**深 圳 大 学 实 验 报 告**

**课程名称：­ 概率论与数理统计**

**实验项目名称： The central limit theorem**

**学院： 电子与信息工程学院**

**专业： 电子信息工程**

**指导教师： 陈昌盛老师**

**报告人： 贾苏健 学号： 2022280485**

**班级： 06**

**实验时间： 2023.12.7**

**实验报告提交时间： 2023.**

**教务处制**

|  |
| --- |
| Aim of Experiment:   * Familiar with the central limit theorem. * Understand the implementation of the central limit theorem in python. * Know how to visualize data in different distributions.   · 熟悉中心极限定理。  · 了解在Python中实施中心极限定理的方法。  · 知道如何在不同分布中可视化数据。 |
| Experiment Content:   一、通过Python模拟演示中心极限定理（CLT） 在这一部分，通过Python模拟演示中心极限定理（CLT）。主要重点是对于任何分布的总体，采样分布（样本均值的分布）在足够大的样本量时趋于正态分布。进行了两个实验来验证这一点： 实验1 - 指数分布总体 1、总体分布：  (1)总体遵循指数分布，通常用于模拟事件发生前需要等待的预期时间。  (2)指数分布由概率密度函数定义：  2、模拟步骤：  1. 参数计算：  - 指数分布总体的速率参数 设置为4。  - 根据速率参数计算总体均值 和标准差 。  2. 小样本模拟 ：  - 从总体中抽取50个大小为2的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果表明，即使对于较小的样本量（如2），样本均值的分布看起来与指数总体的分布非常不同，并更像是对正态分布的糟糕逼近，呈现一些正偏斜。  3. 大样本模拟：  - 从总体中抽取50个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果表明，随着样本量的增加，样本均值的分布更加接近正态分布。  3、验证：  - 比较大样本模拟（）的50个样本均值的均值和标准差与CLT预测的理论值。 实验2 - 二项分布总体 1、总体分布：  - 总体遵循参数为 和 的二项分布。  2、模拟步骤：  1. 大样本模拟：  - 从二项分布总体中抽取50个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果显示，与指数分布类似，随着样本量的增加，样本均值的分布趋向于正态分布。  3、验证：  - 将样本均值的均值与总体均值进行比较，将样本均值的标准差与CLT预测的值进行比较。 二、在投资/交易中应用中心极限定理 在这一部分中，将CLT应用于投资/交易场景：  1、场景：  - 一位投资者（Jyoti）正在考虑在未来五年内投资一只股票（ITC）。  - 投资者是风险厌恶型的，希望选择平均每月回报最好为非负数的股票。  - Jyoti希望使用一个统计框架来确定ITC股票的平均月回报并具有一定的置信度。  2、步骤：  1. 理论框架：  - 讨论金融模型中的正态性假设，例如Black-Scholes期权定价模型。  - 引入CLT作为基于回报分布假设的统计框架。  2. 数据分析：  - 获取过去10年的ITC股票每日收盘数据并进行分析。  - 可视化每日对数回报的分布，以评估正态性假设。  3、结论：  - 每日对数回报的可视化显示与正态性的偏离，突显了在金融数据中假设正态性的挑战。 三、更多可视化练习 在这一部分，使用底特律市的小时天气数据进行了额外的可视化练习：  1、温度分布：  - 绘制了温度数据的直方图，使用100个区间以可视化分布。  - 将该分布与正态分布曲线进行了比较。  2、多图可视化：  - 可视化了底特律市湿度、气压和温度的变化，分别在不同的图中。  - 创建了温度的7天移动平均线图。  - 进行了其他练习，包括在同一图中可视化不同城市的温度、湿度和气压以及创建城市可见性的交互复选框。  3、天气数据探索：  - 通过可视化探索了温度、湿度和气压的变化。  这些练习演示了如何使用连续分布来近似天气数据的观察分布，并探索了温度、湿度和气压变化的各个方面。 |
| Experiment Process： 3.1 通过Python模拟演示中心极限定理（CLT） 在这一部分，通过Python模拟演示中心极限定理（CLT）。主要重点是对于任何分布的总体，采样分布（样本均值的分布）在足够大的样本量时趋于正态分布。进行了两个实验来验证这一点： 3.1.1 实验1 - 指数分布总体 (1)总体分布：  - 总体遵循指数分布，通常用于模拟事件发生前需要等待的预期时间。  - 指数分布由概率密度函数定义：    *exponentially-distributed-population*  (2)模拟步骤：  1. 参数计算：  - 指数分布总体的速率参数 设置为4。  - 根据速率参数计算总体均值 和标准差 。  *# rate parameter for the exponentially distributed population*  theta = 4.0  *# Population mean (mu), representing mean by parameter theta*  mu = theta  *# Population standard deviation (sd), representing sd by parameter theta*  sd = theta  2. 小样本模拟 ：  - 从总体中抽取50个大小为2的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果表明，即使对于较小的样本量（如2），样本均值的分布看起来与指数总体的分布非常不同，并更像是对正态分布的糟糕逼近，呈现一些正偏斜。  *# drawing 50 random samples of size 2 from the exponentially distributed population*  sample\_size = 2  df2 = **pd**.**DataFrame**(index= ['x1', 'x2'])  for i in **range**(1, 51):      exponential\_sample = **np**.