离正态分布，呈现尖峰厚尾的特征。此外还对尾部收益率数据的分布进行了研究，发现其满足幂律分布，并且符合负三次方定律。

关键词：股市收益率; 上证指数; 正态分布; 幂律分布

1 文献综述

收益率分布特征一直是金融与投资领域的核心问题之一，对收益率分布特征的研究层出不穷。在过去金融的理论研究中，资产收益率被认为一个近似服从正态分布的随机变量。基于该理论，研究者提出了 CAPM[1]，布莱克-舒尔斯期权定价模型[2]等。

随着金融的发展，有一部分学者尝试将其他领域的理论和模型融入金融领域，取得了许多意义重大的研究成果。S. Ghashghaie 等对外汇市场价格变化与时间延迟的概率密度图像进行研究，发现其与流体力学中的湍流的概率分布图像类似，揭示了外汇市场的规律。 [3]

在此背景下，关于收益率分布的研究有了进一步的发展。Parameswaran Gopikrishnan 等将统计物理学中的标度概念引人金融市场。分析了美国标准普尔 500 指数、日本日经指数和香港恒生指数的标度特性，发现它们的收益分布都具有标度不变性和幂律特性。[4] Gabaix Xavier 等也揭示了收益率尾部的幂律分布特征。[5]

在国内市场中，也有学者对国内市场收益率分布进行研究。都国雄等对上证指数和深圳成指的高频数据进行实证分析，计算其对数收益率的概率分布，通过图示法对比、特征指数与利维指数的计算，证明了上证指数和深圳成指可以由利维分布和幂律分布的组合进行描述，其满足尖峰胖尾的特征，与正态分布偏离[6]。陈收等对上证指数时间尺度为 5min 和 1 天的收益率分布进行研究，分析得出收益率中心部分服从利维分布，而收益率尾部分布服从幂律分布，但是不同于成熟市场，不满足负三次方定律[7]。

综合上述文献对于收益率分布的研究，这些实证结果都证明了收益率偏离正态分布，尾部满足幂律分布。本文基于此思路，对上证指数多重时间尺度数据(1 分钟，5 分

钟，10 分钟，30 分钟，60 分钟，120 分钟，240 分钟)对数收益率的分布特征进行实证研究，发现这 7 个时间尺度的收益率数据均偏离正态分布，并且尾部满足幂律分布。

2 模型和方法

2.1 对数收益率

对数收益率是指资产从一个时间段结束时的价格到下一个时间段结束时价格的对数之差，通常以自然对数 e 为底，具体到股市指数，在 t 时刻，时间段间隔为 n 的指数对数收益率是 t 时刻的指数与 t-n 时刻指数的对数之差，公式为：

$$R_t = ln(P_t) - ln(P_{t-n}) \tag{1}$$

式中R_t是 t 时刻的对数收益率，P_t是 t 时刻的指数，P_{t-n}是据 j 时刻 n 个时间单位前的指数数据。

在 Python 中对数收益率可以使用矩阵相减的方式计算，其计算原理为：基于原始数据(时间段为 1 到 t)，计算得到两个矩阵，一个时间段从 2 到 t，一个为从 1 到 t-1，将这两个矩阵分别使用 np.log()进行对数化，最后相减得到对数收益率序列。

2.2 经验概率

经验概率是特定事件的发生的次数与总样本数之比，其公式为：

$$P = \frac{M_A}{N} \tag{2}$$

式(2)中M_A是特定事件 A 发生的次数，N是总样本数，P是经验概率。

经验概率分布不同与正态分布，t 分布等，其没有事先的理论基础，而是直接通过计算实际数据得到，用以描述给定样本值的概率分布情况。

在股市收益率数据中计算概率密度的方法为：假设样本值为$(r_1,r_2,r_3,\ldots,r_n)$，在其中取得样本最小值记为r_{min}，样本的最大值记为r_{max}，将区间[r_{min}，r_{max}]等分为 n 个子区间。对于每一个样本值，计算其数值位于哪一个子区间中，将目标子区间的频次加 1，即可算出每个区间的频次，最后使用该频次除以样本总数再除以区间长度即可得到经验概率密度。其公式为：

$$[\,p_1\ p_2\ p_3 \cdots p_n\,] = \frac{[\,f_1\ f_2\ f_3 \cdots f_n\,]}{N \times (b_{i+1} - b_i)} \tag{3}$$

式(3)中，$[\,f_1\ f_2\ f_3 \ldots f_n\,]$是各个区间按照前述计算方法所得的频次矩阵，$N$是样本总数，

$(b_{i+1}-b_i)$是相邻两个区间端点之差，即区间长度，各个区间等差，$[\,p_1\ p_2\ p_3 \dots p_n\,]$是经验概率密度矩阵。

