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：Application of 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

报告人：杨烨 学号：2022280380

班级：06

实验时间：2023 年 11 月 10 日

实验报告提交时间：2023 年 11 月 30 日

教务处制

Aim of Experiment:

1. Familiar with the central limit theorem.
2. Understand the implementation of the central limit theorem in python.
3. Know how to visualize data in different distributions.
4. Familiar with seaborn, a powerful visual database in python;

Experiment Content:

The Central Limit Theorem (CLT) is often referred to as one of the most important theorems, not only in probability & statistics but also in the sciences as a whole.

Try a Python simulation to understand the nature of the central limit theorem.

Experiment Process：

Firstly, we should be familiar with the relevant background knowledge of the Central Limit Theorem and the knowledge points needed for this experiment. Samples and the Sampling Distribution, Central Limit Theorem - Statement & Assumptions

$$\bar{x} \longrightarrow \mathbb{N}\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

Fig1 the Central Limit Theorem

This experiment is divided into **three sections:**

(1) **Demonstration of CLT in action using simulations in Python；**

①Experiment 1 - Exponentially distributed population

②Experiment 2 - Binomially distributed population

(2) **An Application of CLT in Investing/Trading；**

(3) **More visualization exercises；**

①Hourly temperature records in Detroit；

②Hourly humidity, temperature, air pressure, and wind speed records in Detroit；

③Temperature change curves of four cities；

The following is the specific process of this experiment：

(1) Demonstration of CLT in action using simulations in Python:

①Experiment 1 - **Exponentially distributed** population:

$$f(x) = \begin{cases} \frac{1}{\theta} e^{-\frac{x}{\theta}} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Fig2 Density function of exponential distribution

Suppose that **θ=4**；Calculate the mean and variance of the exponential distribution.

Implementation code：

```
# rate parameter for the exponentially distributed population
theta = 4.0

# Population mean (mu), representing mean by parameter theta

#
# YOUR CODE HERE
mu = theta
#

# Population standard deviation (sd), representing sd by parameter theta

#
# YOUR CODE HERE
variance = theta**2
sd = np.sqrt(variance)
#
```

Fig3 Code to calculate the mean and variance

Keep the original form and use the **Dataframe** data type in the pandas library to implement the operation, Plot of an exponential distribution sample with a sample size of 500.

```
# Please refer to the previous example to draw 50 random samples of size 500
# Tips: You can use the "np.random.exponential" function to implement a exponential distribution.

#
# YOUR CODE HERE\

sample_size = 500 # 样本大小修改成500每列
df500 = pd.DataFrame(index=range(1, sample_size + 1))

for i in range(1, 51):
    exponential_sample = np.random.exponential(theta, sample_size)
    col = f'sample {i}'
    df500[col] = exponential_sample

# Taking a peek at the samples
df500

# Calculating sample means and plotting their distribution
df500_sample_means = df500.mean()
sns.distplot(df500_sample_means)
```

Fig4 Code to calculate the mean and variance

②Experiment 2 - **Binomially distributed** population:

$$P(x) = \begin{cases} \binom{k}{x}(p)^x(1-p)^{1-x} & \text{if } x = 0, 1, 2, \ldots, k \\ 0 & \text{otherwise} \end{cases}$$

Fig5 Binomially distributed

Suppose that k=30 and p=0.9；

Calculate the mean and variance of the exponential distribution：

Implementation code：

```
# YOUR CODE HERE


sample_size = 500
k = 30
p = 0.9
df500_binomial = pd.DataFrame(index= range(1, sample_size + 1))

for i in range(1, 51):
    binomial_sample = np.random.binomial(k, p, sample_size)
    col = f'sample {i}'
    df500_binomial[col] = binomial_sample

# Taking a peek at the samples
df500_binomial

# Calculating sample means and plotting their distribution
df500_binomial_sample_means = df500_binomial.mean()
sns.distplot(df500_binomial_sample_means)
```

Fig6 Code to calculate the mean and variance

(2) An Application of CLT in Investing/Trading:

In financial models, we often use relevant knowledge of central limit theorem for investment and return. For analysis, we first import some standard Python libraries and obtain daily closing price ITC stock data from yfinance library.