**random**.exponential(theta, sample\_size)      col = f'sample {i}'      df2[col] = exponential\_sample  *# Taking a peek at the samples*  df2  *# Calculating sample means and plotting their distribution*  df2\_sample\_means = df2.**mean**()  **sns**.**distplot**(df2\_sample\_means)  3. 大样本模拟：  - 从总体中抽取50个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果表明，随着样本量的增加，样本均值的分布更加接近正态分布。  sample\_size = 500  sample\_num = 50  df500 = **pd**.**DataFrame**(index=**range**(1, sample\_size + 1))  for i in **range**(1, sample\_num + 1):      exponential\_sample\_large = **np**.**random**.exponential(theta, sample\_size)      col = f'sample {i}'      df500[col] = exponential\_sample\_large  df500\_sample\_means = df500.**mean**()  **sns**.**distplot**(df500\_sample\_means)  (3)验证：  - 比较大样本模拟（）的50个样本均值的均值和标准差与CLT预测的理论值。  *# An estimate of the standard deviation of the sampling distribution can be obtained as:*  sample\_data\_std\_deviation  = df500\_sample\_means.**std**()  **print**("Estimate of standard deviation of the sampling distribution:", sample\_data\_std\_deviation )  *# If you calculate correctly, the above value is very close to the value stated by the CLT, which is:*  sd/**np**.sqrt(sample\_size) 3.1.2 实验2 - 二项分布总体 (1)总体分布：  - 总体遵循参数为 和 的二项分布。    probability-function-clt  (2)模拟步骤：  1. 大样本模拟：  - 从二项分布总体中抽取50个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果显示，与指数分布类似，随着样本量的增加，样本均值的分布趋向于正态分布。  *# drawing 50 random samples of size 500 from a Binomial distribution with parameters k= 30 and p=0.9*  *# Tips: You can use the "np.random.binomial" function to implement a binomial distribution.*  k = 30  p = 0.9  sample\_size = 500  sample\_num = 50  df500\_binomial = **pd**.**DataFrame**(index=**range**(1, sample\_size + 1))  for i in **range**(1, sample\_num + 1):      binomial\_sample\_large = **np**.**random**.binomial(k, p, sample\_size)      col = f'sample {i}'      df500\_binomial[col] = binomial\_sample\_large  df500\_binomial\_sample\_means = df500\_binomial.**mean**()  **sns**.**distplot**(df500\_binomial\_sample\_means)  (3)验证：  - 将样本均值的均值与总体均值进行比较，将样本均值的标准差与CLT预测的值进行比较。  *# Please compute the mean of sample means. If you calculate correctly, mean of sample means is close to the population mean.*  mean\_of\_sample\_means = df500\_binomial\_sample\_means.**mean**()  mean\_of\_sample\_means\_n = 0  for i in **range**(0, sample\_num ):      mean\_of\_sample\_means\_n += df500\_binomial\_sample\_means[i]  mean\_of\_sample\_means\_n /= sample\_num  **print**("Mean of sample means:", mean\_of\_sample\_means)  **print**("Mean of sample means:", mean\_of\_sample\_means\_n)  *# Please compute the standard deviation of sample means. If you calculate correctly, standard deviation of sample means is close to population standard deviation divided by square root of sample size.*  std\_dev\_of\_sample\_means = df500\_binomial\_sample\_means.