在 python 中可以使用 Numpy 库计算经验概率密度，具体流程为：

1) 对收益率区间进行等差划分，使用的函数是 np.linspace(start, stop, num)，表示在区间 start 到 stop 内划分出 num-1 的等差区间。

2) 生成和区间数对应的全 0 一维矩阵，使用的函数是 np.zeros(shape)，其可以根据输入的 shape 数据生成对应维度的全 0 矩阵

3) 使用两层循环计算区间频次，外层循环是收益率数据，内层循环是区间，如果收益率在区间中则让记录频次的全 0 矩阵对应位置的数值加 1.

4) 使用 np.diff()函数计算区间两个端点之差，由于在这里使用的是等差区间，也可以使用任意两个区间端点之差作为替代。根据公式计算出经验概率。

2.3 正态分布

正态分布是一种常见的概率分布，如果随机变量 X 的概率密度函数满足：

$$f(x;\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4}$$

则称随机变量 X 满足一维正态分布。在式(4)中，σ是正态分布的标准差，μ是正态分布的平均值。

正态分布在金融领域中应用十分广泛，大量金融理论基于正态分布，因此对于数据是否满足正态分布的检验十分重要。正态性检验方法可分为图示法和解析法，图示法中可以绘制出数据的检验分布图像与对应正态分布图像进行对比。解析法主要是 JB 检验(Jarque-Bera test)，其通过计算数据的峰度和偏度来定量比较数据分布和正态分布[8]，计算公式为：

$$Skewness=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2)^{1.5}} \tag{5}$$

$$Kurtosis=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2)^2} \tag{6}$$

式(5)为偏度公式，式(6)为峰度公式，其中x_i是数据样本，$\bar{x}$是样本均值。

JB 检验的计算公式如下：

$$JB = \frac{Skewness^2}{6/n} + \frac{(Kurtosis-3)^2}{24/n} \quad (7)$$

式(7)中 n 为样本量，Skewness 是偏度，Kurtosis 是峰度。

若变量服从正态分布，则可以得到偏度为 0，峰度为 3，因而 JB 统计量的值为零；如果 JB 统计量的值不满足假设检验结果，则可以认为数据不满足正态分布。

在 python 中可以使用 scipy.stats.norm.fit()拟合正态分布得到对应正态分布的均值和残差，接着通过 scipy.stats.norm.pdf()可以得到正态分布概率密度值。

2.4 幂律分布

幂律分布是一种具有无标度特性的长尾分布，通常表现为一条先陡峭下降然后逐渐趋于平缓、延伸至横轴的曲线，其概率密度函数可以表示为

$$f(x) = cx^{-\alpha} \quad (8)$$

式(8)中α和 c 都是幂律分布中的参数。

在双对数坐标轴下，符合幂律分布的数据曲线是一条斜率为负的直线，由此可得判断幂律分布的方法：对 x 和 y 取对数得 lnx 和 lny，对 lnx 和 lny 利用最小二乘法进行线性回归，如果 x 和 y 之间满足幂律分布，则可以看到 x 和 y 数据分布近似于线性，而拟合直线的斜率就是幂律分布的特征指数，当该指数近似于-3 时，则认为其满足负三次方定律。

幂律分布主要涉及对数变换和线性回归，在 python 中可以使用 np.log()函数对 np 数组进行以 e 为底的对数变换；对于线性回归，可以使用 sklearn 库中的 LinearRegression 类，为该类创建对象后，可以使用 fit()方法进行，可以通过 intercept_和 coef_获取拟合参数。

3 结果与讨论

3.1 数据介绍与处理

实验中使用了上证指数(SSEC)数据，数据本身以 1 分钟为时间频率，对数据进行清洗，去除收益率大于涨停板 10%和跌停板 10%的异常值后，计算得到不同时间尺度(1 分钟、5 分钟、10 分钟、30 分钟、60 分钟、120 分钟、240 分钟)的数据。对不同时间尺度的数据分别计算对数收益率，最终得到目标数据。

3.2 实证分析

3.2.1 上证指数对数收益率的经验概率

将上证指数数据划分为 100 个等差区间，对于每一个收益率样本，求其在对应区间出现的次数作为区间的频率，使用该频率除以区间长度再除以样本总数得到上证指数数据的经验概率密度，并在 x-logy 坐标轴中画出，如图 1 所示：