Use 1*2 subgraphs to plot the result:

```
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")

#
# YOUR CODE HERE

sns.lineplot(daily_data['daily_return'], color="r")
plt.xlabel('Date')  # 添加 x 轴标签
plt.ylabel('Daily Log Return')  # 添加 y 轴标签
plt.title('Daily Log Returns')  # 添加图表标题

#

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")
#  系统警告说，使用histplot函数是更好的方案

#
# YOUR CODE HERE

sns.histplot(daily_data['daily_return'], kde=False, color="r")
plt.xlabel('Daily Log Return')  # 添加 x 轴标签
plt.ylabel('Frequency')  # 添加 y 轴标签
plt.title('Distribution of Daily Log Returns')  # 添加图表标题
#
```

Fig7 Implement 1*2 subgraphs to present data visualization code

(3) **More visualization exercises；**

A continuous random variable represents an infinite number of possible outcomes. For a sample of a continuous random variable X, we may not be able to cover the entire sample space. Since the sample space has an infinite number of observations, we cannot accurately estimate its exact distribution, so a parameterized continuous distribution can be used to approximate the observed distribution. In this task, we will use hourly weather data sets for the city of Detroit to approximate temperature records in a continuous distribution.

①Hourly temperature records in **Detroit**；

```
# YOUR CODE HERE
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
import numpy as np

data = pd.read_csv('temperature.csv', parse_dates=[0], index_col=0)
detroit_data = data['Detroit']  # 假设底特律的数据列名为 'Detroit'

plt.hist(detroit_data, bins=100)
plt.xlabel('Temperature')
plt.ylabel('Frequency')
plt.title('Histogram of Temperature in Detroit')
plt.show()
```

Fig8　A code that presents hourly temperature data for Detroit

②Hourly humidity, temperature, air pressure, and wind speed records in Detroit;

```
humidity_data = pd.read_csv('humidity.csv', parse_dates=[0], index_col=0)
temperature_data = pd.read_csv('temperature.csv', parse_dates=[0], index_col=0)
pressure_data = pd.read_csv('pressure.csv', parse_dates=[0], index_col=0)
wind_speed_data = pd.read_csv('wind_speed.csv', parse_dates=[0], index_col=0)

detroit_humidity = humidity_data['Detroit']
detroit_temperature = temperature_data['Detroit']
detroit_pressure = pressure_data['Detroit']
detroit_wind_speed = wind_speed_data['Detroit']

fig, axs = plt.subplots(2, 2, figsize=(10, 8))

axs[0, 0].hist(detroit_humidity, bins=30)
axs[0, 0].set_xlabel('Humidity')
axs[0, 0].set_ylabel('Frequency')
axs[0, 0].set_title('Histogram of Humidity in Detroit')

axs[0, 1].hist(detroit_temperature, bins=30)
axs[0, 1].set_xlabel('Temperature')
axs[0, 1].set_ylabel('Frequency')
axs[0, 1].set_title('Histogram of Temperature in Detroit')

axs[1, 0].hist(detroit_pressure, bins=30)
axs[1, 0].set_xlabel('Pressure')
axs[1, 0].set_ylabel('Frequency')
axs[1, 0].set_title('Histogram of Pressure in Detroit')

axs[1, 1].hist(detroit_wind_speed, bins=30)
axs[1, 1].set_xlabel('Wind Speed')
axs[1, 1].set_ylabel('Frequency')
axs[1, 1].set_title('Histogram of Wind Speed in Detroit')

plt.tight_layout()
plt.show()
```

Fig 9　A code that presents hourly humidity, temperature, air pressure, and wind speed data for Detroit

③Temperature change curves of four cities;

```
In [18]: import pandas as pd
         import matplotlib.pyplot as plt

         data = pd.read_csv('temperature.csv', parse_dates=[0], index_col=0)

         cities = ['Detroit', 'New York', 'Chicago', 'Los Angeles']
         fig, axs = plt.subplots(2, 2, figsize=(10, 8))

         for i, city in enumerate(cities):
             row = i // 2
             col = i % 2
             city_data = data[city]
             axs[row, col].plot(city_data.index, city_data.values)
             axs[row, col].set_xlabel('Date')
             axs[row, col].set_ylabel('Temperature (° C)')
             axs[row, col].set_title(f'Temperature in {city}')

         plt.tight_layout()
         plt.show()
```

Fig10 Code that shows the temperature curve of four cities

Data Logging and Processing:

(1) Demonstration of CLT in action using simulations in Python:

