|  |
| --- |
| 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)∪[*r*1, *r*2)∪[*r*2, *r*3) ∪[*r*3, *r*5) ∪…∪[*r*i-1, *r*i) ∪…∪[*r*n-1, *r*n], 再分别计算每个区间内的样本个数*yi*，进而计算概率*pi*=*yi*/(*ri*-*ri*-1)/*m*, 其中*m*为样本总个数。画图时，区间概率*pi*对应*ri*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）** |
| 在本次实验中，通过对中国股市上证指数分钟数据的深入分析，探讨了不同时间尺度下收益率分布的特征。实验结果揭示了收益率分布的非正态特性和幂律分布特征，尖峰厚尾的现象，以及在不同时间尺度下的显著差异。  首先，实验结果表明，随着时间尺度的增大，收益率分布逐渐扩散，小尺度收益率尖峰厚尾特征更为明显。这一现象表明，在较短的时间尺度下，市场波动可能更加剧烈，极端事件的发生概率较高。其次，通过正态分布拟合和尾部分布特征的检验，进一步确认了中国股市收益率分布的非正态特性和幂律分布特征。KS统计量和P值的计算结果表明，各时间尺度下的收益率序列均显著偏离正态分布。这一发现对于传统的风险评估模型提出了挑战，因为这些模型通常基于正态分布假设。因此，投资者和风险管理者需要采用更为复杂的模型来准确评估市场风险。  在实验过程中，我对Python编程语言及其相关库（如Numpy、Matplotlib和Pandas）有了更深入的理解和应用。通过实际编程，我掌握了如何读取和处理金融数据、计算收益率序列、绘制概率密度图以及使用powerlaw等第三方库进行幂律分布拟合相关的数据统计分析。这些技能不仅在本次实验中发挥了重要作用，也为我未来在金融领域的研究和工作打下了坚实的基础。  在实验心得方面，我认为理论与实践的结合是学习金融计算的关键。通过本次实验，我不仅巩固了理论知识，还提高了实际操作能力。在未来的学习中，我将继续努力提高自己的数据分析能力，以便更好地理解和应对金融市场的挑战。  对于课程的改进建议，我认为目前理论加上机实践的模式非常好，在理论可上可以增加更多关于金融市场实际案例的分析，以及如何将理论知识应用于实际问题的讨论。在实践课上可以提供更多的编程练习机会，解决实际问题的过程中能更好提高编程技能和金融分析能力。 |
| 教师批阅： 实验成绩：  教师签名: 日期： |

**中国股市收益率的分布特征**

摘要：本文旨在通过实证分析，探讨中国股市收益率的分布特征。研究基于2003年1月2日至2015年12月31日的上证指数分钟数据，采用Python编程语言，结合Numpy、Matplotlib和Pandas等库，对不同时间尺度（1分钟至240分钟）的收益率序列进行了深入分析。研究发现，随着时间尺度的增大，收益率分布逐渐扩散，呈现出尖峰厚尾的特征，且小时间尺度下的收益率分布特征更为明显。此外，通过正态分布拟合和尾部分布特征的检验，本文进一步揭示了中国股市收益率分布的非正态特性和尾部分布的幂律特征。研究结果对于理解市场动态、风险管理和投资策略制定具有重要意义。

1 文献综述

在股票市场收益率分布的研究领域，学者们已经提出了多种理论和模型。早期的研究多基于正态分布假设，认为股票收益率是独立同分布的随机变量（Mandelbrot, 1963）。然而，随着金融市场的发展和数据的积累，越来越多的证据表明，股票收益率的分布具有尖峰厚尾（Levy, 1972）和波动率聚集（Engle, 1982）等非正态特性。这些特性意味着极端事件的发生概率远高于正态分布所预测的，对金融市场的风险管理提出了新的挑战。

在中国市场，都国雄和宁宣熙（2007）对中国股市收益概率分布进行了统计特性分析，发现收益率分布呈现出明显的非正态特征。陈收等人（2008）研究了中国股票市场的多标度幂律特征和临界现象，揭示了市场收益率在不同时间尺度下的复杂动态。钟春仿和佟孟华（2009）则从分形特征的角度对我国股市运行进行了实证研究，进一步丰富了对市场行为的理解。

本研究在前人的基础上，通过实证分析上证指数的分钟数据，旨在更准确地描述中国股市收益率的分布特征，并探讨其对投资者行为和市场稳定性的影响。通过对不同时间尺度收益率序列的分析，本文期望为投资者提供更为精确的风险评估工具，同时也为监管机构制定相关政策提供理论支持。

2 模型和方法

本实验中将上证指数在2003年1月2日至2015年12月31日之间的1分钟指数数据序列作为研究样本。所有数据读取自SSEC\_min.mat文件，数据处理程序使用python编写，图像使用matplotlib生成。

2.1不同时间尺度下收益率序列的计算

在本实验中，需要使用到1分钟、5分钟、10分钟、30分钟、60分钟、120分钟、240分钟的收益率序列数据。通过读取SSEC\_min.mat文件能够获取到1分钟的上证指数数据，再通过一定的数据处理即可获得所需的7中时间尺度下的收益率从序列。

本实验中选择使用对数差分收益（式2-1）的计算方式来计算得到1分钟、5分钟、10分钟、30分钟、60分钟、120分钟、240分钟的收益率序列数据。

收益率计算公式： 𝑧(Δ𝑡)=ln𝑃(𝑡)−ln𝑃(𝑡−Δ𝑡) (2-1)

在python中的计算代码：

p\_min = matdata['p'][:,0] # 获取到上证指数的分钟数据

#计算上证指数的1分钟收益率序列

r\_min1 = np.log(p\_min[1:])-np.log(p\_min[:-1])

p\_min5 = p\_min[::5] # 按照5分钟一次的尺度进行采样

#计算上证指数的5分钟收益率序列

r\_min5 = np.log(p\_min5[1:]) - np.log(p\_min5[:-1])

p\_min10 = p\_min[::10] # 按照10分钟一次的尺度进行采样

#计算上证指数的10分钟收益率序列

r\_min10 = np.log(p\_min10[1:]) - np.log(p\_min10[:-1])

p\_min30 = p\_min[::30] # 按照30分钟一次的尺度进行采样

#计算上证指数的30分钟收益率序列

r\_min30 = np.log(p\_min30[1:]) - np.log(p\_min30[:-1])

p\_min60 = p\_min[::60] # 按照60分钟一次的尺度进行采样

#计算上证指数的60分钟收益率序列

r\_min60 = np.log(p\_min60[1:]) - np.log(p\_min60[:-1])

p\_min120 = p\_min[::120] # 按照120分钟一次的尺度进行采样

#计算上证指数的120分钟收益率序列

r\_min120 = np.log(p\_min120[1:]) - np.log(p\_min120[:-1])

p\_min240 = p\_min[::240] # 按照240分钟一次的尺度进行采样

#计算上证指数的240分钟收益率序列

r\_min240 = np.log(p\_min240[1:]) - np.log(p\_min240[:-1])

根据股票市场的相关规定，每个交易日股票的涨跌幅有10%的限制，所以以上通过计算获得的收益率序列中的数据数值应在-0.1和0.1之间。因此继续对各个时间尺度的收益率序列的异常值进行排除。

Python代码如下：

r\_min1 = r\_min1[(r\_min1>=-0.1)&(r\_min1<=0.1)]

r\_min5 = r\_min5[(r\_min5>=-0.1)&(r\_min5<=0.1)]

r\_min10 = r\_min10[(r\_min10>=-0.1)&(r\_min10<=0.1)]

r\_min30 = r\_min30[(r\_min30>=-0.1)&(r\_min30<=0.1)]

r\_min60 = r\_min60[(r\_min60>=-0.1)&(r\_min60<=0.1)]

r\_min120 = r\_min120[(r\_min120>=-0.1)&(r\_min120<=0.1)]

r\_min240 = r\_min240[(r\_min240>=-0.1)&(r\_min240<=0.1)]