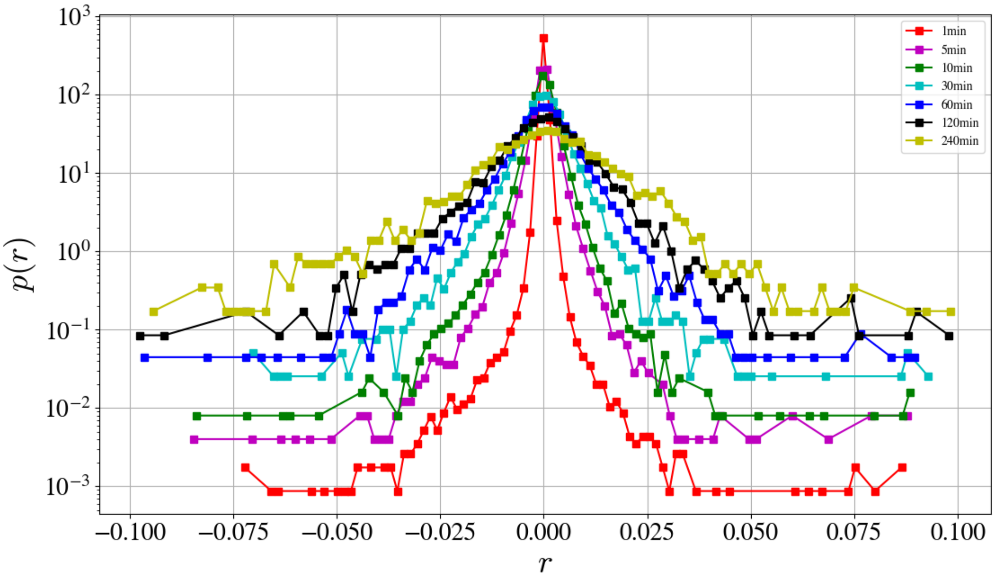
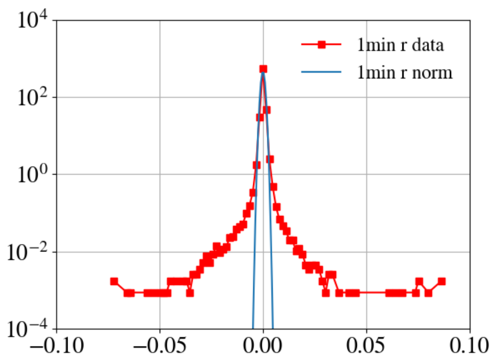


图 1 上证指数经验概率密度图

从图中可以看出不同时间尺度的收益率分布都呈现对称性，以 r=0.000 为对称轴，说明 r=0.000 出现的概率最高。同时观察比较各个时间尺度收益率特征可知随着时间尺度增大(从 1 分钟到 240 分钟)，其分布的峰逐渐变缓，而正负尾逐渐变厚。从该特征中可以推断随着时间尺度增大，极端高(低)收益的出现频率更高。推测可能是随着时间累积，股票指数收益率偏向于向两极前进。

3.2.2 收益率正态性检验

对上证指数对数收益率 1 分钟、5 分钟、10 分钟、30 分钟、60 分钟、120 分钟、240 分钟数据进行正态性检验，首先采用图示法，将数据的经验概率和对应拟合的正态分布图进行比较，结果如图 2 到图 8 所示，图例中 r data 为数据的经验概率分布， r norm 为拟合的正态分布。



