哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

开课学期：2024 秋季

课程名称：统计计算

实验名称：实验一

学生班级：3 班

学生学号：220810332

学生姓名：斯蓬

实验与创新实践教育中心制

2024 年 1 月

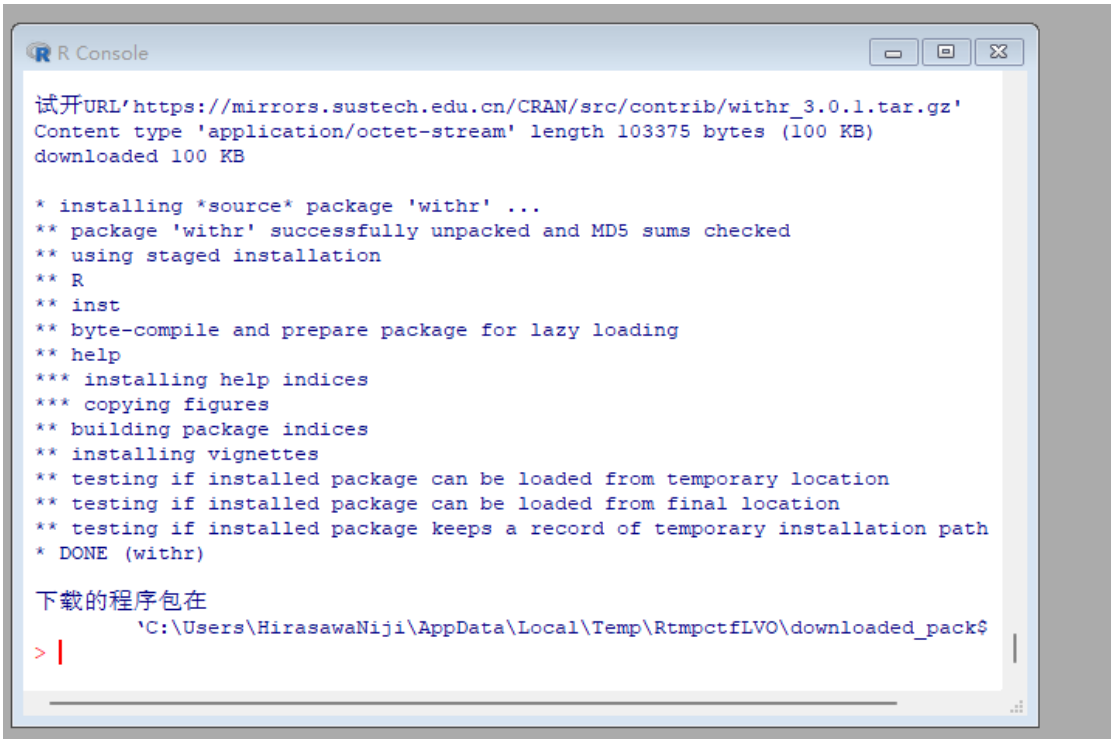
一、 实验环境

操作系统：Windows11 23H2

环境：R 4.2.1、Python 3.11

二、 实验内容

1、安装好 R（以及 Rstudio），熟悉安装过程，并尝试安装一个宏包，并阅读其说明文档。



如图，安装了 ggplot2 包

2、生成如下随机数并画其散点图和直方图

(1) U(0,1) 随机数 10 个

(2) U(0,1) 随机数 10000 个

(3) U(1,10) 随机数 10 个

(4) N(0,1) 随机数 10 个

(5) N(0,1) 随机数 10000 个

(6) N(1, 5) 随机数 10000 个

(7) Binomial(1, 0.5) 随机数 10 个

(8) Binomial(1, 0.9) 随机数 10 个

(9) Binomial(10,0.5) 随机数 10 个

(10)Binomial(10, 0.9) 随机数 10 个

以下是 R 语言的代码：

```
# 加载绘图包
library(ggplot2)

# 设置随机数种子
set.seed(111)

# U(0,1) 随机数 10 个
u_01_10 <- runif(10, 0, 1)
# U(0,1) 随机数 10000 个
u_01_10000 <- runif(10000, 0, 1)
# U(1,10) 随机数 10 个
u_1_10_10 <- runif(10, 1, 10)
# N(0,1) 随机数 10 个
n_01_10 <- rnorm(10, 0, 1)
# N(0,1) 随机数 10000 个
n_01_10000 <- rnorm(10000, 0, 1)
# N(1, 5) 随机数 10000 个
n_15_10000 <- rnorm(10000, 1, 5)
# Binomial(1, 0.5) 随机数 10 个
binom_1_05_10 <- rbinom(10, 1, 0.5)
# Binomial(1, 0.9) 随机数 10 个
binom_1_09_10 <- rbinom(10, 1, 0.9)
# Binomial(10, 0.5) 随机数 10 个
binom_10_05_10 <- rbinom(10, 10, 0.5)
# Binomial(10, 0.9) 随机数 10 个
binom_10_09_10 <- rbinom(10, 10, 0.9)
```

```
# 生成散点图和直方图函数
plot_random <- function(data, title) {
  par(mfrow=c(1, 2))  # 设置一行两列
  plot(data, main=paste(title, "Scatter Plot"), pch=19)
  hist(data, main=paste(title, "Histogram"), col="lightblue")
}

# 画图
par(mfrow=c(5, 4))  # 设置布局

plot_random(u_01_10, "U(0,1) 10")
plot_random(u_01_10000, "U(0,1) 10000")
plot_random(u_1_10_10, "U(1,10) 10")
plot_random(n_01_10, "N(0,1) 10")
plot_random(n_01_10000, "N(0,1) 10000")
plot_random(n_15_10000, "N(1,5) 10000")
plot_random(binom_1_05_10, "Binom(1,0.5) 10")
plot_random(binom_1_09_10, "Binom(1,0.9) 10")
plot_random(binom_10_05_10, "Binom(10,0.5) 10")
plot_random(binom_10_09_10, "Binom(10,0.9) 10")
```