至此，1分钟、5分钟、10分钟、30分钟、60分钟、120分钟、240分钟的收益率序列计算完成。

2.2计算不同尺度收益率的经验概率密度

经验概率指的是特定事件发生次数在总样本中的比例。在本实验中计算经验概率密度的思路为将分析样本覆盖的范围分成若干区间，即[r0, r1)∪[r1, r2)∪[r2, r3) ∪[r3, r5) ∪…∪[ri-1, ri) ∪…∪[rn-1, rn], 再分别计算每个区间内的样本个数yi，进而计算概率pi=yi/(ri-ri-1)/m, 其中m为样本总个数。

实验中实现经验概率密度计算的python函数如下：

def myfun\_emp\_pdf(data\_sample, num\_bin=31):

bin = np.linspace(np.min(data\_sample), np.max(data\_sample), num\_bin)

x\_emp = np.zeros(len(bin)-1)

y\_emp = np.zeros(len(bin)-1)

for i in range(len(bin)-1):

x\_emp[i] = (bin[i] + bin[i+1])/2

y\_emp[i] = np.sum( (data\_sample >= bin[i]) & (data\_sample < bin[i+1]) )/len(data\_sample)/(bin[i+1] - bin[i])

return x\_emp, y\_emp

之后使用该函数分别带入1分钟、5分钟、10分钟、30分钟、60分钟、120分钟、240分钟的收益率序列数据，对于不同时间尺度的数据该函数会返回其对应的x\_emp,和y\_emp，即区间和经验概率。

实验中的计算代码如下：

x\_emp\_min1, y\_emp\_min1 = myfun\_emp\_pdf(r\_min1, num\_bin=51)

x\_emp\_min5, y\_emp\_min5 = myfun\_emp\_pdf(r\_min5, num\_bin=51)

x\_emp\_min10, y\_emp\_min10 = myfun\_emp\_pdf(r\_min10, num\_bin=51)

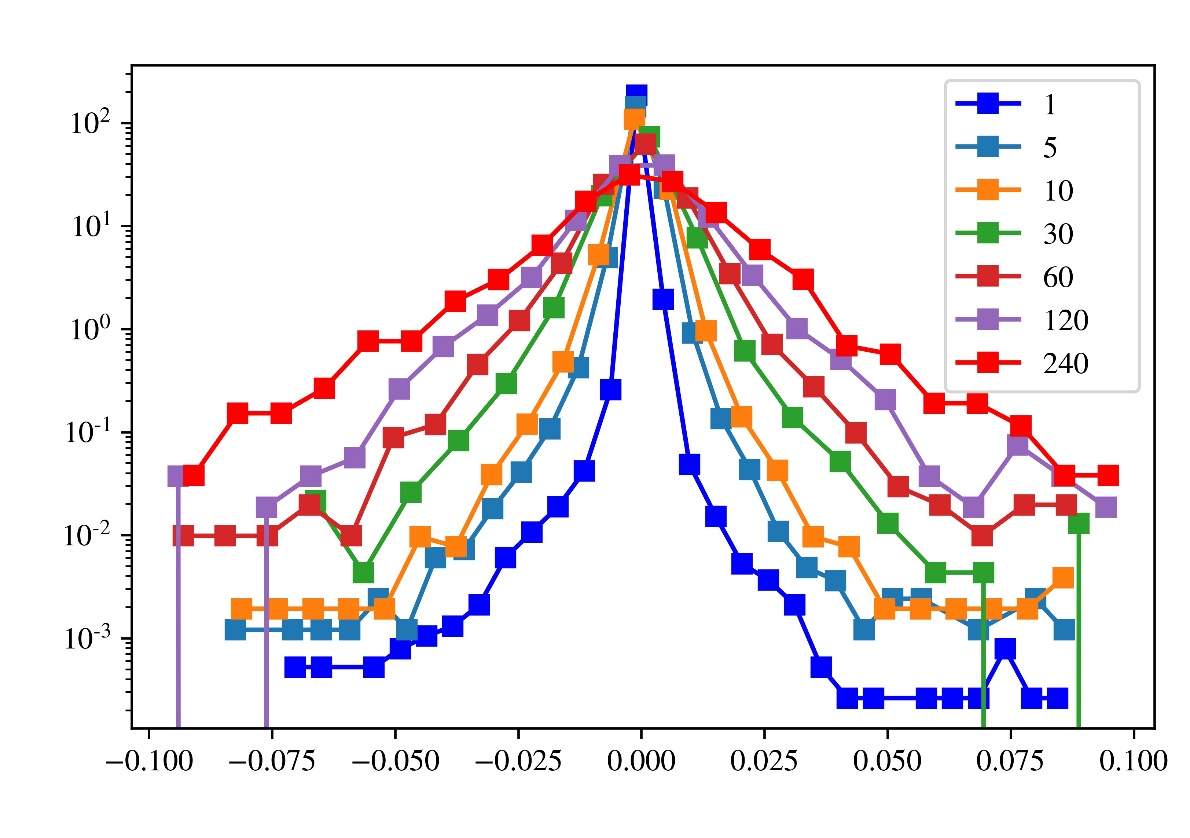
x\_emp\_min30, y\_emp\_min30 = myfun\_emp\_pdf(r\_min30)

x\_emp\_min60, y\_emp\_min60 = myfun\_emp\_pdf(r\_min60)

x\_emp\_min120, y\_emp\_min120 = myfun\_emp\_pdf(r\_min120)

x\_emp\_min240, y\_emp\_min240 = myfun\_emp\_pdf(r\_min240)

将7类不同时间尺度数据的经验概率在同一张图上展示，最终得出的图片如下所示：



**图2-1 不同时间尺度下的收益率经验概率图**

2.3计算不同尺度收益率的正态拟合分布

2.3.1正态分布的拟合

通过拟合不同尺度收益率的正态分布，可以将实际收益率的概率分布与正态分布的偏离程度进行比较。正态分布概率密度如式2-2

(2-2)

在实验中，使用scipy.stats 包下的norm函数进行正态分布拟合，使用norm.fit函数进行拟合，使用norm.pdf进行概率密度的计算。

实验中编写的python代码如下：

# 以1分钟和240分钟数据为例

mu\_daily, sigma\_daily = norm.fit(r\_daily)

mu\_min, sigma\_min = norm.fit(r\_min)

x\_fit\_daily = np.linspace(-0.1, 0.1, 300)

y\_fit\_daily = norm.pdf(x\_fit\_daily, loc=mu\_daily, scale=sigma\_daily)

x\_fit\_min = np.linspace(-0.1, 0.1, 300)

y\_fit\_min = norm.pdf(x\_fit\_min, loc=mu\_min, scale=sigma\_min)

得到相应数据后使用matplotlib进行可视化展示：

plt.figure(1)

plt.semilogy(x\_emp\_daily, y\_emp\_daily, 'o-r', lw=2, ms=7, mfc='k', label=r'$r\_d$ Emp PDF')

plt.semilogy(x\_fit\_daily, y\_fit\_daily, '-m', lw=2, label=r'$r\_d$ Norm Fits')

plt.semilogy(x\_emp\_min, y\_emp\_min, 's-b', lw=2, ms=7, mfc='k', label=r'$r\_m$ Emp PDF')

plt.semilogy(x\_fit\_min, y\_fit\_min, '-k', lw=2, label=r'$r\_m$ Norm Fits')

plt.xlim([-0.1, 0.1])

plt.xticks([-0.1, -0.05, 0, 0.05, 0.1])

plt.ylim([10\*\*-4, 10\*\*3])

plt.yticks(10.\*\*np.arange(-3, 4, 2))

plt.xlabel(r'$r$', fontsize=26)