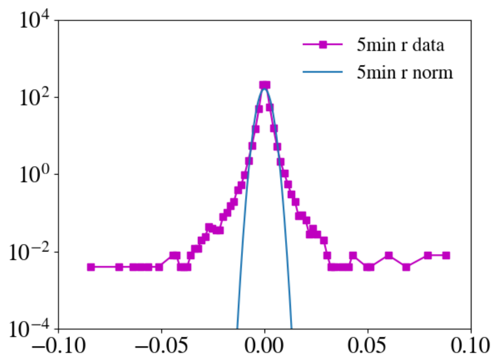


图 2　1min 收益率分布

图 3　5min 收益率分布

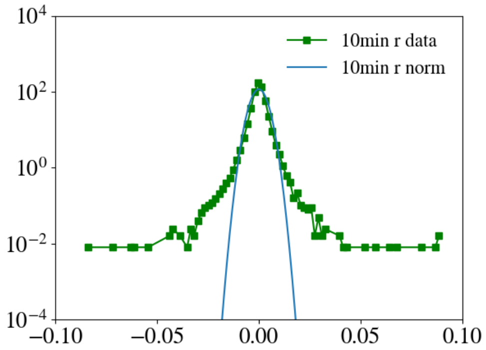


图 4　10min 收益率分布

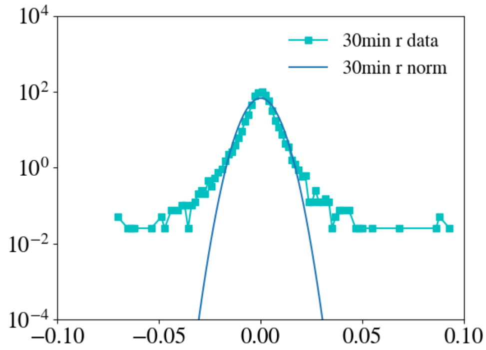


图 5　30min 收益率分布

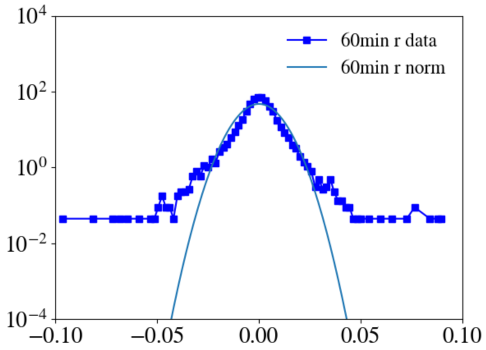


图 6　60min 收益率分布

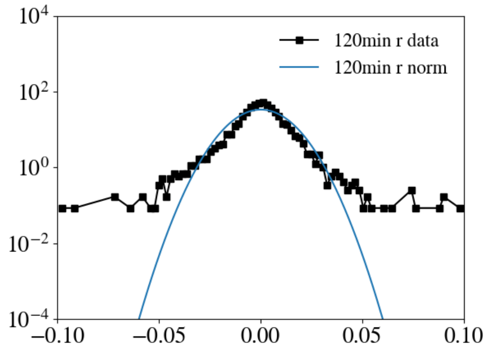


图 7　120min 收益率分布

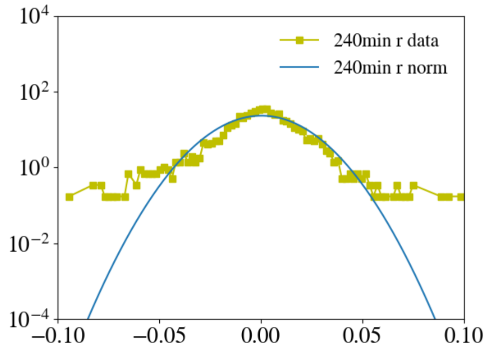


图 8　240min 收益率分布

从图 2 到图 8 中可以看出，不同时间尺度下收益率数据均偏离与正态分布。与正态分布相比，每个数据都呈现了不同程度的尖峰厚尾特征，时间尺度越小，该特征就越明显。

下面对收益率数据计算峰度和偏度，并使用 JB 检验定量检验数据的正态性，检验结果如表 2 所示：

表 2　收益率 JB 检验结果

收益率时间尺度	峰度	偏度	JB P-value
1 min	1287.506	1.539	0.000***
5 min	97.829	0.255	0.000***
10 min	54.362	0.339	0.000***
30 min	22.749	0.157	0.000***
60 min	15.084	-0.017	0.000***
120 min	10.534	0.049	0.000***
240 min	6.775	-0.255	0.000***

表中，*表示在 10%水平下显著，**表示在 5%水平下显著，***表示在 1%水平下显著

JB 检验的原假设为数据满足正态分布。由表 2 可以看出不同时间收益率数据 JB 统计量的 P 值全都在 1%水平下显著，由此可以拒绝原假设，收益率数据不满足正态分布。

3.2.3 幂律分布和负三次方定律检验

首先考察收益率正尾和负尾的分布，为了验证幂律分布的概率密度公式，如式 8 所示，将其分布在双对数坐标轴下画出，将概率密度公式变为式 9:

$$log(f(x)) = -alog(x) + log(c) \tag{9}$$

只要检验在双对数坐标轴下收益率尾部是否近似一条截距为正，斜率为负的直线即可检验尾部是否满足幂律分布。

收益率正半部分和负半部分在双对数坐标轴下的分布如图 9 和图 10 所示:

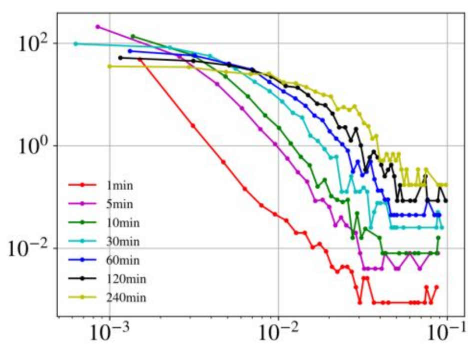


图 9　收益率正半部分

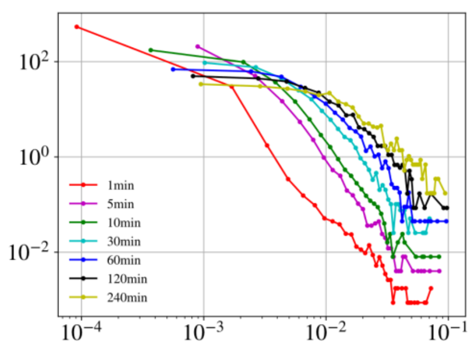


图 10　收益率负半部分

从图中可以无论是正半部分还是负半部分，在不同时间尺度下的收益率尾部都有一段分布都近似于直线，可以得到收益率尾部满足幂律分布。

下面对收益率尾部是否满足负三次方定律进行研究。研究选取不同时间尺度下收益率的尾部分布满足幂律分布(近似为直线)的部分，在此基础上对数据点进行线性回归得到斜率，检验斜率是否近似于-3。

对于正尾的斜率检验结果如表 3 所示。

表 3　正尾斜率检验结果

时间尺度	斜率
1 min	-2.485
5 min	-2.788
10 min	2.779
30 min	-2.895
60 min	-2.864
120 min	-2.721
240 min	-2.684

对于负尾的斜率检验结果如表 4 所示。

表 4　负尾斜率检验结果

时间尺度	斜率
1 min	-2.654
5 min	-3.057
10 min	-3.071
30 min	-3.111
60 min	-2.949
120 min	-2.714
240 min	-2.345