**std**()  std\_dev\_of\_sample\_means\_n = 0  for i in **range**(0, sample\_num ):      std\_dev\_of\_sample\_means\_n += (df500\_binomial\_sample\_means[i] - mean\_of\_sample\_means\_n) \*\* 2  std\_dev\_of\_sample\_means\_n /= sample\_num  std\_dev\_of\_sample\_means\_n = **np**.sqrt(std\_dev\_of\_sample\_means\_n)  **print**("Standard deviation of sample means:", std\_dev\_of\_sample\_means)  **print**("Standard deviation of sample means:", std\_dev\_of\_sample\_means\_n) 3.2 在投资/交易中应用中心极限定理 在这一部分中，将CLT应用于投资/交易场景：  (1)场景：  - 一位投资者（Jyoti）正在考虑在未来五年内投资一只股票（ITC）。  - 投资者是风险厌恶型的，希望选择平均每月回报最好为非负数的股票。  - Jyoti希望使用一个统计框架来确定ITC股票的平均月回报并具有一定的置信度。  (2)步骤：  1. 理论框架：  - 讨论金融模型中的正态性假设，例如Black-Scholes期权定价模型。  - 引入CLT作为基于回报分布假设的统计框架。  *# standard imports*  import **pandas** as **pd**  import **numpy** as **np**  import **pyfolio** as **pf**  from **IPython**.**core**.**interactiveshell** import **InteractiveShell**  **InteractiveShell**.ast\_node\_interactivity = "all"  **pd**.**set\_option**('display.max\_rows', None)  *# Fetching the ITC stock data from yfinance library for the past 10 years*  ITC = **pd**.**read\_csv**('ITC.NS.csv')  daily\_data = ITC.**copy**().**round**(4)  *# Calculating daily Log returns*  daily\_data['daily\_return']= **np**.log(daily\_data['Adj Close']**/**daily\_data['Adj Close'].**shift**())  daily\_data.**dropna**(inplace=True)  *# Taking a peek at the fetched data*  daily\_data.**head**()  2. 数据分析：  - 获取过去10年的ITC股票每日收盘数据并进行分析。  - 可视化每日对数回报的分布，以评估正态性假设。  *# Visualizing the daily log returns*  **sns**.**set**(style="white", palette="muted", color\_codes=True)  **plt**.**figure**(figsize=(12,5))  **plt**.**subplot**(1,2,1)  *# Plot a simple histogram with binsize determined automatically.*  *# Please add the labels of the x, y axes and titles in the figure.*  *# Tips: sns.lineplot(daily\_data.index,daily\_data['daily\_return'], color="r")*  **plt**.**plot**(daily\_data.index, daily\_data['daily\_return'], color="r")  **plt**.**xlabel**("Date")  **plt**.**ylabel**("Daily Log Returns")  **plt**.**title**("Daily Log Returns Over Time")  **plt**.**subplot**(1,2,2)  *# Plot a simple histogram with binsize determined automatically.*  *# Please add the labels of the x, y axes and titles in the figure.*  *# Tips: sns.distplot(daily\_data['daily\_return'], kde=False, color="r")*  **sns**.**distplot**(daily\_data['daily\_return'], kde=False, color="r")  **plt**.**xlabel**("Daily Log Returns")  **plt**.**ylabel**("Frequency")  **plt**.**title**("Distribution of Daily Log Returns")  **plt**.**tight\_layout**()  **plt**.**show**()  (3)结论：  - 每日对数回报的可视化显示与正态性的偏离，突显了在金融数据中假设正态性的挑战。 3.3 更多可视化练习 在这一部分，使用底特律市的小时天气数据进行了额外的可视化练习：  1. 温度分布：  - 绘制了温度数据的直方图，使用100个区间以可视化分布。  - 将该分布与正态分布曲线进行了比较。  import **pandas** as **pd**  import **numpy** as **np**  import **seaborn** as **sns**  import **matplotlib**.**pyplot** as **plt**  from **scipy**.**stats** import norm  temperature\_data = **pd**.**read\_csv**('temperature.csv')  detroit\_temperature = temperature\_data['Detroit']  **plt**.**figure**(figsize=(12, 12))  **sns**.