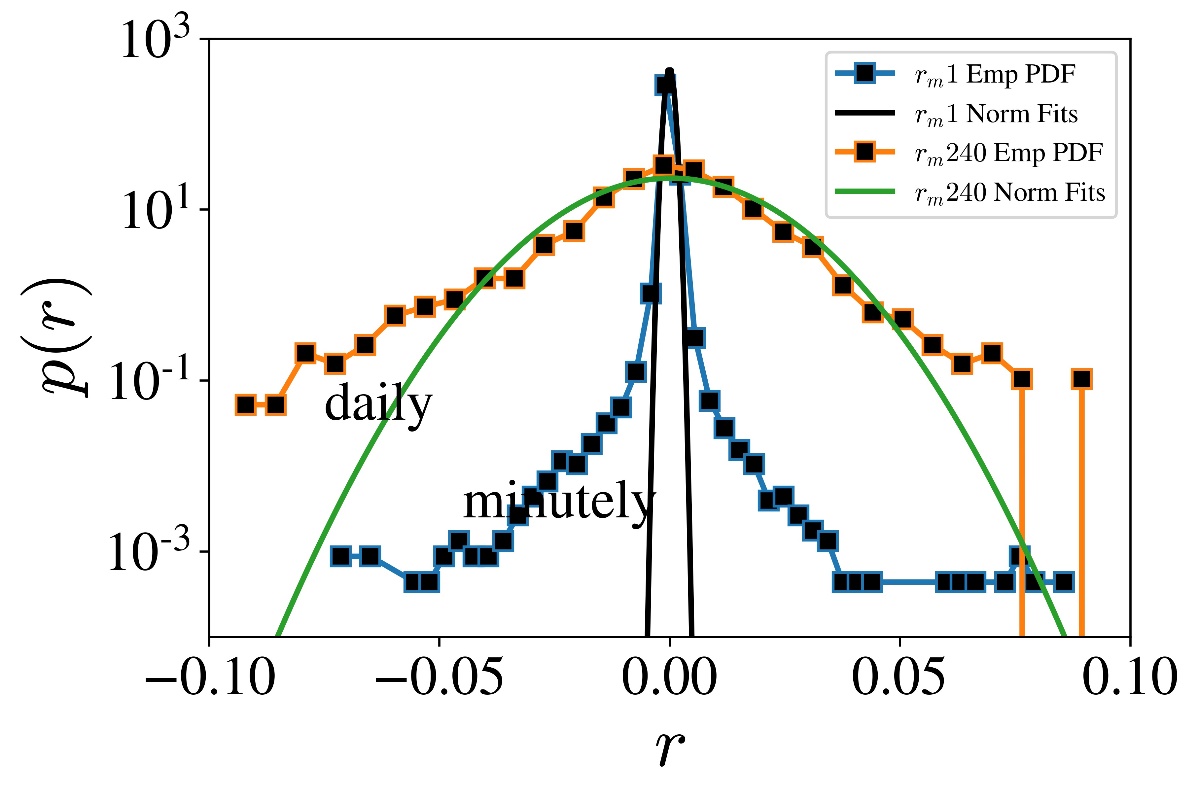
plt.ylabel(r'$p(r)$', fontsize=26)

plt.text(-0.075, 0.035, 'daily', fontsize=20)

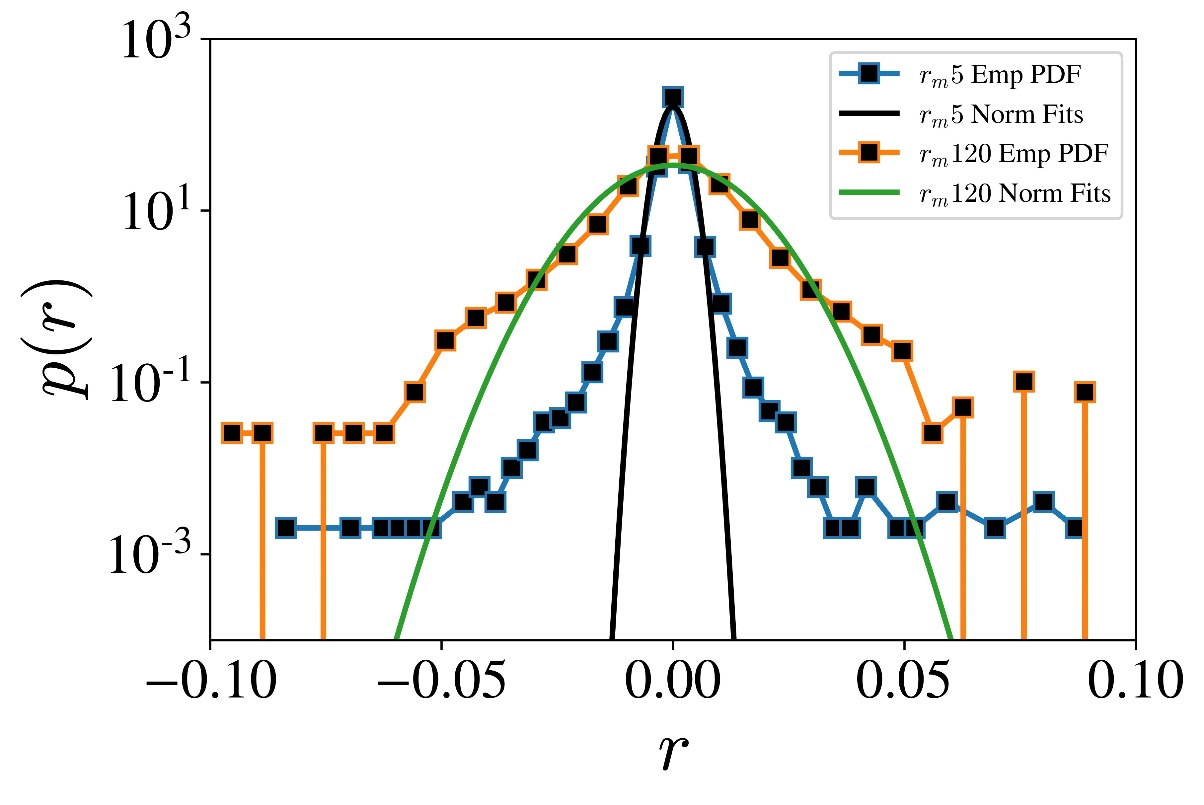
plt.text(-0.045, 0.0025, 'minutely', fontsize=20)

plt.legend(loc='upper right', fontsize=10)