下面是 Python 的代码：

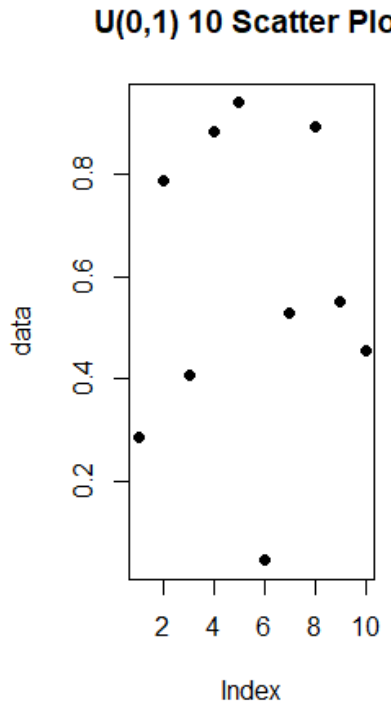
```
import numpy as np
import matplotlib.pyplot as plt

# U(0,1) random numbers
u_01_10 = np.random.uniform(0, 1, 10)
u_01_10000 = np.random.uniform(0, 1, 10000)

# U(1,10) random numbers
u_1_10_10 = np.random.uniform(1, 10, 10)

# N(0,1) random numbers
n_01_10 = np.random.normal(0, 1, 10)
n_01_10000 = np.random.normal(0, 1, 10000)

# N(1, 5) random numbers
n_15_10000 = np.random.normal(1, 5, 10000)
```

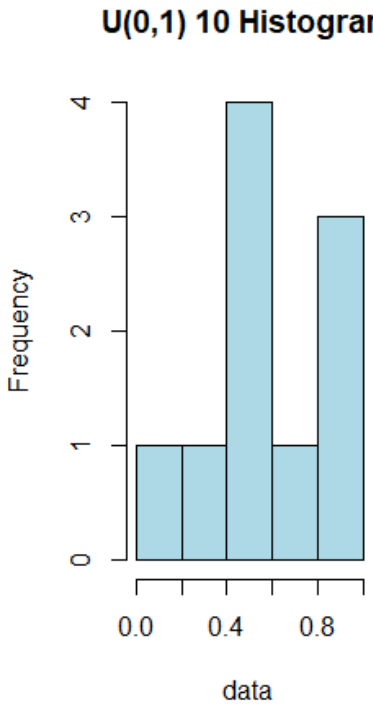
```
# Binomial random numbers
binom_1_05_10 = np.random.binomial(1, 0.5, 10)
binom_1_09_10 = np.random.binomial(1, 0.9, 10)
binom_10_05_10 = np.random.binomial(10, 0.5, 10)
binom_10_09_10 = np.random.binomial(10, 0.9, 10)

# Plotting
fig, axs = plt.subplots(5, 4, figsize=(16, 16))
fig.suptitle('Random Number Generation and Plots', fontsize=16)

# U(0,1) random numbers 10
axs[0, 0].scatter(range(10), u_01_10)
axs[0, 0].set_title('U(0,1) 10 Scatter')
axs[0, 1].hist(u_01_10, bins=5)
axs[0, 1].set_title('U(0,1) 10 Histogram')

# U(0,1) random numbers 10000
axs[0, 2].scatter(range(10000), u_01_10000, s=1)
axs[0, 2].set_title('U(0,1) 10000 Scatter')
axs[0, 3].hist(u_01_10000, bins=50)
axs[0, 3].set_title('U(0,1) 10000 Histogram')

# U(1,10) random numbers 10
axs[1, 0].scatter(range(10), u_1_10_10)
axs[1, 0].set_title('U(1,10) 10 Scatter')
axs[1, 1].hist(u_1_10_10, bins=5)
axs[1, 1].set_title('U(1,10) 10 Histogram')

# N(0,1) random numbers 10
axs[1, 2].scatter(range(10), n_01_10)
axs[1, 2].set_title('N(0,1) 10 Scatter')
axs[1, 3].hist(n_01_10, bins=5)
axs[1, 3].set_title('N(0,1) 10 Histogram')

# N(0,1) random numbers 10000
axs[2, 0].scatter(range(10000), n_01_10000, s=1)
axs[2, 0].set_title('N(0,1) 10000 Scatter')
axs[2, 1].hist(n_01_10000, bins=50)
axs[2, 1].set_title('N(0,1) 10000 Histogram')

# N(1,5) random numbers 10000
axs[2, 2].scatter(range(10000), n_15_10000, s=1)
axs[2, 2].set_title('N(1,5) 10000 Scatter')
axs[2, 3].hist(n_15_10000, bins=50)
```

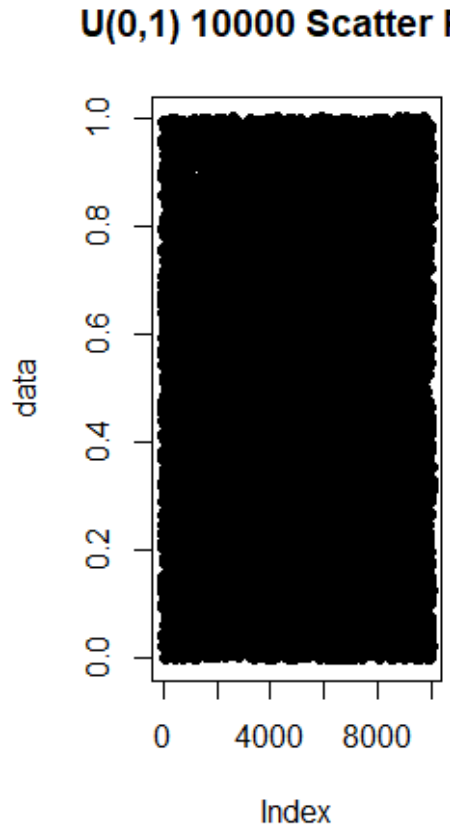
```
axs[2, 3].set_title('N(1,5) 10000 Histogram')

# Binomial(1, 0.5) random numbers 10
axs[3, 0].scatter(range(10), binom_1_05_10)
axs[3, 0].set_title('Binomial(1,0.5) 10 Scatter')
axs[3, 1].hist(binom_1_05_10, bins=2)
axs[3, 1].set_title('Binomial(1,0.5) 10 Histogram')

# Binomial(1, 0.9) random numbers 10
axs[3, 2].scatter(range(10), binom_1_09_10)
axs[3, 2].set_title('Binomial(1,0.9) 10 Scatter')
axs[3, 3].hist(binom_1_09_10, bins=2)
axs[3, 3].set_title('Binomial(1,0.9) 10 Histogram')

# Binomial(10,0.5) random numbers 10
axs[4, 0].scatter(range(10), binom_10_05_10)
axs[4, 0].set_title('Binomial(10,0.5) 10 Scatter')
axs[4, 1].hist(binom_10_05_10, bins=10)
axs[4, 1].set_title('Binomial(10,0.5) 10 Histogram')

# Binomial(10,0.9) random numbers 10
axs[4, 2].scatter(range(10), binom_10_09_10)
axs[4, 2].set_title('Binomial(10,0.9) 10 Scatter')
axs[4, 3].hist(binom_10_09_10, bins=10)
axs[4, 3].set_title('Binomial(10,0.9) 10 Histogram')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

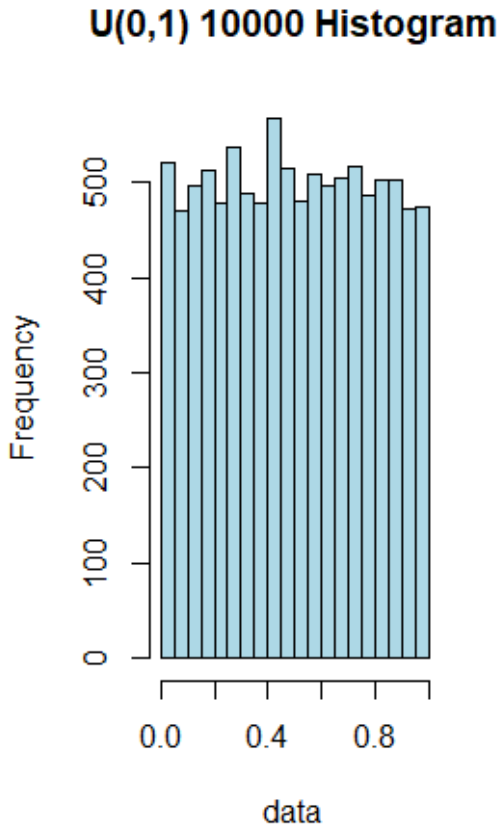
第一题运行结果：

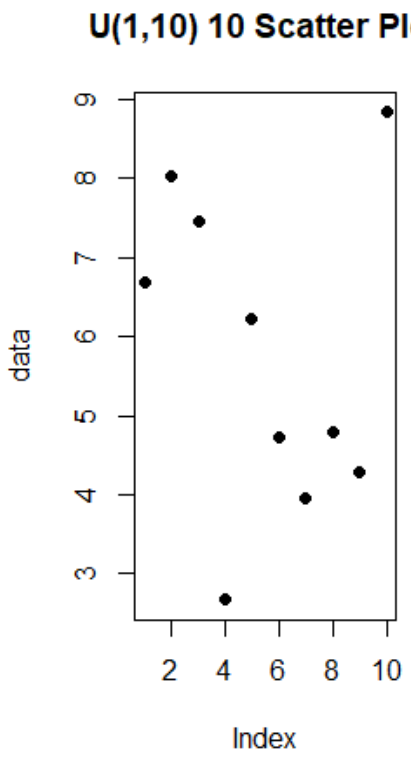
R 语言：

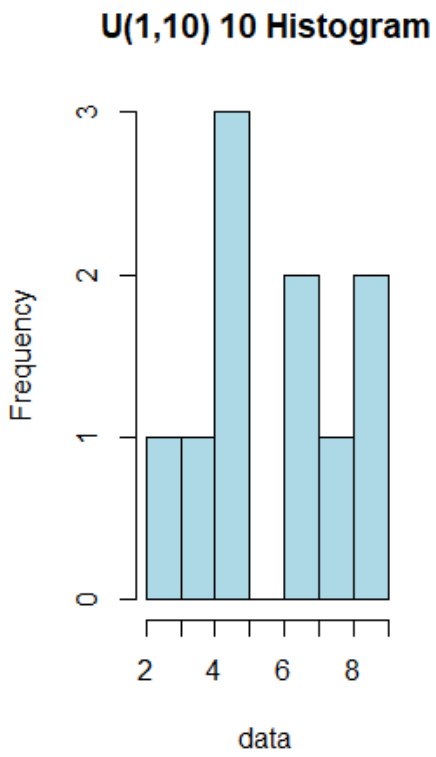

U(0,1) 10 Scatter Plot
data
0.2
0.4
0.6
0.8
2
4
6
8
10
Index

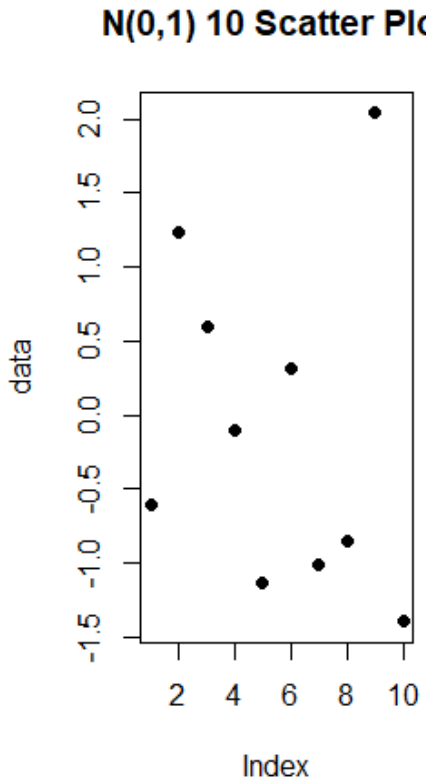

U(0,1) 10 Histogram
Frequency
0
1
2
3
4
0.0
0.4
0.8
data

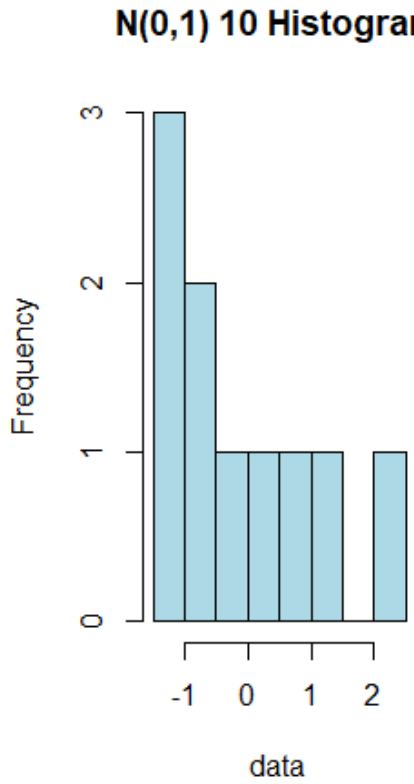

U(0,1) 10000 Scatter Plot
data
0.0
0.2
0.4
0.6
0.8
1.0
0
4000
8000
Index

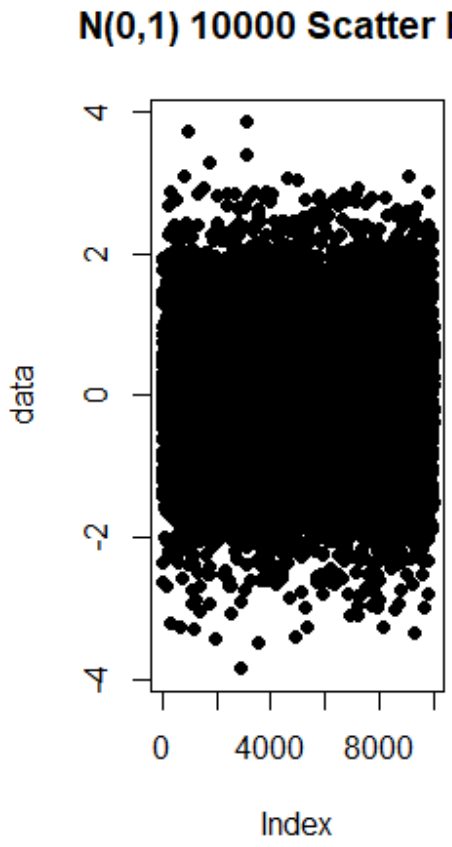

U(0,1) 10000 Histogram
Frequency
0
100
200
300
400
500
0.0
0.4
0.8
data

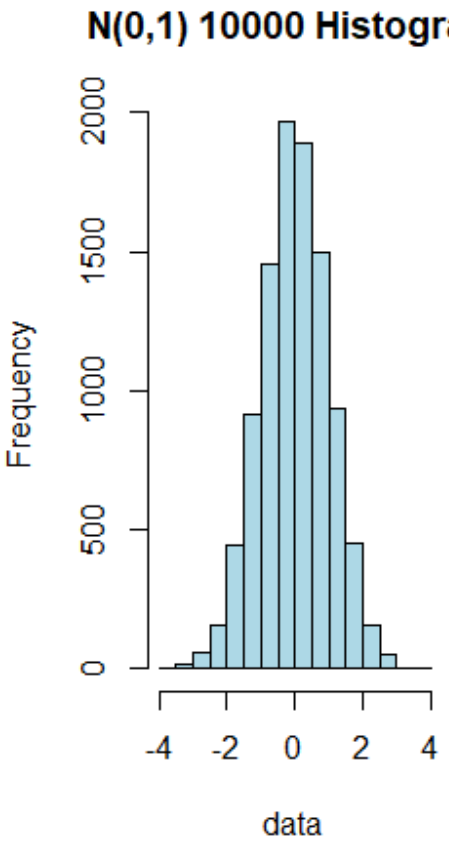

U(1,10) 10 Scatter Plot
data
Index
3
4
5
6
7
8
9
2
4
6
8
10

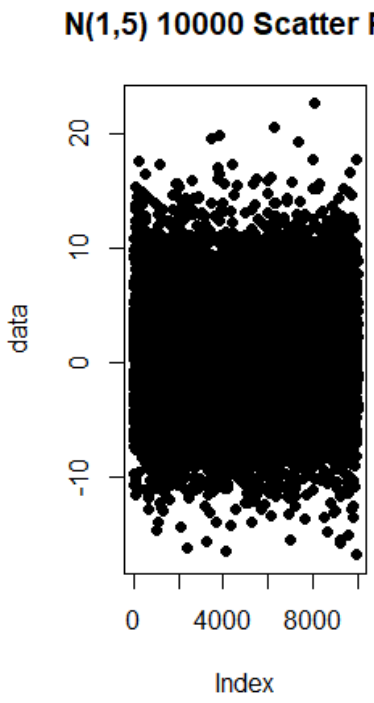

U(1,10) 10 Histogram
Frequency
data
0
1
2
3
2
4
6
8

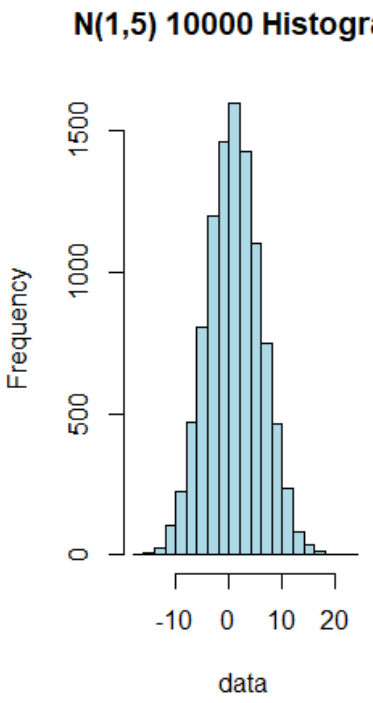

N(0,1) 10 Scatter Plot
data
Index
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
2
4
6
8
10

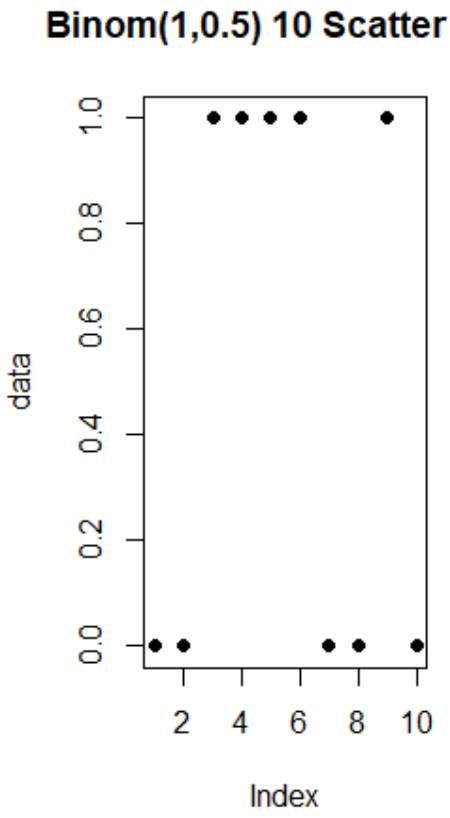

N(0,1) 10 Histogram
Frequency
data
0
1
2
3
-1
0
1
2

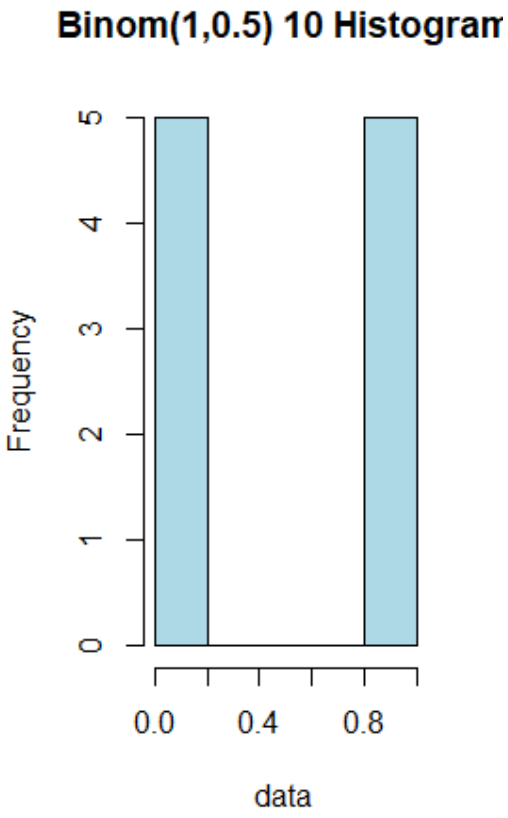

N(0,1) 10000 Scatter Plot
data
Index
4
2
0
-2
-4
0
4000
8000

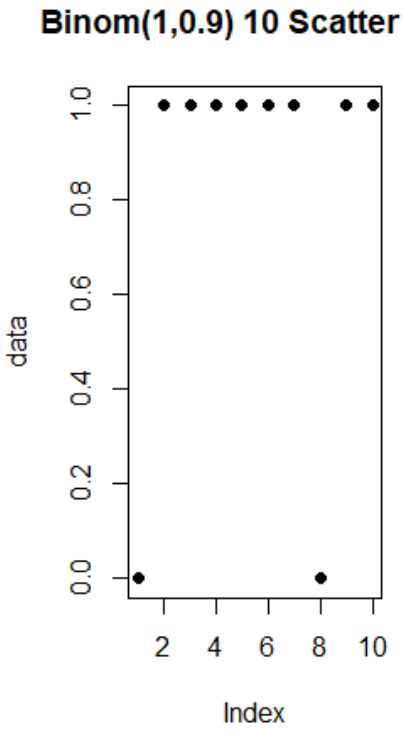

N(0,1) 10000 Histogram
Frequency
data
2000
1500
1000
500
0
-4
-2
0
2
4

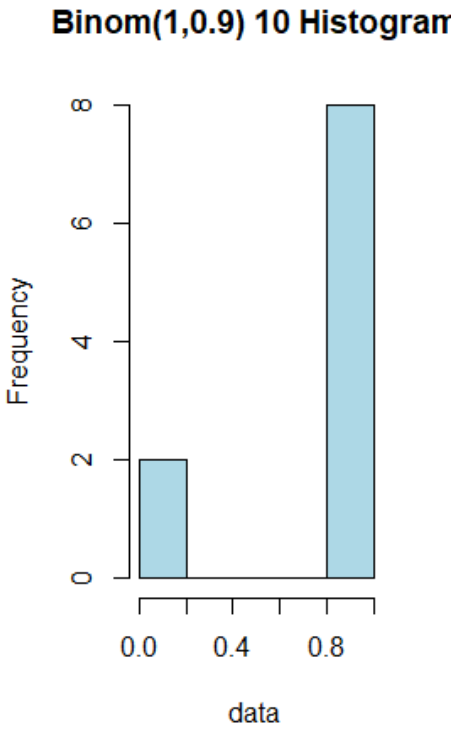

N(1,5) 10000 Scatter Plot
data
Index
20
10
0
-10
0
4000
8000

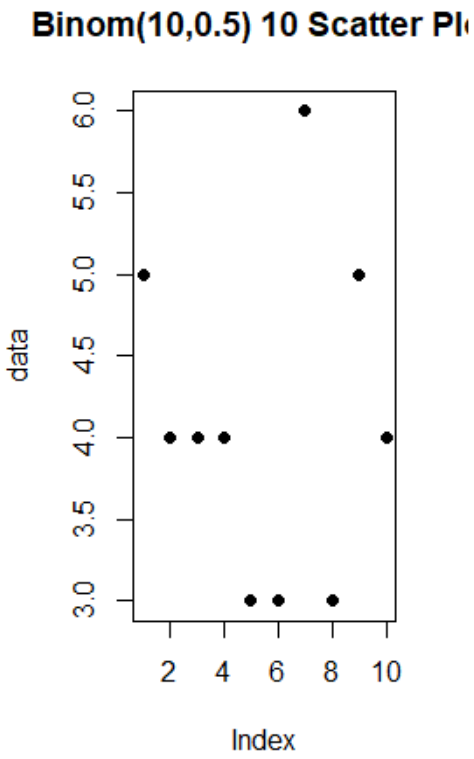

N(1,5) 10000 Histogram
Frequency
data
1500
1000
500
0
-10
0
10
20

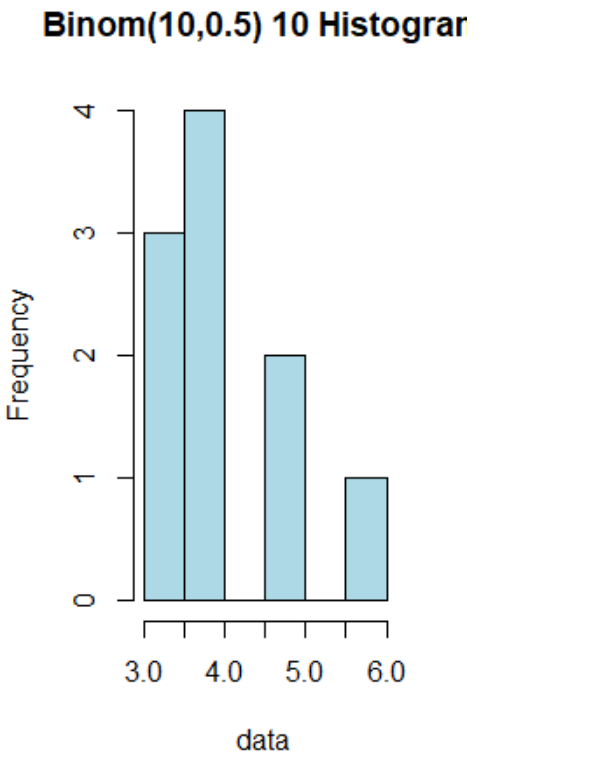

Binom(1,0.5) 10 Scatter Plc
data
Index
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10

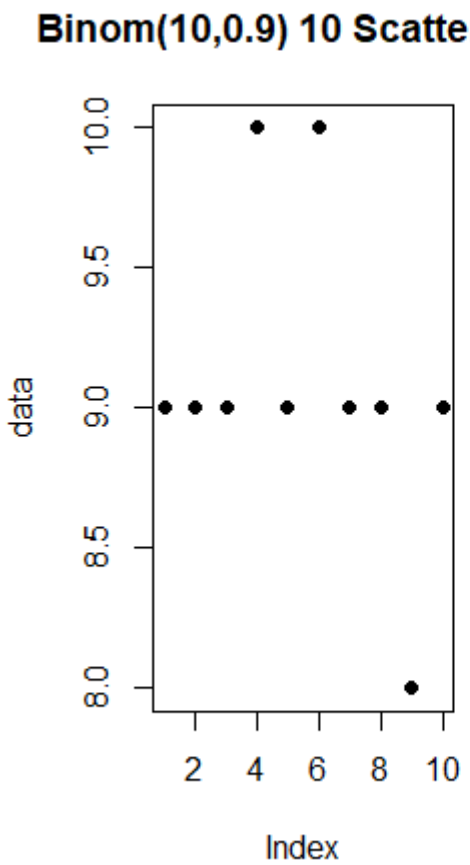

Binom(1,0.5) 10 Histogran
Frequency
data
0
1
2
3
4
5
0.0
0.4
0.8

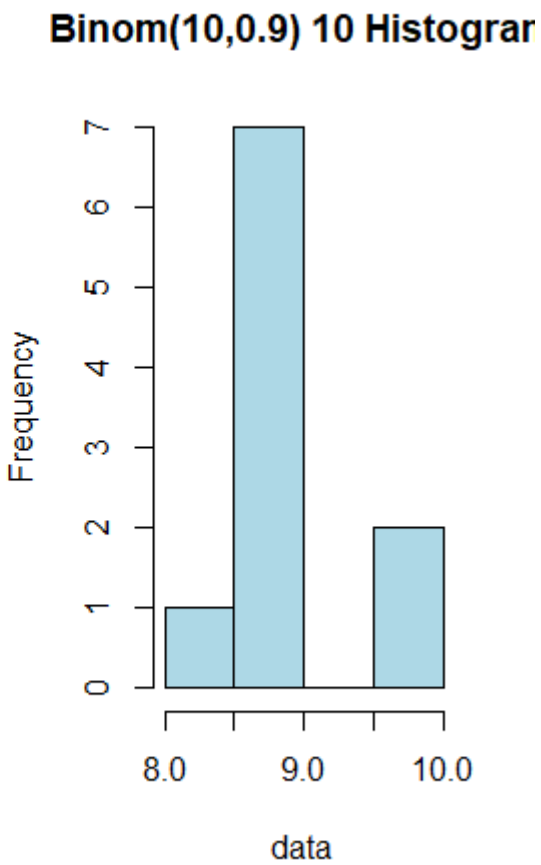

Binom(1,0.9) 10 Scatter Plc
data
Index
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10


Binom(1,0.9) 10 Histogran
Frequency
data
0
2
4
6
8
0.0
0.4
0.8

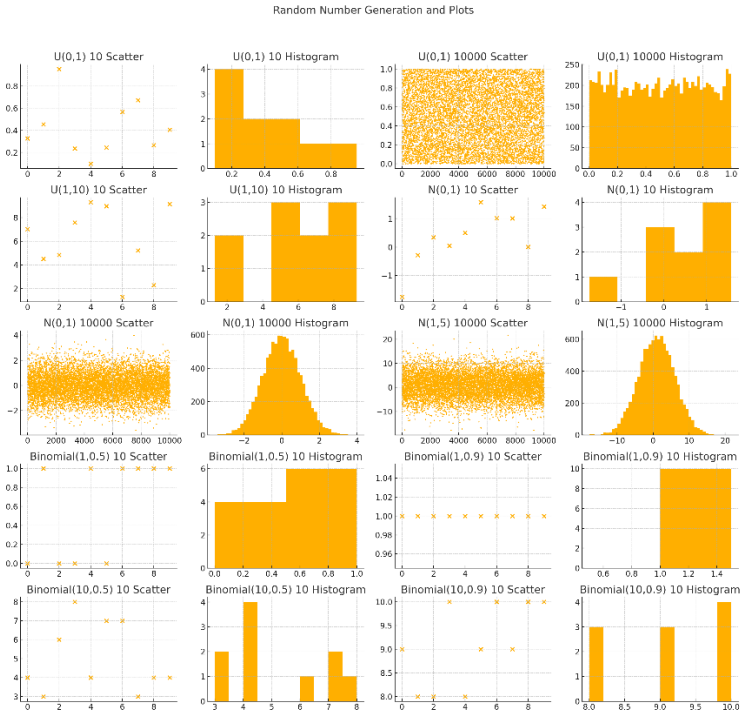

Binom(10,0.5) 10 Scatter Pl
data
Index
3.0
3.5
4.0
4.5
5.0
5.5
6.0
2
4
6
8
10


Binom(10,0.5) 10 Histograr
Frequency
data
0
1
2
3
4
3.0
4.0
5.0
6.0


Binom(10,0.9) 10 Scatter Pl
data
Index
8.0
8.5
9.0
9.5
10.0
2
4
6
8
10


Binom(10,0.9) 10 Histograr
Frequency
data
0
1
2
3
4
5
6
7
8.0
9.0
10.0

Python（可放大查看）：



3、在飞行棋游戏中，每名玩家有 4 架飞机棋子，飞机棋子起飞规则如下：（1）当玩家（抛骰子）抛 6 点的点数，飞机才能准备起飞。（2）当玩家没有（抛骰子）到 6 点，则飞机不能起飞直到抛到 6 点。 假设玩家 1 独自练习，规则为：只要抛出一次 6 点，就记一次准备起飞，反复抛一枚骰子 10 次，记 10 次中累计准备起飞次数为X。模拟玩家 1 练习过程 1000 次，画出X的直方图。（提示：产生多项分布随机数，计数，并重复）

R 语言代码：

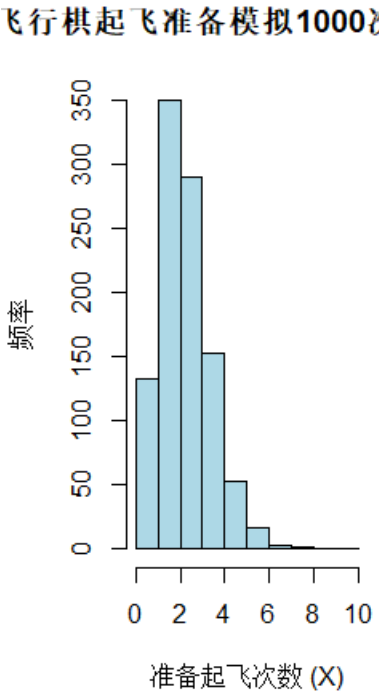
```
# 模拟玩家 1 抛骰子并统计准备起飞次数
simulate_game <- function() {
```

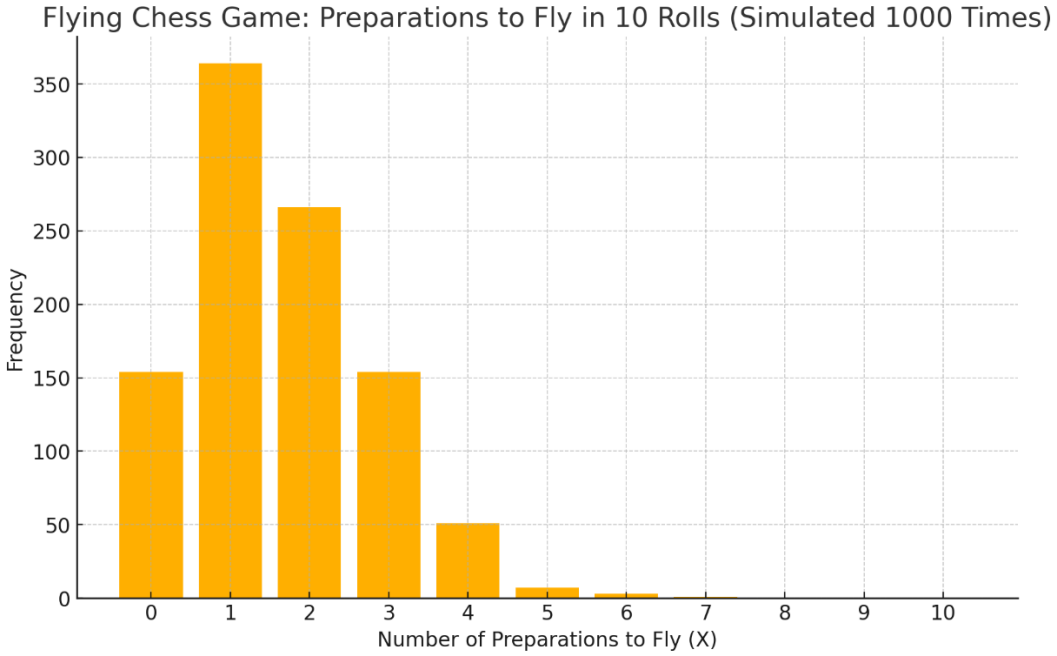
```
  return(sum(rbinom(10, 1, 1/6)))
}

# 进行 1000 次模拟
simulations <- replicate(1000, simulate_game())

# 画出直方图
hist(simulations, breaks=0:10, right=FALSE, col="lightblue",
     main="飞行棋起飞准备模拟 1000 次的结果", xlab="准备起飞次数 (X)",
ylab="频率")
```

Python 代码：

```
def simulate_game():
    return np.sum(np.random.binomial(1, 1/6, 10))

# Simulate the process 1000 times
simulations = [simulate_game() for _ in range(1000)]

# Plot the histogram of preparations to fly
plt.hist(simulations, bins=range(12), align='left', rwidth=0.8)
plt.title('Flying Chess Game: Preparations to Fly in 10 Rolls (Simulated
1000 Times)')
plt.xlabel('Number of Preparations to Fly (X)')
plt.ylabel('Frequency')
plt.xticks(range(11))
plt.show()
```

结果展示（上为 R 语言运行结果下为 Python 运行结果）：


飞行棋起飞准备模拟1000次的
频率
0
50
100
150
200
250
300
350
0
2
4
6
8
10
准备起飞次数 (X)


Flying Chess Game: Preparations to Fly in 10 Rolls (Simulated 1000 Times)
Frequency
0
50
100
150
200
250
300
350
0
1
2
3
4
5
6
7
8
9
10
Number of Preparations to Fly (X)