**histplot**(detroit\_temperature, bins=100, kde=False, color="blue", stat="density")  xmin, xmax = **plt**.**xlim**()  x = **np**.**linspace**(xmin, xmax, 100)  p = norm.**pdf**(x, detroit\_temperature.**mean**(), detroit\_temperature.**std**())  **plt**.**plot**(x, p, 'k', linewidth=2)  **plt**.**xlabel**("Temperature (C)")  **plt**.**ylabel**("Density")  **plt**.**title**("Distribution of Temperature in Detroit")  **plt**.**show**()  2. 多图可视化：  - 可视化了底特律市湿度、气压和温度的变化，分别在不同的图中。  - 创建了温度的7天移动平均线图。  - 进行了其他练习，包括在同一图中可视化不同城市的温度、湿度和气压以及创建城市可见性的交互复选框。  humidity\_data = **pd**.**read\_csv**('humidity.csv')  pressure\_data = **pd**.**read\_csv**('pressure.csv')  merged\_data = **pd**.**DataFrame**({      'Date': temperature\_data['datetime'],      'Temperature': temperature\_data['Detroit'],      'Humidity': humidity\_data['Detroit'],      'Pressure': pressure\_data['Detroit']  })  merged\_data['Date'] = **pd**.**to\_datetime**(merged\_data['Date'])  merged\_data.**set\_index**('Date', inplace=True)  resampled\_data = merged\_data.**resample**('D').**mean**()  window\_size = 7  smoothed\_data = resampled\_data.**rolling**(window=window\_size).**mean**()  **plt**.**figure**(figsize=(12, 12))  *# 湿度*  **plt**.**subplot**(3, 1, 1)  **plt**.**plot**(smoothed\_data.index, smoothed\_data['Humidity'], color='b')  **plt**.**xlabel**("Date")  **plt**.**ylabel**("Humidity")  **plt**.**title**("Smoothed Humidity Variation in Detroit")  *# 气压*  **plt**.**subplot**(3, 1, 2)  **plt**.**plot**(smoothed\_data.index, smoothed\_data['Pressure'], color='r')  **plt**.**xlabel**("Date")  **plt**.**ylabel**("Pressure")  **plt**.**title**("Smoothed Pressure Variation in Detroit")  *# 温度*  **plt**.**subplot**(3, 1, 3)  **plt**.**plot**(smoothed\_data.index, smoothed\_data['Temperature'], color='g')  **plt**.**xlabel**("Date")  **plt**.**ylabel**("Temperature (C)")  **plt**.**title**("Smoothed Temperature Variation in Detroit")  **plt**.**tight\_layout**()  **plt**.**show**()  3. 天气数据探索：  - 通过可视化探索了温度、湿度和气压的变化。  import **pandas** as **pd**  import **matplotlib**.**pyplot** as **plt**  import **ipywidgets** as **widgets**  from **IPython**.**display** import **display**  temperature\_data = **pd**.**read\_csv**('temperature.csv')  temperature\_data['datetime'] = **pd**.**to\_datetime**(temperature\_data['datetime'])  temperature\_data.**set\_index**('datetime', inplace=True)  resampled\_data = temperature\_data.**resample**('D').**mean**()  window\_size = 7  smoothed\_data = resampled\_data.**rolling**(window=window\_size).**mean**()  **plt**.**figure**(figsize=(12, 8))  initial\_cities = ['Vancouver', 'Portland', 'San Francisco', 'Seattle', 'Los Angeles']  lines = {city: {'line': None, 'visible': True} for city in initial\_cities}  for city in initial\_cities:      lines[city]['line'], = **plt**.**plot**(smoothed\_data.index, smoothed\_data[city], label=city)  **plt**.**xlabel**("Date")  **plt**.**ylabel**("7-Day Moving Average Temperature")  **plt**.