正尾的检验曲线图如图 11 到 17 所示：

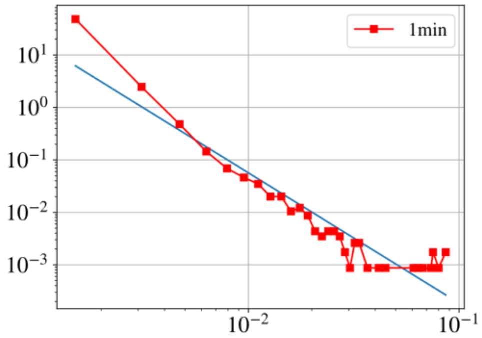


图 11　1min 正尾线性回归

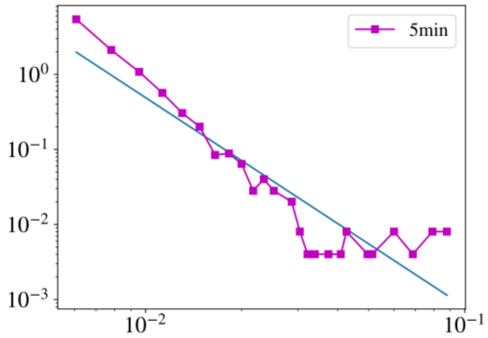


图 12　5min 正尾线性回归

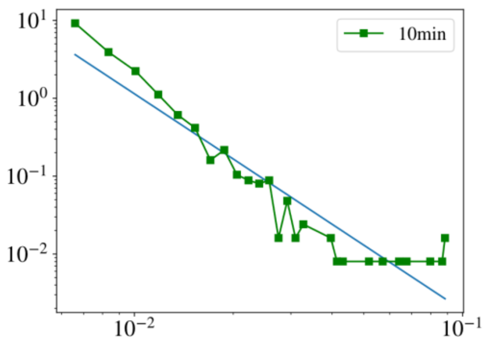


图 13　10min 正尾线性回归

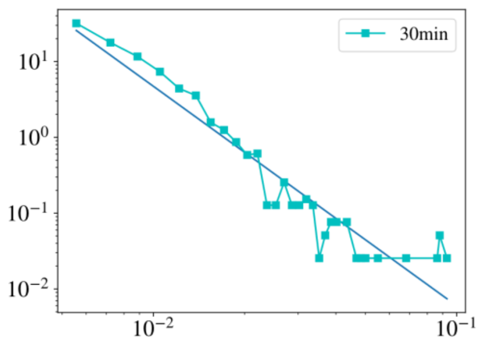


图 14　30min 正尾线性回归

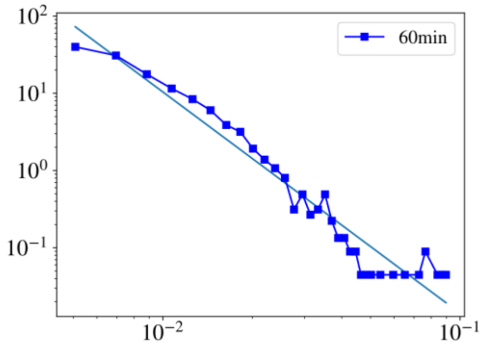


图 15　60min 正尾线性回归

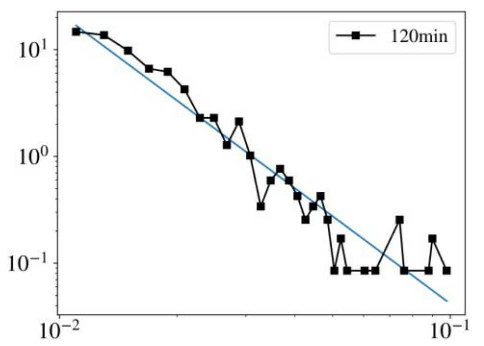


图 16　120min 正尾线性回归

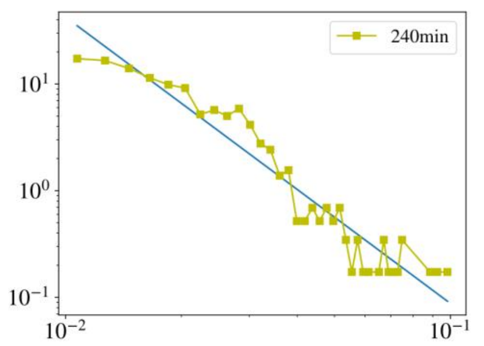


图 17　240min 正尾线性回归

负尾的检验曲线图如图 18 到 24 所示：

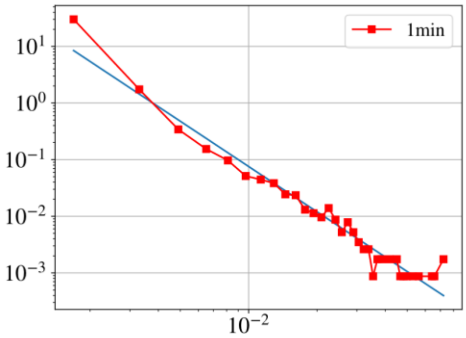


图 18　1min 负尾线性回归

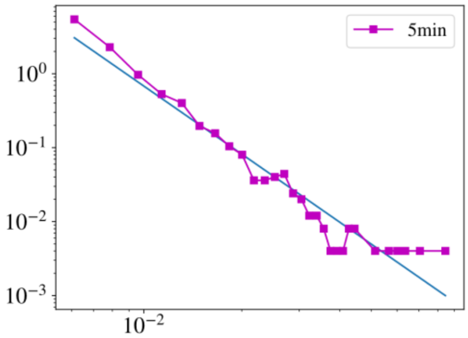


图 19　5min 负尾线性回归

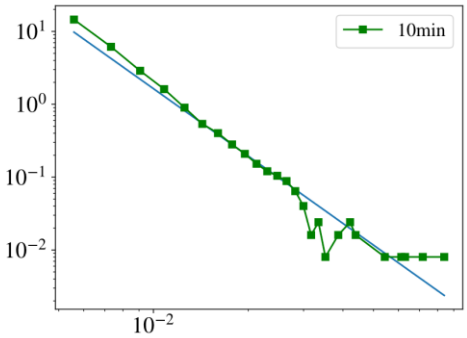


图 20　10min 负尾线性回归

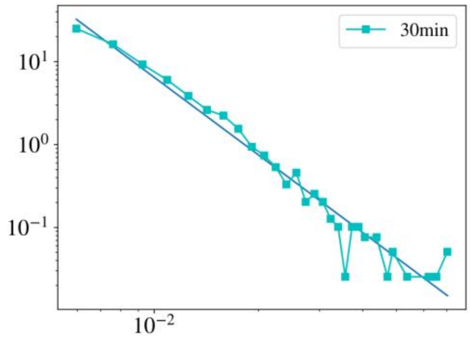


图 21　30min 负尾线性回归

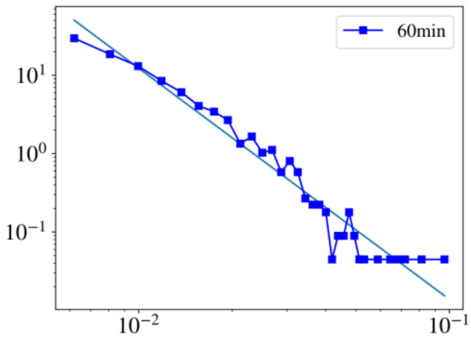


图 22　60min 负尾线性回归

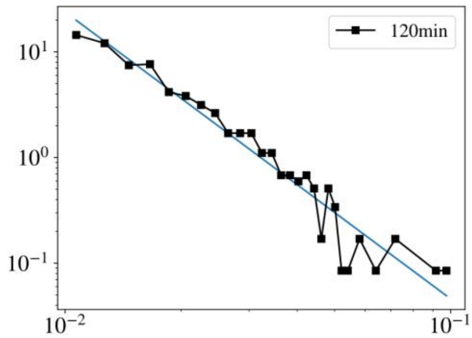


图 23　120min 负尾线性回归

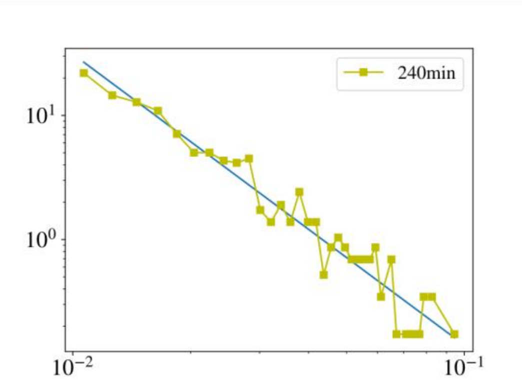


图 24　120min 负尾线性回归

由表 3 和表 4 可知。对于不同时间尺度的收益率数据，不论是正尾还是负尾，其满足幂律分布的部分斜率基本接近于-3，这证明了幂律分布以及负三次方定律在本文选取的上证指数从 1min 到 240min 为时间尺度的收益率数据中是成立的。

4 参考文献

[1] Sharpe, William F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk, Journal of Finance, 19 (3), 425-442

[2] Fischer Black and Myron Scholes. The Pricing of Options and Corporate Liabilities[J]. Journal of Political Economy, 1973, 81(3) : 637-654.

[3] S. Ghashghaie et al. Turbulent cascades in foreign exchange markets[J]. Nature, 1996, 381 : 767.

[4] Parameswaran Gopikrishnan，Vasiliki Plerou．Luls A．Nunes Amaral，et a1．Scaling of the distdhution on fluo—tuations of financial market indices[J]．Physical Review E，1999，60(5)：5305—5316．

[5] Gabaix Xavier et al. A theory of power-law distributions in financial market fluctuations.[J]. Nature, 2003, 423(6937) : 267-70.

[6] 都国雄,宁宣熙.我国股市收益概率分布的统计特性分析[J].中国管理科学,2007(05):16-22.DOI:10.16381/j.cnki.issn1003-207x.2007.05.017.

[7] 陈收,杨宏林,李双飞.中国股票市场多标度幂律特征和临界现象[J].中国管理科学,2008(03):8-15.DOI:10.16381/j.cnki.issn1003-207x.2008.03.027.

[8] 庞欢. 厚尾分布下我国证券市场风险测度研究[D].内蒙古财经大学,2022.DOI:10.27797/d.cnki.gnmgc.2022.000118.

5 附录

1.读取数据求对数收益率

```python
import numpy as np
import pandas as pd
import scipy.io as sio
from scipy import stats
import matplotlib.pyplot as plt
from sympy import re
#不同尺度收益率的数据
class DataMinDiff:
    def __init__(self,data_file) -> None:
        self.data_file = data_file
        self.data_min = None
    def getData(self):
        #从文件中读取数据
        mat_file = sio.loadmat(self.data_file)
        data_min = mat_file['p'][:,0]
        self.data_min = data_min

    def callLogR(self,minute):
        #获取数据
        data_min = self.data_min
        #根据指定的时间间隔获取数据
        data = data_min[::minute]
        data = data[data>0]
        #计算指数收益率
        r_data = np.log(data[1:])-np.log(data[:-1])

        #清洗数据
        r_data = r_data[~np.isnan(r_data)]
        r_data = r_data[(r_data >= -0.1) & (r_data <= 0.1)]
        return r_data


if __name__== "__main__":
    dataMinDiff = DataMinDiff("./dataset/SSEC_min.mat")
    r_data = dataMinDiff.callLogR(1)

    print(r_data.shape)
    plt.figure(figsize=(30,20))
    plt.plot(r_data)
    plt.grid(True)
    plt.show()
```