①Experiment 1 - Exponentially distributed population

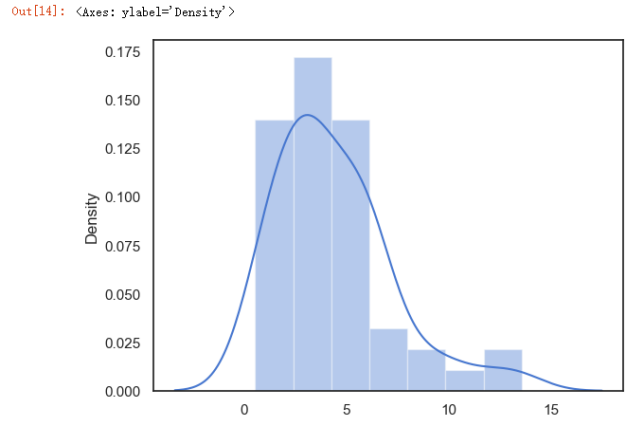

Fig11 Exponential distribution of small sample size (Size:2)

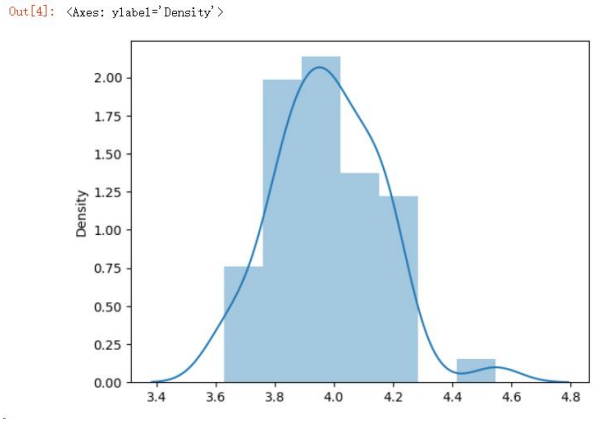

Fig12 Exponential distribution of large sample size (Size: 500)

We can observe that the mean of all the sample means is very close to the population: mean (μ =4):

```
In  [5]: #The first 5 values from the 50 sample means
         df500_sample_means.head()

Out[5]: sample 1    4.039621
        sample 2    4.056470
        sample 3    4.008401
        sample 4    4.079657
        sample 5    3.864268
        dtype: float64
```

Fig13 The result of the calculation of the average of the first five samples

Similarly, we can observe that the standard deviation of the 50 sample means is quite close to the value stated by the CLT, $\sigma / \sqrt{n} = 0.178$:

```
Standard deviation of sampling distribution: 0.1782637369110422
```

Fig14 The result of standard deviation calculation

②Experiment 2 - Binomially distributed population:

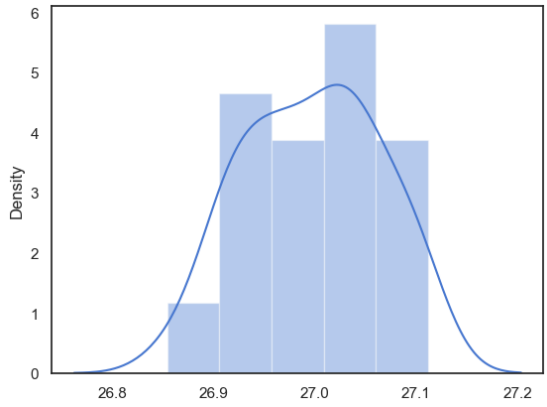

Fig15 Binomially distributed of large sample size (Size: 500)

The sampling distribution should be approximately normal with mean population:

Theoretical value:

mean = 27;

standard deviation =0.0734;

Actual value:

Mean of sample means: 26.99712;

Standard deviation of sampling distribution: 0.0779756;

(2)An Application of CLT in Investing/Trading:

Taking a peek at the fetched data:

	Date	Adj Close	daily_return
1	2010/1/5	65.9113	0.009808
2	2010/1/6	66.0656	0.002338
3	2010/1/7	65.8598	-0.003120
4	2010/1/8	66.0013	0.002146
5	2010/1/11	66.0270	0.000389

Fig16　First five data

Now that we have the daily log returns for ITC. Visualize both the returns and their distribution according to the diagrams below:

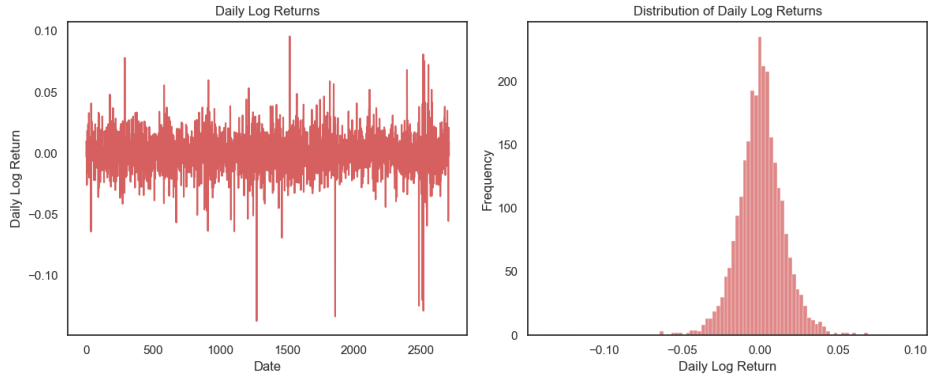

Fig17　Daily Closing price - Visualization of closing price ITC stock data

(3)More visualization exercises；

①Hourly temperature records in Detroit；

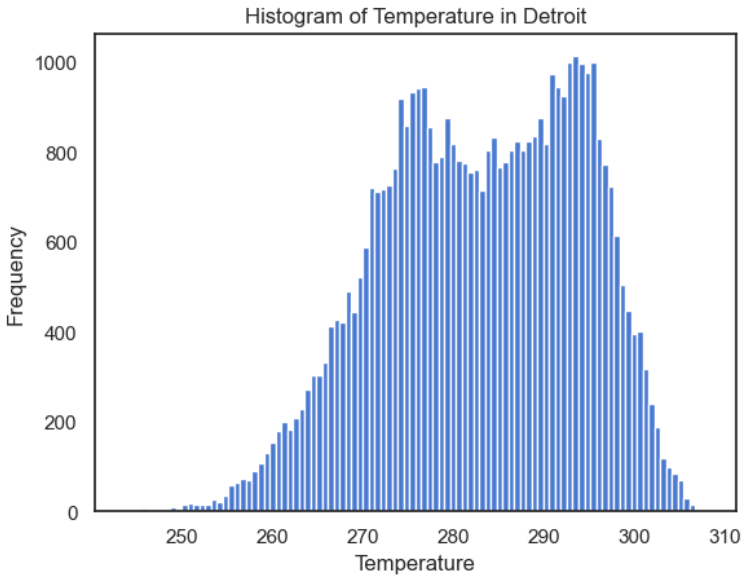

Fig18　Hourly temperature records in Detroit；

②Hourly humidity, temperature, air pressure, and wind speed records in Detroit；

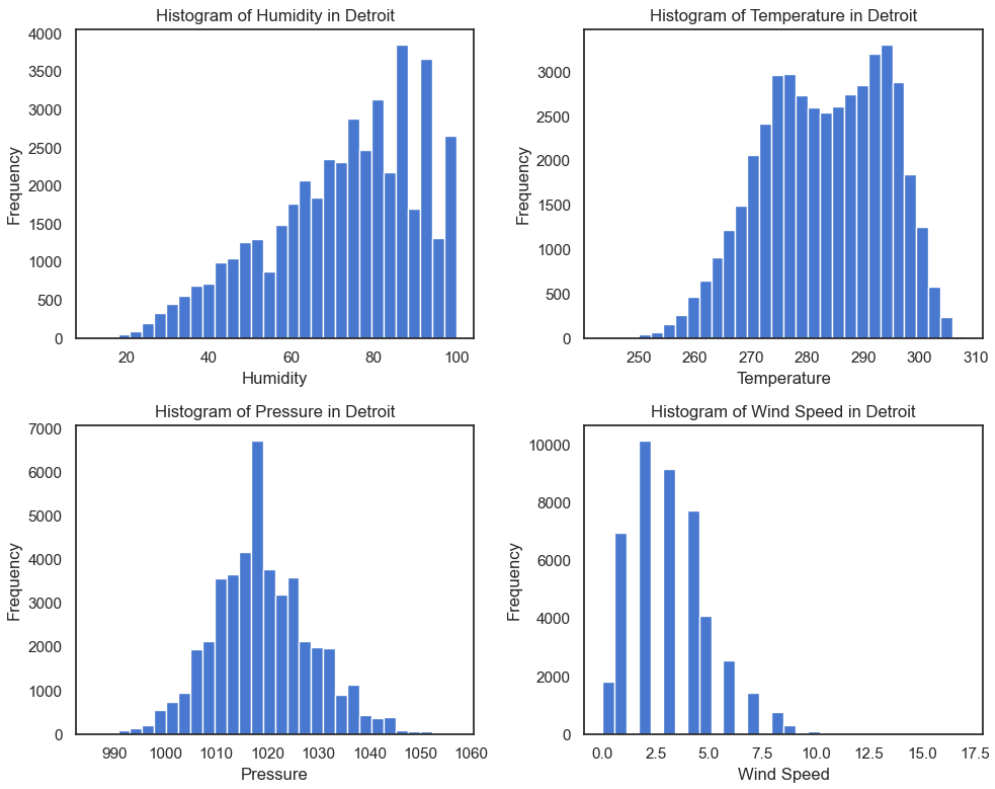

Fig19　Hourly humidity, temperature, air pressure, and wind speed records in Detroit；

③Temperature change curves of four cities；

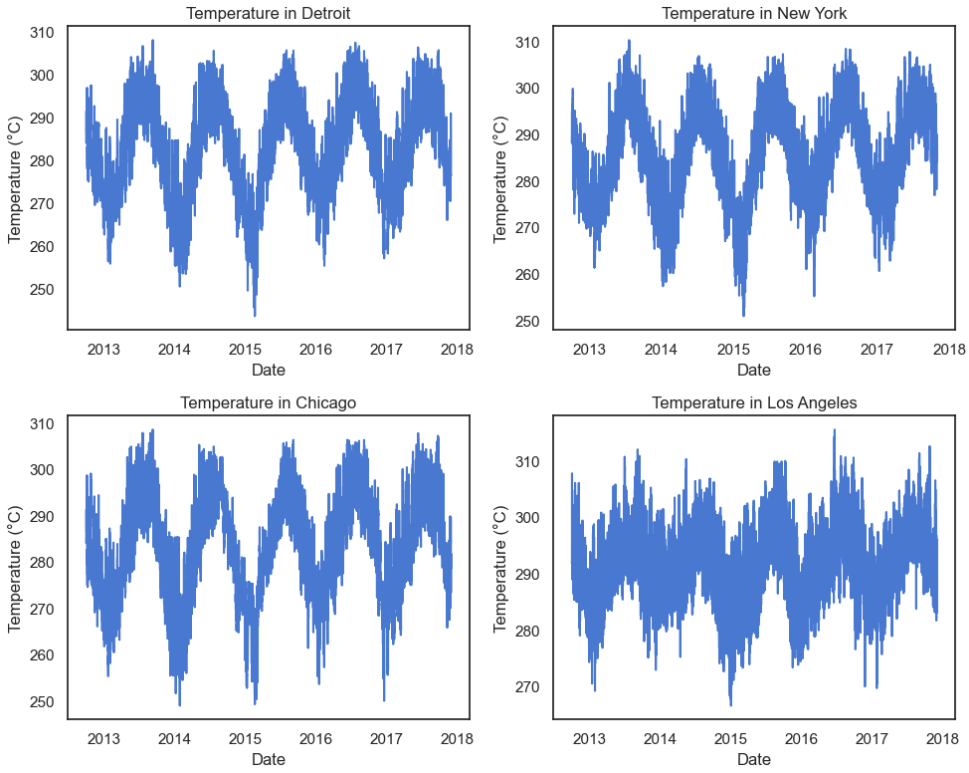

Fig 20　Temperature change curves of four cities

Experimental Results and Analysis:

The experimental results are as follows:

In this experiment, I successfully realized the sampling analysis of exponential distribution and binomial distribution, drew the histogram of relative frequency distribution and calculated the corresponding mean and variance; We also successfully completed the data analysis of financial models and the visualization of different weather indicators for various cities in the United States.

Analysis:

There are also some parts of this experiment that can be improved:

Deeper theoretical exploration: For each distribution and model, the mathematical and statistical theory behind it can be further explored to better understand its application scenarios and limitations.

More comprehensive data analysis: In terms of financial models and weather indicator visualization, more data sets and variables can be explored and more comprehensive data analysis and visual presentation can be performed.

More detailed experimental design: When sampling analysis is performed, more experimental sessions and different parameter combinations can be designed to more fully explore the nature and characteristics of these distributions.

指导教师批阅意见：

成绩评定：

指导教师签字：

年　　月　　日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。