**title**("7-Day Moving Average Temperature for Different Cities")  **plt**.**legend**()  checkbox\_options = {city: **widgets**.**Checkbox**(description=city, value=True) for city in initial\_cities}  checkboxes = [checkbox\_options[city] for city in initial\_cities]  checkbox\_container = **widgets**.**HBox**(checkboxes)  **display**(checkbox\_container)  def **update\_visibility**(change):      for city, checkbox **in** checkbox\_options.**items**():          lines[city]['visible'] = checkbox.value          lines[city]['line'].set\_visible(lines[city]['visible'])  **print**(f"City: {city}, Visible: {lines[city]['visible']}")  **plt**.**draw**()  for checkbox in checkboxes:      checkbox.**observe**(**update\_visibility**, names='value')  **plt**.**show**() |
| Data Logging and Processing: 3.1 通过Python模拟演示中心极限定理（CLT）3.1.1 实验1 - 指数分布总体 数据获取：  for i in **range**(1, 51):      exponential\_sample = **np**.**random**.exponential(theta, sample\_size)      col = f'sample {i}'      df2[col] = exponential\_sample   3.1.2 实验2 - 二项分布总体 数据获取：  sample\_size = 500  sample\_num = 50  df500 = **pd**.**DataFrame**(index=**range**(1, sample\_size + 1))  for i in **range**(1, sample\_num + 1):      exponential\_sample\_large = **np**.**random**.exponential(theta, sample\_size)      col = f'sample {i}'      df500[col] = exponential\_sample\_large   3.2 在投资/交易中应用中心极限定理 数据获取：  从yfinance库中获取了过去10年的ITC股票数据，并将其保存到名为 ITC 的数据框中。  *# Fetching the ITC stock data from yfinance library for the past 10 years*  ITC = **pd**.**read\_csv**('ITC.NS.csv')  数据处理：  首先，通过复制 ITC 数据框并将所有数值舍入到小数点后4位，创建了一个名为 daily\_data 的新数据框。接着，计算了每日的对数收益率，该收益率通过取当日收盘价与前一日收盘价的比值，并对该比值取自然对数得到。最后，删除了包含缺失值的行，确保数据的完整性。  daily\_data = ITC.**copy**().**round**(4)  *# Calculating daily Log returns*  daily\_data['daily\_return']= **np**.log(daily\_data['Adj Close']**/**daily\_data['Adj Close'].**shift**())  daily\_data.**dropna**(inplace=True) 3.3 更多可视化练习 1. 温度分布：  *# 读取小时天气数据集*  temperature\_data = **pd**.**read\_csv**('temperature.csv')  *# 提取底特律市的温度记录*  detroit\_temperature = temperature\_data['Detroit']  2. 多图可视化：  数据获取：  *# 读取湿度和气压数据*  humidity\_data = **pd**.**read\_csv**('humidity.csv')  pressure\_data = **pd**.**read\_csv**('pressure.csv')  数据处理：七天连续平滑  *# 使用移动平均来平滑曲线*  window\_size = 7  *# 7天的移动平均窗口*  smoothed\_data = resampled\_data.**rolling**(window=window\_size).**mean**()  3. 天气数据探索：  *# 读取小时天气数据集*  temperature\_data = **pd**.**read\_csv**('temperature.csv')  *# 将日期列设置为索引*  temperature\_data['datetime'] = **pd**.**to\_datetime**(temperature\_data['datetime'])  temperature\_data.**set\_index**('datetime', inplace=True)  *# 降采样数据，每天一个数据点，使用每天的平均值*  resampled\_data = temperature\_data.**resample**('D').**mean**()  *# 使用移动平均来平滑曲线*  window\_size = 7  *# 7天的移动平均窗口*  smoothed\_data = resampled\_data.**rolling**(window=window\_size).**mean**() |