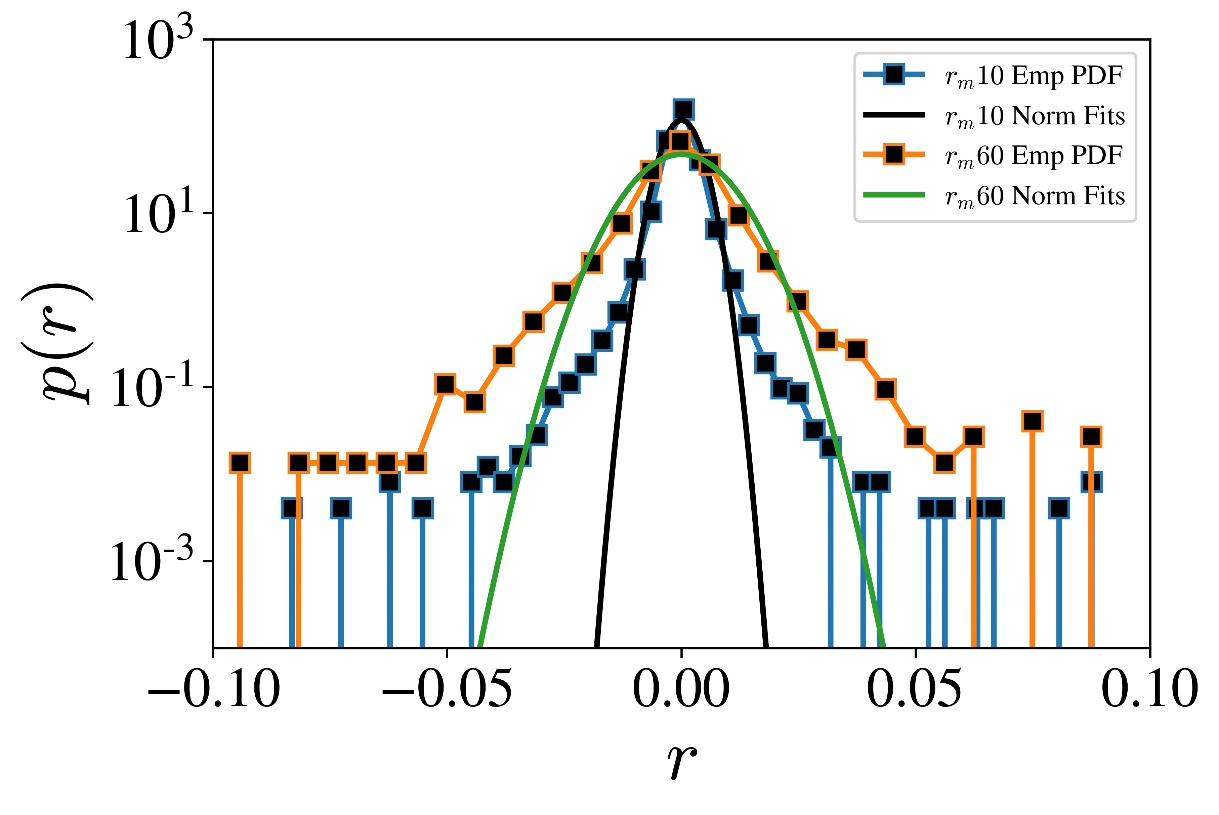
绘制出的图像如下：



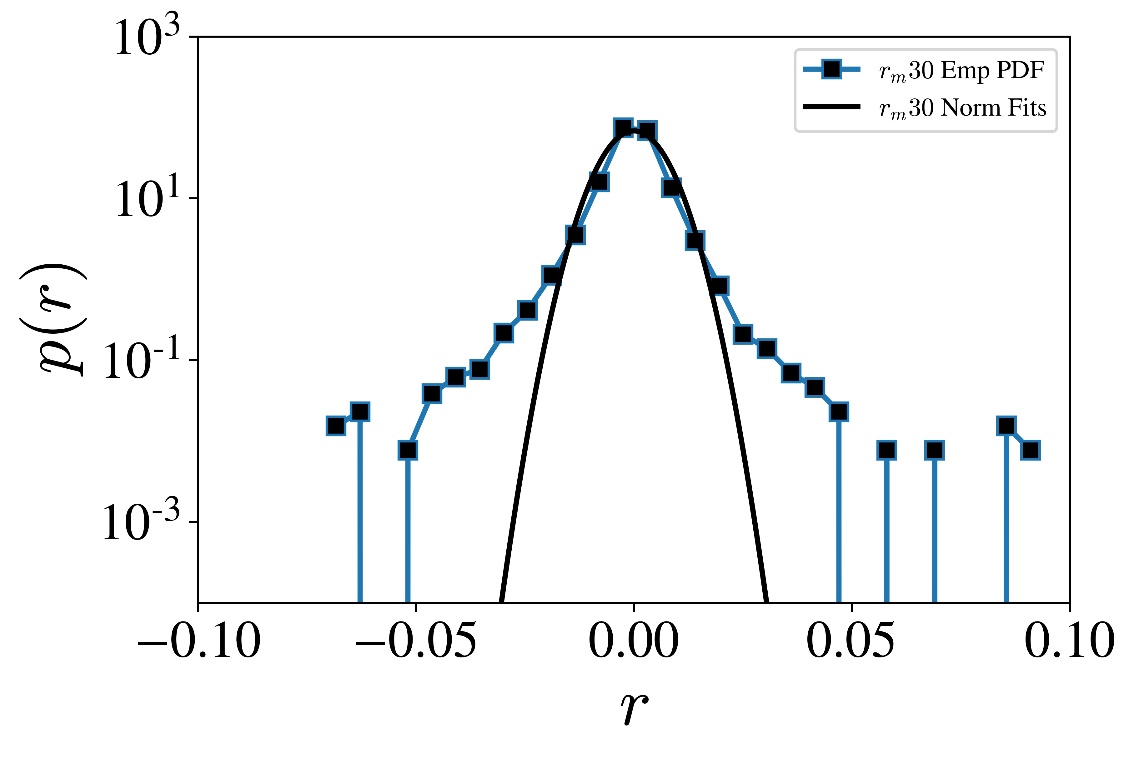
**图2-2 1分钟和240分钟收益率的正态拟合分布**



**图2-3 5分钟和120分钟收益率的正态拟合分布**



**图2-4 10分钟和60分钟收益率的正态拟合分布**

****

**图2-5 30分钟收益率的正态拟合分布**

2.3.2计算不同时间尺度下正态分布偏离程度

为计算不同时间尺度下正态分布偏离程度，本实验中用到了scipy.stats下的kstest函数，该函数会返回两个值KS统计量和P值。

KS统计量表示样本数据与正态分布的偏离程度，KS统计量越大，样本数据与正态分布的偏离程度越大。

P值代表了可以拒绝数据符合正态分布的假设的可能性，P值越接近零（低于显著性水平0.05）表面可以拒绝数据符合正态分布的假设。

本实验中计算偏离程度的python代码为：

from scipy.stats import kstest

def ks (r\_min):

ksStatistic, pvalue = kstest(r\_min,'norm')

return ksStatistic, pvalue

计算得出不同时间尺度收益率的KS统计量和P值见表2-1

**表2.1 各收益率序列正态分布偏离程度**

|  |  |  |
| --- | --- | --- |
| **收益率类别** | **KS统计量** | **P值** |
| **1分钟** | **0.4972** | **0.0** |
| **5分钟** | **0.4935** | **0.0** |
| **10分钟** | **0.4919** | **0.0** |
| **30分钟** | **0.4883** | **0.0** |
| **60分钟** | **0.4836** | **0.0** |
| **120分钟** | **0.4788** | **0.0** |
| **240分钟** | **0.4731** | **0.0** |

2.4检验不同尺度收益率的尾部分布特征是否满足幂律分布

2.4.1考察不同尺度收益率的尾部分布特征

在考察不同尺度收益率的尾部分布特征中我们使用matplotlib中的loglog函数对收益率尾分布进行可视化。

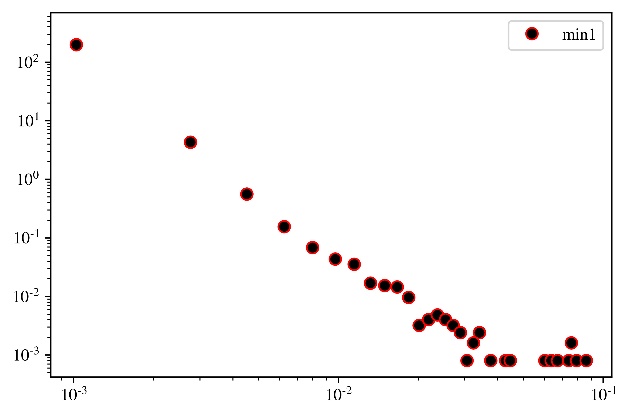
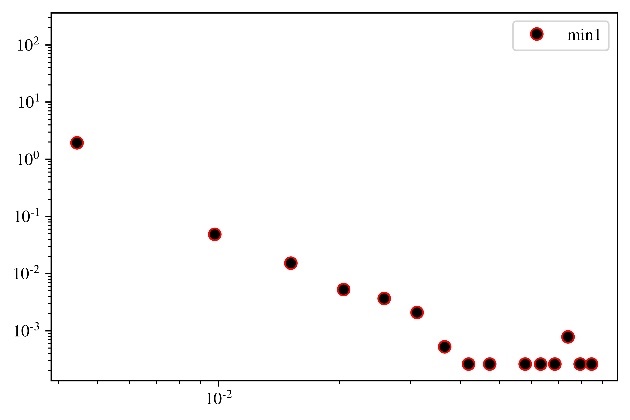
使用到的python程序如下：

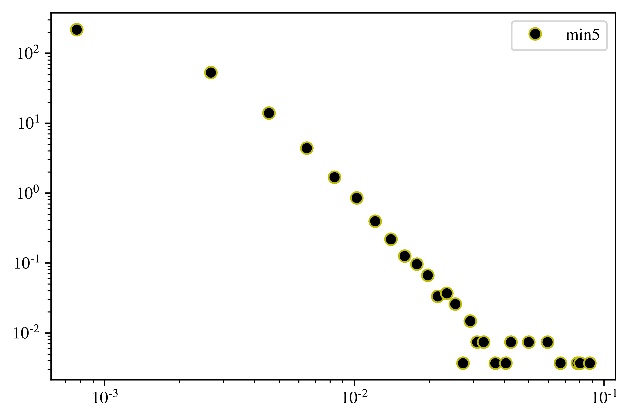
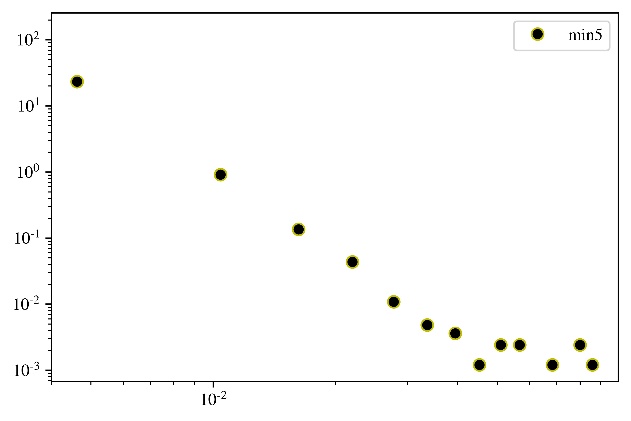
#上证指数1分钟收益率的尾部分布特征

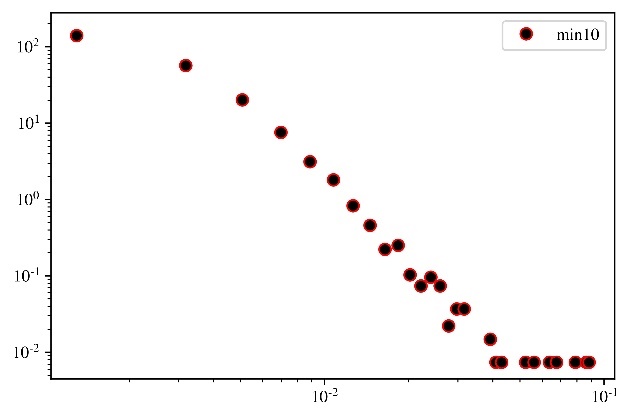
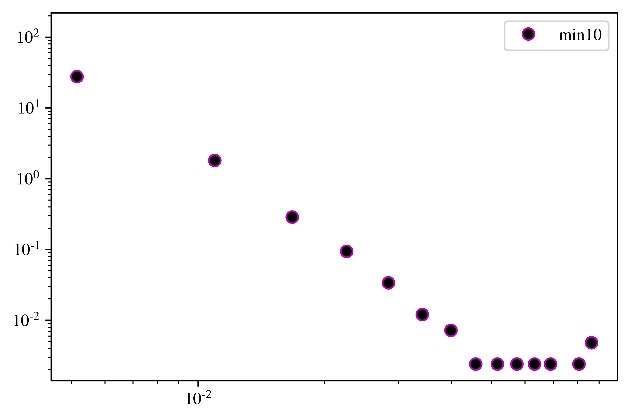
plt.loglog(x\_emp\_min1, y\_emp\_min1, 'or', lw=2, ms=7, mfc='k')

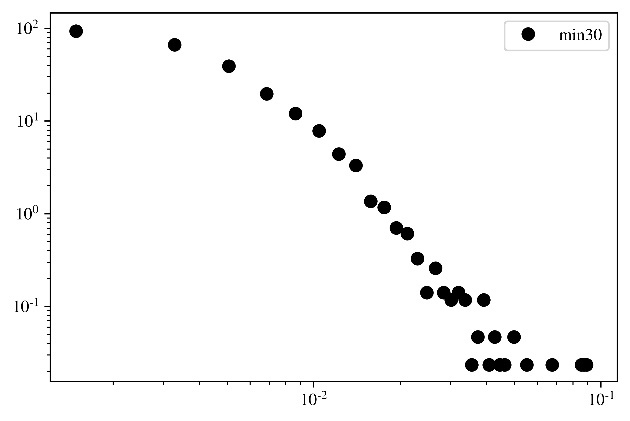
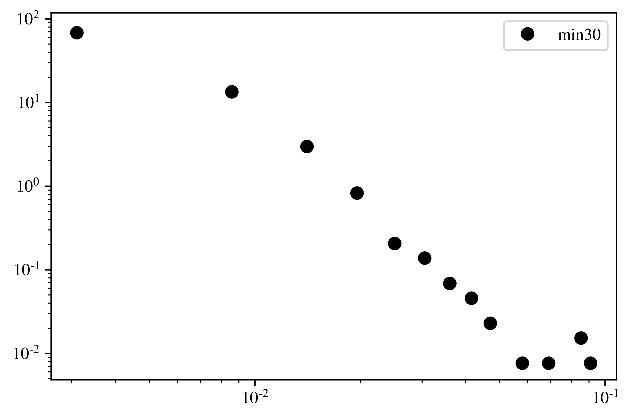
由于原先进行经验概率密度计算时把数据范围分成了num\_bin=31个小区间,获取到的概率密度信息较少，所以实验中又将num\_bin设置为93进行绘图，下方左侧为num\_bin=31时绘制的尾部分布特征，右侧则是num\_bin=93时绘制的尾部分布特征。

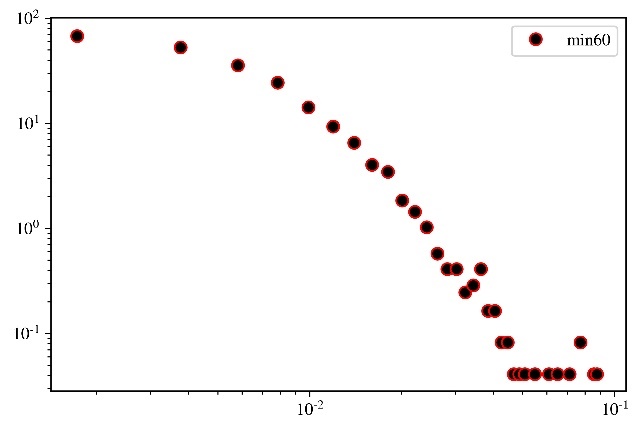
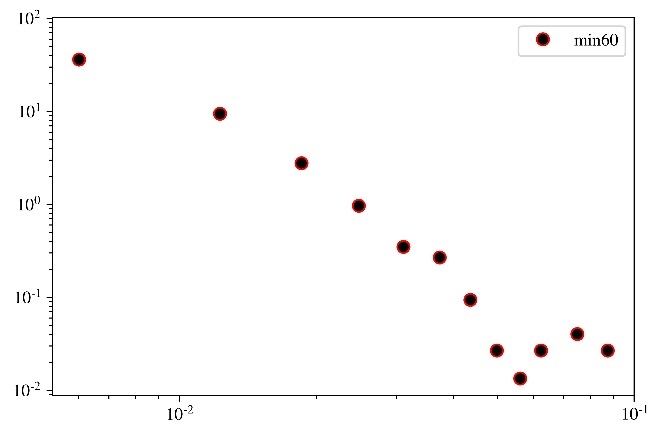
不同尺度下收益率的尾部分布特征图如下：

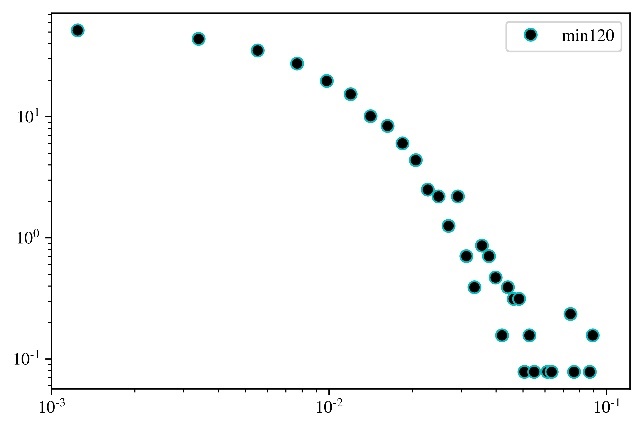
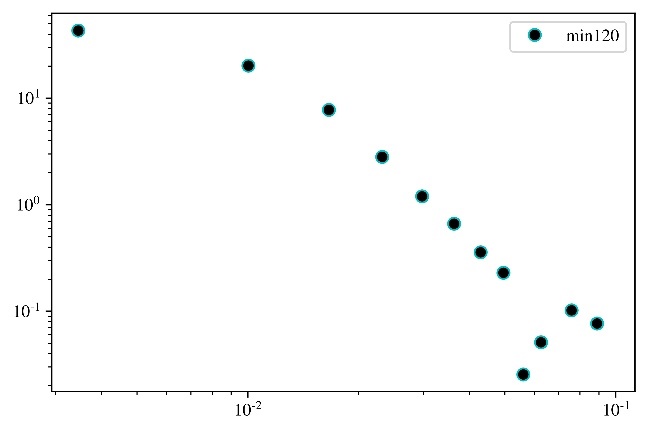


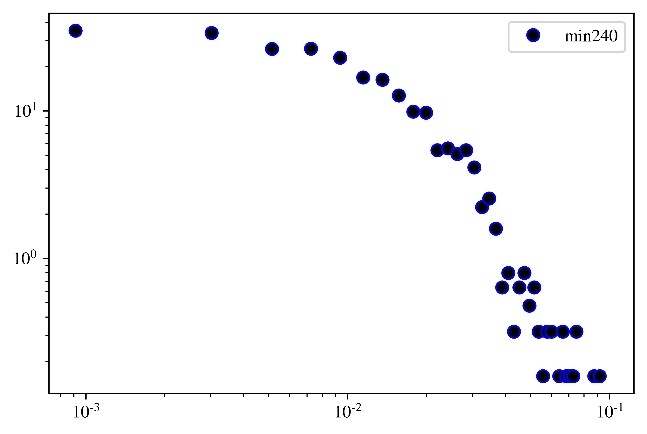
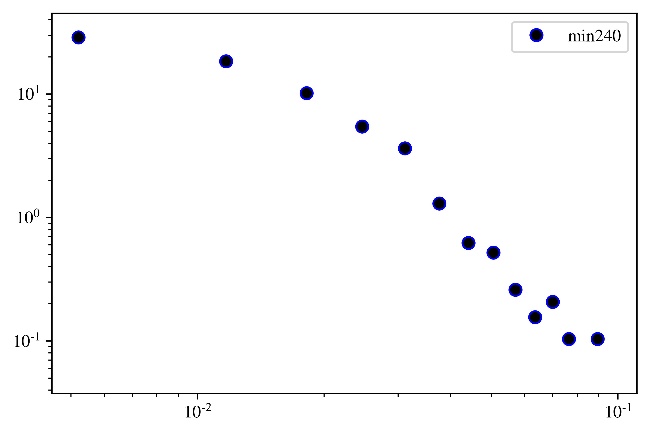












2.4.2进一步使用统计量研究收益率的尾部分布特征

为更加精确地研究不同尺度收益率的尾部分布特征是否满足幂律分布（式2-2），在本实验中使用到了第三方库powerlaw，用于进行幂律分布的拟合，并得到拟合结果的统计信息。

(2-2)

以下是对计算得到的统计信息的说明：

1. alpha：在函数中通过**fit.alpha**计算得到，是通过拟合得到的幂律指数。
2. R值（对数似然比检验）：对数似然比统计量，表示幂律分布和对数正态分布之间的对数似然比。当该值为负值时，幂律分布的对数似然值更大，即更好地拟合了数据，意味着数据更符合幂律分布。
3. P值：用于判断两个模型的拟合质量是否显著不同, 如果P值较小（通常小于显著性水平，如0.05），则可以拒绝两个分布是相同的假设，表明幂律分布可能更好地描述数据。

本实验中计算以上统计信息的python代码如下：

def powerlawTest(data):

fit = powerlaw.Fit(data[::100])

print('alpha:', fit.alpha)

R, p = fit.distribution\_compare('power\_law', 'lognormal')

计算得出不同时间尺度收益率统计信息值见表2-2

**表2.2 不同时间尺度收益率幂律拟合统计信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **收益率类别** | **alpha** | **R值** | **P值** |
| **1分钟** | **3.2686** | **-3.5158** | **0.1542** |
| **5分钟** | **3.7115** | **0.0009** | **0.9364** |
| **10分钟** | **4.8575** | **-0.7777** | **0.0016** |
| **30分钟** | **2.6507** | **-0.0781** | **0.1178** |
| **60分钟** | **2.7375** | **-1.3556** | **0.2860** |
| **120分钟** | **2.2476** | **-0.3010** | **0.6582** |
| **240分钟** | **3.7611** | **-0.5826** | **0.0002** |

3 结果与讨论

3.1根据不同尺度收益率的经验概率密度比较收益率分布的区别

通过对不同尺度收益率进行经验概率密度计算并绘制图像（图2-1），可以发现在各个尺度下收益率分布几乎是对称的，随着时间跨度的增大，分布逐渐扩散。相比于正态分布各个尺度下收益率都呈现尖峰厚尾的特征，且时间尺度越小，尖峰厚尾的特征越明显。

根据这一结论和股票市场实际进行讨论，收益率序列尖峰厚尾的特征表明在较小的时间尺度下，市场可能会出现极端的收益率波动，投资者的风险也许会因此增加。进一步了解这种特征有助于投资者制定更有效的风险管理策略，以适应短期市场波动。尖峰厚尾的分布特征反映着市场中存在的非理性行为和极端事件的概率。从这一方向入手可以更深入地探讨市场参与者的行为动因，有助于理解市场中的非理性决策和交易行为。

3.2分析不同尺度收益率对正态分布偏离程度

根据实验中使用matplotlib绘制出的图像（图2-2，2-3，2-4，2-5）和使用kstest函数计算得到的KS统计量和P值进行分析（数据见表2-1），可以明显的发现收益率分布与时间尺度有关，并且1分钟、5分钟、10分钟、30分钟、60分钟、120分钟、240分钟所有时间跨度的收益率都偏离了正态分布。

从P统计量来看，P值若是小于显著性水平（如0.05），则拒绝其符合正态分布的假设。从P值计算结果来看，所有时间尺度下的P值均为0，说明所有时间尺度下的收益率都偏离了正态分布。从KS统计量来看，KS统计量描述了样本数据与正态分布的偏离程度，KS统计量越大，样本数据与正态分布的偏离程度越大。在实验结果中随着时间尺度的增大， KS也随之减小，说明时间尺度越小，收益率偏离正态越厉害。

3.3不同尺度收益率的尾部分布特征

考察收益率的尾部分布特征，需要分析其图像以及计算所得的统计量进行综合考虑。某尺度收益率的尾部分布特征是否满足幂律分布，需要考虑其图像、R值、P值（数据见表2-2）。对于是否服从“负三次方定律”还需进一步考虑计算得出的alpha值。

对于1分钟收益率序列，从使用loglog函数画出的图像来看，1分钟收益率序列符合幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-3.5158（为负），P值为0.1542，综合分析可以得出1分钟收益率尾部分布特征满足幂律分布。并且其alpha值（通过拟合得到的幂律指数）为3.26，可以认为其服从“负三次方定律”。

对于5分钟收益率序列，从使用loglog函数画出的图像来看，5分钟收益率序列不太符合幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-0.0009（为负），P值为0.9364，综合分析后判断5分钟收益率尾部分布特征不满足幂律分布。

对于10分钟收益率序列，从使用loglog函数画出的图像来看，无法判断10分钟收益率序列是否存在幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-0.7777（为负），P值为0.0016，综合分析后判断10分钟收益率尾部分布特征满足幂律分布。其alpha值（通过拟合得到的幂律指数）为4.8575，不服从“负三次方定律”。

对于30分钟收益率序列，从使用loglog函数画出的图像来看，30分钟收益率序列不太符合幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-0.0781（为负），P值为0.1178。统计数据显示其有幂律分布的特征，但在图中其尾部分布特征已明显偏离幂律分布特征，考虑到R值、P值是作为参考值提供的并不一定准确，综合分析判断30分钟收益率尾部分布特征不满足幂律分布。

对于60分钟收益率序列，从使用loglog函数画出的图像来看，60分钟收益率序列不符合幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-1.3556（为负），P值为0.2860，P值明显偏大。综合图像和数据判断60分钟收益率尾部分布特征不满足幂律分布。

对于120分钟收益率序列，从使用loglog函数画出的图像来看，120分钟收益率序列不符合幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-0.3010（为负），P值为0.6582，P值明显偏大。综合图像和数据判断120分钟收益率尾部分布特征不满足幂律分布。

对于240分钟收益率序列，从使用loglog函数画出的图像来看，240分钟收益率序列不符合幂律分布的特征。根据powerlawTest函数计算得到的统计量，R值为-0.5826，P值为0.0002，从数据来看240分钟收益率存在幂律分布特征，但图像中直观展示了其尾部分布，其尾部分布特征并不满足幂律分布。考虑到计算中数据处理、选择范围等因素可能存在误差，并且R值、P值是作为参考值提供的所以并不一定准确，所以最终认为240分钟收益率尾部分布特征不满足幂律分布。

根据以上结论和股票市场实际进行讨论，对于满足幂律分布特征的收益率序列，可能存在更极端的事件风险，需要谨慎处理。对收益率尾部分布特征的研究对于度量、预测和控制资本市场的极端波动，提供科学的风险管理决策无疑具有很大的帮助，但金融市场是一个复杂的系统，其行为受到多种因素的影响，包括市场参与者行为、市场机制、宏观经济状况等。因此，仅仅通过尾部分布特征来判断市场的行为是复杂且有限的。

4 参考文献

1. 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
2. 22-23．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，2008，16(3)
3. 钟春仿，佟孟华. 我国股市运行的分形特征及与幂律关系的实证研究[J]. 生产力研究，2009
4. 王新宇，宋学锋. 拟合中国股票市场收益的统计分布[J]. 系统工程理论与实践，2006

[5] S.Ghashghaie, W.Breymann, J.Peinke, P.Talkner, Y.Dodge. Turbulent cascades in foreign exchange markets[J]. Nature, 1996

5 附录

实验环境：python3.8,Jupyter Notebook

代码：

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from scipy.io import loadmat

from scipy.stats import norm

import powerlaw

%matplotlib inline

plt.rcParams.update({"font.family": "STIXGeneral",

"font.size": 20,

"mathtext.fontset": "cm"})

# matdata中数据字段

# 日期 上涨指数 时间

matdata = loadmat('./SSEC\_min.mat')

p\_min = matdata['p'][:,0] # 获取到上证指数的分钟数据

r\_min1 = np.log(p\_min[1:])-np.log(p\_min[:-1]) #计算上证指数的1分钟收益率序列

p\_min5 = p\_min[::5] # 按照5分钟一次的尺度进行采样

r\_min5 = np.log(p\_min5[1:]) - np.log(p\_min5[:-1]) #计算上证指数的5分钟收益率序列

p\_min10 = p\_min[::10] # 按照10分钟一次的尺度进行采样

r\_min10 = np.log(p\_min10[1:]) - np.log(p\_min10[:-1]) #计算上证指数的10分钟收益率序列

p\_min30 = p\_min[::30] # 按照30分钟一次的尺度进行采样

r\_min30 = np.log(p\_min30[1:]) - np.log(p\_min30[:-1]) #计算上证指数的30分钟收益率序列

p\_min60 = p\_min[::60] # 按照60分钟一次的尺度进行采样

r\_min60 = np.log(p\_min60[1:]) - np.log(p\_min60[:-1]) #计算上证指数的60分钟收益率序列

p\_min120 = p\_min[::120] # 按照120分钟一次的尺度进行采样

r\_min120 = np.log(p\_min120[1:]) - np.log(p\_min120[:-1]) #计算上证指数的120分钟收益率序列

p\_min240 = p\_min[::240] # 按照240分钟一次的尺度进行采样

r\_min240 = np.log(p\_min240[1:]) - np.log(p\_min240[:-1]) #计算上证指数的240分钟收益率序列

# 对各个时间尺度的收益率序列的异常值进行排除

r\_min1 = r\_min1[(r\_min1>=-0.1)&(r\_min1<=0.1)]

r\_min5 = r\_min5[(r\_min5>=-0.1)&(r\_min5<=0.1)]

r\_min10 = r\_min10[(r\_min10>=-0.1)&(r\_min10<=0.1)]

r\_min30 = r\_min30[(r\_min30>=-0.1)&(r\_min30<=0.1)]

r\_min60 = r\_min60[(r\_min60>=-0.1)&(r\_min60<=0.1)]

r\_min120 = r\_min120[(r\_min120>=-0.1)&(r\_min120<=0.1)]

r\_min240 = r\_min240[(r\_min240>=-0.1)&(r\_min240<=0.1)]

# 1分钟、5分钟、10分钟、30分钟、60分钟、120分钟、240分钟的收益率序列计算完成

# 用于计算经验概率密度的函数

def myfun\_emp\_pdf(data\_sample, num\_bin=31):

bin = np.linspace(np.min(data\_sample), np.max(data\_sample), num\_bin)

x\_emp = np.zeros(len(bin)-1)

y\_emp = np.zeros(len(bin)-1)

for i in range(len(bin)-1):

x\_emp[i] = (bin[i] + bin[i+1])/2

y\_emp[i] = np.sum( (data\_sample >= bin[i]) & (data\_sample < bin[i+1]) )/len(data\_sample)/(bin[i+1] - bin[i])

return x\_emp, y\_emp

x\_emp\_min1, y\_emp\_min1 = myfun\_emp\_pdf(r\_min1, num\_bin=51)

x\_emp\_min5, y\_emp\_min5 = myfun\_emp\_pdf(r\_min5, num\_bin=51)

x\_emp\_min10, y\_emp\_min10 = myfun\_emp\_pdf(r\_min10, num\_bin=51)

x\_emp\_min30, y\_emp\_min30 = myfun\_emp\_pdf(r\_min30)

x\_emp\_min60, y\_emp\_min60 = myfun\_emp\_pdf(r\_min60)

x\_emp\_min120, y\_emp\_min120 = myfun\_emp\_pdf(r\_min120)

x\_emp\_min240, y\_emp\_min240 = myfun\_emp\_pdf(r\_min240)

#为避免时间间隔太短收益率为0的情况

x\_emp\_min1 = x\_emp\_min1[y\_emp\_min1>0]

y\_emp\_min1 = y\_emp\_min1[y\_emp\_min1>0]

x\_emp\_min5 = x\_emp\_min5[y\_emp\_min5>0]

y\_emp\_min5 = y\_emp\_min5[y\_emp\_min5>0]

# 将经验概率密度画在一张图上

plt.semilogy(x\_emp\_min1, y\_emp\_min1,'s-b',label='1分钟收益率')

plt.semilogy(x\_emp\_min5, y\_emp\_min5, 's-',label='5分钟收益率')

plt.semilogy(x\_emp\_min10, y\_emp\_min10,'s-',label='10分钟收益率')

plt.semilogy(x\_emp\_min30, y\_emp\_min30,'s-',label='30分钟收益率')

plt.semilogy(x\_emp\_min60, y\_emp\_min60,'s-',label='60分钟收益率')

plt.semilogy(x\_emp\_min120, y\_emp\_min120,'s-',label='120分钟收益率')

plt.semilogy(x\_emp\_min240, y\_emp\_min240,'s-r',label='240分钟收益率')

plt.title('不同时间尺度的收益率经验概率密度')

plt.xlabel('收益率')

plt.ylabel('经验概率密度')

plt.legend()

# 拟合正态分布

mu\_min1, sigma\_min1 = norm.fit(r\_min1) #拟合1分钟收益率序列，获得拟合的均值mu和标准差sigma

mu\_min5, sigma\_min5 = norm.fit(r\_min5)

mu\_min10, sigma\_min10 = norm.fit(r\_min10)

mu\_min30, sigma\_min30 = norm.fit(r\_min30)

mu\_min60, sigma\_min60 = norm.fit(r\_min60)

mu\_min120, sigma\_min120 = norm.fit(r\_min120)

mu\_min240, sigma\_min240 = norm.fit(r\_min240)

x\_fit\_general = np.linspace(-0.1, 0.1, 300) #用于绘制不同尺度收益率拟合正态分布的 x 值范围

# 计算拟合正态分布的 y 值

y\_fit\_min1 = norm.pdf(x\_fit\_general, loc=mu\_min1, scale=sigma\_min1) #1分钟收益率的y值

y\_fit\_min5 = norm.pdf(x\_fit\_general, loc=mu\_min5, scale=sigma\_min5)

y\_fit\_min10 = norm.pdf(x\_fit\_general, loc=mu\_min10, scale=sigma\_min10)

y\_fit\_min30 = norm.pdf(x\_fit\_general, loc=mu\_min30, scale=sigma\_min30)

y\_fit\_min60 = norm.pdf(x\_fit\_general, loc=mu\_min60, scale=sigma\_min60)

y\_fit\_min120 = norm.pdf(x\_fit\_general, loc=mu\_min120, scale=sigma\_min120)

y\_fit\_min240 = norm.pdf(x\_fit\_general, loc=mu\_min240, scale=sigma\_min240)

# 绘制图表

plt.semilogy(x\_emp\_min1, y\_emp\_min1, 's-b') # 1分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min1, '-k') #1分钟收益率正态分布拟合

plt.semilogy(x\_emp\_min5, y\_emp\_min5, 's-') # 5分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min5, '-') #5分钟收益率正态分布拟合

plt.semilogy(x\_emp\_min10, y\_emp\_min10, 's-') # 10分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min10, '-') #10分钟收益率正态分布拟合

plt.semilogy(x\_emp\_min30, y\_emp\_min30, 's-') # 30分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min30, '-') #30分钟收益率正态分布拟合

plt.semilogy(x\_emp\_min60, y\_emp\_min60, 's-') # 60分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min60, '-') #60分钟收益率正态分布拟合

plt.semilogy(x\_emp\_min120, y\_emp\_min120, 's-') # 120分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min120, '-') #120分钟收益率正态分布拟合

plt.semilogy(x\_emp\_min240, y\_emp\_min240, 's-') # 240分钟收益率的经验概率密度图

plt.semilogy(x\_fit\_general, y\_fit\_min240, '-') #240分钟收益率正态分布拟合

plt.xlim([-0.1, 0.1])

plt.xticks([-0.1, -0.05, 0, 0.05, 0.1])

plt.ylim([10\*\*-4, 10\*\*3])

plt.yticks(10.\*\*np.arange(-3, 4, 2))

plt.xlabel(r'$r$', fontsize=26)

plt.ylabel(r'$p(r)$', fontsize=26)

plt.text(-0.075, 0.035, 'daily', fontsize=20)

plt.text(-0.045, 0.0025, 'minutely', fontsize=20)

plt.legend(loc='upper right', fontsize=10)

#计算偏离程度

def ks (r\_min):

ksStatistic, pvalue = kstest(r\_min,'norm')

return ksStatistic, pvalue

# 画出收益率尾部分布图

plt.loglog(x\_emp\_min1, y\_emp\_min1, 'o-', lw=2, ms=7, mfc='k')

plt.loglog(x\_emp\_min5, y\_emp\_min5, 'o-', lw=2, ms=7, mfc='k')

plt.loglog(x\_emp\_min10, y\_emp\_min10, 'o-', lw=2, ms=7, mfc='k')

plt.loglog(x\_emp\_min30, y\_emp\_min30, 'o-', lw=2, ms=7, mfc='k')

plt.loglog(x\_emp\_min60, y\_emp\_min60, 'o-', lw=2, ms=7, mfc='k')

plt.loglog(x\_emp\_min120, y\_emp\_min120, 'o-', lw=2, ms=7, mfc='k')

plt.loglog(x\_emp\_min240, y\_emp\_min240, 'o-', lw=2, ms=7, mfc='k')

# 幂律分布拟合

def powerlawTest(data):

fit = powerlaw.Fit(data[::100])

print('alpha:', fit.alpha)

print('xmin:', fit.xmin)

R, p = fit.distribution\_compare('power\_law', 'lognormal')

print(R)

print(R)

ks (r\_min1)

ks (r\_min5)

ks (r\_min10)

ks (r\_min30)

ks (r\_min60)

ks (r\_min120)

ks (r\_min240)

powerlawTest(r\_min1)

powerlawTest(r\_min5)

powerlawTest(r\_min10)

powerlawTest(r\_min30)

powerlawTest(r\_min60)

powerlawTest(r\_min120)

powerlawTest(r\_min240)