2.计算经验概率密度

```python
from xml.dom import minidom
import numpy as np
import pandas as pd
import scipy.io as sio
from scipy import stats
import matplotlib.pyplot as plt
from sympy import re
from DataMin import DataMinDiff


class EmpiricalProbability:
    def __init__(self) -> None:
        pass
    def getP(self,r_data,bin_num):
        bin = np.linspace(np.min(r_data),np.max(r_data),bin_num)
        #记录每个区间的频次
        f = np.zeros(len(bin)-1)
        mid = np.zeros(len(bin)-1)
        for r in r_data:
            for i in range(len(bin)-1):
                mid[i] = (bin[i]+bin[i+1])/2
                if r>=bin[i] and r<=bin[i+1]:
                    f[i] = f[i] + 1
        #计算经验概率
        p = f/np.diff(bin)/len(r_data)
        #数据清洗
        ind = p>0
        p = p[ind]
        f = f[ind]
        mid = mid[ind]
        return mid,p


```

3.总体画图

```python
def r_picture():
    for i,scale in enumerate(data_scale):
        #获取收益率
        r_data = data_min.calLogR(scale)
        #计算经验概率
        mid,p = ep.getP(r_data,101)
        #画图
        plt.semilogy(mid,p,'s-'+color[i],label=str(scale)+'min r')
        plt.legend(loc=0)
    plt.savefig('data.png')
    plt.show()

```

4.拟合正态分布

5.幂律分布:

```python
def r_norm():
    for i,scale in enumerate(data_scale):
        #获取收益率
        r_data = data_min.calLogR(scale)


        #拟合正态分布
        r_mean,r_std = stats.norm.fit(r_data)
        x_fit_min = np.linspace(-0.1, 0.1, 300)
        y = stats.norm.pdf(x_fit_min,loc = r_mean,scale = r_std)
        mid,p = ep.getP(r_data,101)


        #计算峰度和偏度
        k = stats.kurtosis(r_data)+3
        s = stats.skew(r_data)

        print(str(scale)+"分钟数据的数量为\n",len(r_data))
        print(str(scale)+"分钟数据的偏度为\n",s)
        print(str(scale)+"分钟数据的峰度为\n",k)
        JB_sta = (s**2) / (6/len(r_data)) + ((k-3)**2) / (24/len(r_data))

        p_value = 1 - stats.chi2.cdf(JB_sta, df=2)
        print(str(scale)+"分钟数据的JB统计量为\n",JB_sta)
        print(str(scale)+"分钟数据的JB统计量的p值为\n",p_value)


        plt.semilogy(mid,p,'s-'+color[i],label=str(scale)+'min r data')

        plt.semilogy(x_fit_min,y,label=str(scale)+'min r norm')
        plt.xlim([-0.1,0.1])
        plt.ylim([10e-5,10e3])
        plt.legend(frameon=False,loc="upper right",fontsize='small')
        plt.savefig('./img/'+'norm-'+str(scale)+'min.png')
        plt.show()
```

```python
def var_fu():
    for i, scale in enumerate(data_scale):
        # 获取收益率
        r_data = data_min.callogR(scale)
        mid, p = ep.getP(r_data, 101)
        linear = LinearRegression()
        #获取负半部分尾部
        ind = mid <= -(0.01)
        #获取正半部分尾部
        #ind = mid >= (0.01)

        x_mid = mid[ind]
        y_p = p[ind]
        x_mid = np.abs(x_mid)
        y_p = np.abs(y_p)

        #线性回归
        X = np.expand_dims(np.log(x_mid), axis=1)
        Y = np.expand_dims(np.log(y_p), axis=1)
        result = linear.fit(X, Y)
        y = linear.predict(X)
        intercept = result.intercept_
        k = result.coef_
        print(k[0,0])
        #还原回去，用loglog作图
        X = np.exp(X)
        y = np.exp(y)

        plt.loglog(X, y)

        plt.loglog(x_mid, y_p, 's-' + color[i], label=str(scale) + 'min')

        plt.legend(loc="upper right", fontsize='small')


        plt.savefig('./img/' + str(scale) + 'min' + '负尾.svg')

        plt.show()
def tail_show():
    for i, scale in enumerate(data_scale):
        # 获取收益率
        r_data = data_min.callogR(scale)
        mid, p = ep.getP(r_data, 101)
        linear = LinearRegression()
        ind = mid > 0
        x_mid = mid[ind]
        y_p = p[ind]
        x_mid = np.abs(x_mid)
        y_p = np.abs(y_p)
        plt.loglog(x_mid, y_p, '.-' + color[i], label=str(scale) + 'min')
        plt.legend(frameon=False,loc="lower left", fontsize='xx-small')

    plt.savefig('./img/' + '-正尾集合.svg')
    plt.show()
```