| Experimental Results and Analysis: 3.1 通过Python模拟演示中心极限定理（CLT）3.1.1 实验1 - 指数分布总体： 1. 参数计算：  - 指数分布总体的速率参数 设置为4。  - 根据速率参数计算总体均值 和标准差 。  2. 小样本模拟 ：  - 从总体中抽取50个大小为2的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果表明，即使对于较小的样本量（如2），样本均值的分布看起来与指数总体的分布非常不同，并更像是对正态分布的糟糕逼近，呈现一些正偏斜。    3. 大样本模拟：  - 从总体中抽取50个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果表明，随着样本量的增加，样本均值的分布更加接近正态分布。     3.1.2 实验2 - 二项分布总体 模拟步骤：  1. 大样本模拟：  - 从二项分布总体中抽取50个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。  - 结果显示，与指数分布类似，随着样本量的增加，样本均值的分布趋向于正态分布。        2. 大样本模拟：  - 从二项分布总体中抽取5000个大小为500的随机样本。  - 为每个样本计算样本均值，并绘制这些样本均值的分布。        - 结果显示，与指数分布类似，随着样本量的增加，样本均值的分布趋向于正态分布。 3.2 在投资/交易中应用中心极限定理 2. 数据分析：  - 获取过去10年的ITC股票每日收盘数据并进行分析。  - 可视化每日对数回报的分布，以评估正态性假设。  actual-values-of-daily-log-returns-over-time-1distribution-of-daily-log-returns  (3)结论：  我们可以观察到，在我们使用的这十年中，ITC的每日收益率集中在0附近。然而，曾经有多次超过上涨五个百分点的情况，甚至在下跌方面达到了负十个百分点。  这就是高峰度或者说厚尾的原因，这在正态分布中是不会出现的。每日收益率分布的图表看起来也是尖峭的。 3.3 更多可视化练习 在这一部分，使用底特律市的小时天气数据进行了额外的可视化练习：  1. 温度分布：  - 绘制了温度数据的直方图，使用100个区间以可视化分布。  - 将该分布与正态分布曲线进行了比较。    2. 多图可视化：  - 可视化了底特律市湿度、气压和温度的变化，分别在不同的图中。  - 创建了温度的7天移动平均线图。  - 进行了其他练习，包括在同一图中可视化不同城市的温度、湿度和气压以及创建城市可见性的交互复选框。      3. 天气数据探索：  - 通过可视化探索了温度、湿度和气压的变化。    这些练习演示了如何使用连续分布来近似天气数据的观察分布，并探索了温度、湿度和气压变化的各个方面。 |
| 关于中心极限定理的我的一些理解： 在课堂上，我们对X的平均分布情况进行了研究，通过实验绘制了X的平均频率分布图。当实验次数足够多时，这个图像近似于正态分布曲线。然而，需要注意的是，图中的纵坐标表示的是频率而非概率。  若我们绘制 平均的概率分布图，或许能更好地理解。在这种情况下，影响正态分布的唯一因素就是样本大小 ，即 求和的变量数。  接下来以投掷骰子为例：  如果有两个骰子，其点数之和的取值范围为 2-12；  如果有三个骰子，其点数之和的取值范围为 3-18；  ... ...  如果有n个骰子，其点数之和的取值范围为 n-6n;  这样就好理解了，对和进行取平均。得到的就是 1-6中5的单位长度进行 5n+1 划分，对应的 n 越大，划分的就越细致，也就越逼近正态分布。具体看下图：  以投掷骰子为例，假设  有两个骰子，它们的点数之和的取值范围为2-12；  有三个骰子，则点数之和的范围为3-18；以此类推。  如果有n个骰子，点数之和的取值范围为n-6n。  通过对这些和值取平均，我们将1-6中的每个5的单位长度分成5n+1个部分，其中n越大，划分越细致，逼近正态分布。具体请参考下图：        代码如下：  用到了from ipywidgets import interact, widgets中的交互式工具实现样本大小的动态选取。  from **ipywidgets** import interact, **widgets**  import **matplotlib**.**pyplot** as **plt**  from **IPython**.**core**.**interactiveshell** import **InteractiveShell**  **InteractiveShell**.ast\_node\_interactivity = "last"  def **calculate\_dice\_probability**(n):      dp = [[0] \* (6 \* n + 1) for \_ in **range**(n + 1)]      for j in **range**(1, 7):          dp[1][j] = 1      for i in **range**(2, n + 1):          for j in **range**(i, 6 \* i + 1):              for k in **range**(1, 7):                  if j - k > 0:                      dp[i][j] += dp[i - 1][j - k]      total\_ways = 6 \*\* n      probabilities = [count / total\_ways for count in dp[n][n:]]      return probabilities  def **update\_plot**(n):      result = **calculate\_dice\_probability**(n)  **plt**.**clf**()  **plt**.**figure**(figsize=(10, 5))  **plt**.**bar**(**range**(n, 6 \* n + 1), result)  **plt**.**xlabel**('Sum of Dice')  **plt**.**ylabel**('Probability')  **plt**.**title**(f'Probability Distribution of Dice Sum ({n} Dice)')  **plt**.**show**()  dice\_slider = **widgets**.**IntSlider**(value=1, min=1, max=50, step=1, description='骰子个数')  interact(**update\_plot**, n=dice\_slider) |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |