```r
library(reticulate)
use_python("/Users/che/pythonProject2/venv/bin/python")
```

```r
knitr::opts_chunk$set(
  echo = TRUE,
  fig.align = "center",
  fig.path = "fig/",
  fig.showtext = TRUE,
  message = FALSE,
  warning = FALSE
)
```

作业说明

仿照**二项分布**的例子,试给出**超几何分布**、**多项分布**和**多元超几何分布**的例子,列出参考出处(书本文献,或者网页),并给出相应例子的**R**和**Python**实现.

- 提交作业截止时间：2022-03-18-21:00 (周五)
- 打包发送QQ邮箱; 文件名和邮件名格式如: amsa-hw2-20191003699-renyifei.zip, 包含如:
  - amsa-hw2-20191003699-renyifei.md
  - amsa-hw2-20191003699-renyifei.rmd
  - amsa-hw2-20191003699-renyifei.pdf
  - amsa-hw2-20191003699-renyifei.html
- > 注意事项
  >
  > - **打包文件zip里面一定要有pdf文件, 便于批注和批改**
  >   - 可使用Typora编辑Markdown文件, 然后导出pdf, 再打包.md和.pdf文件
  >   - 亦可使用RStudio编辑R Markdown文件, 然后生成pdf, 再打包.rmd和.pdf文件
  >   - R Markdown转pdf有困难, 可导出html并设定`keep_md: yes`,再将md转pdf
  >   - 需要提供.rmd, .html, .md和.pdf, 将其打包上传

二项分布

如果随机变量有如下的概率分布:

$$P(X=k)=C_n^k p^k q^{n-k}, \quad k=0,1,\cdots,n$$

则称 X 服从二项分布, 其中 $pq>0, p+q=1$, 记做 $X\sim B(n,p)$.

称为二项分布的原因是 $C_n^k p^k q^{n-k}$ 为二项展开式

$$(p+q)^n=\sum_{k=0}^{n} C_n^k p^k q^{n-k}$$

的第 $k+1$ 项. B 是 Binomial的缩写.

例

说野外正在进行 $9(n=9)$ 口石油勘探井的发掘工作，每一口井能够开发出油的概率是 $0.1(p=0.1)$。请问，最终所有的勘探井都勘探失败的概率?

我们手动用公式计算得:

$$\binom{n}{k}p^k(1-p)^{n-k}=\binom{9}{0}0.1^0(1-0.1)^9=0.1^9\approx0.3874$$

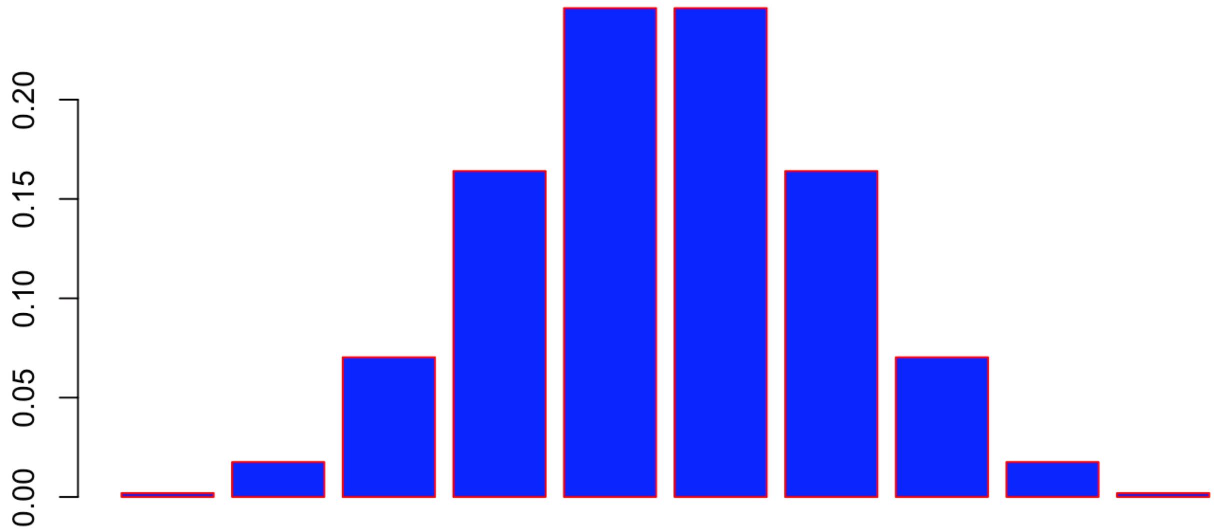
R实现

```
n = 9; p = 0.1; k = 0
dbinom(x=k,size=n,prob=p)
```

```
##[1] 0.3874205
```

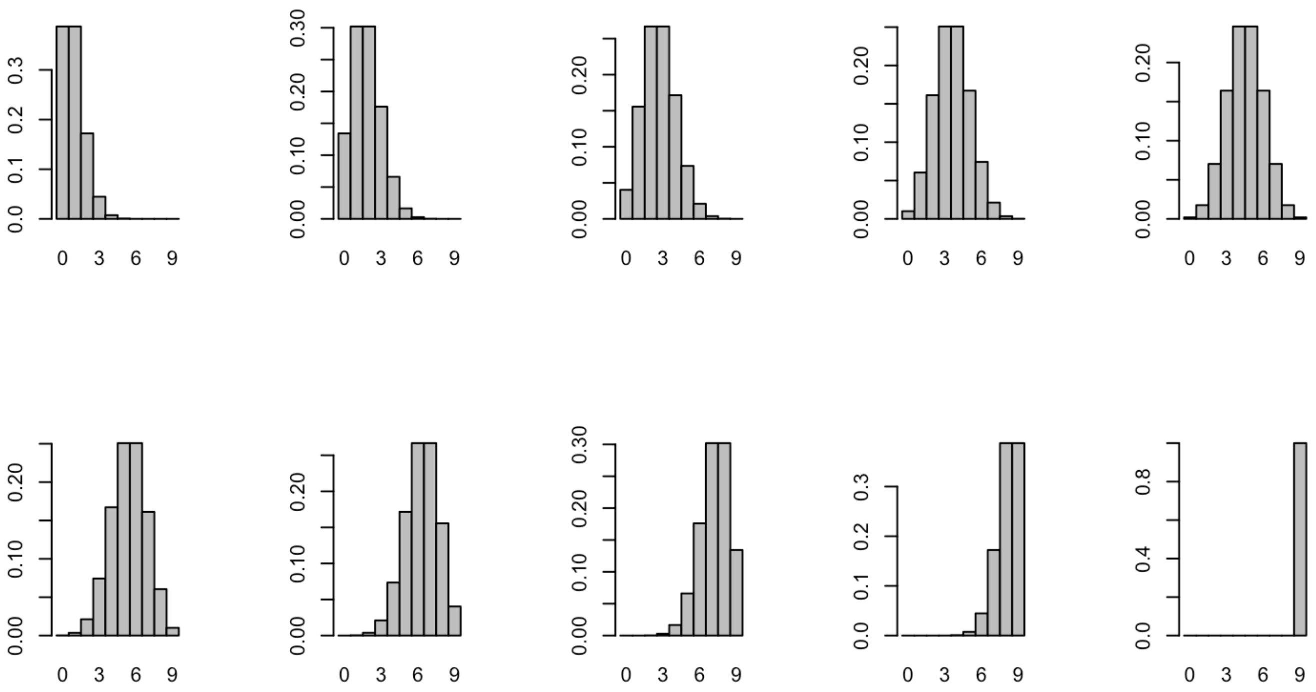
画图

```
### https://blog.csdn.net/zfy0910/article/details/120631214
x = seq(0,9,by = 1) # 创建一个0到9的数组间隔为1
y = dbinom(x,9,0.1) #分别求出实现实验成功从0到50次的概率
plot(x,y) #画出二项分布(p=0.1)的图像
z = dbinom(x,9,0.5) #p=0.5; 概率越接近0.5，图像越对称
plot(x,z)
barplot(z,col="blue",border="red") #实现柱状图
```



多图表示

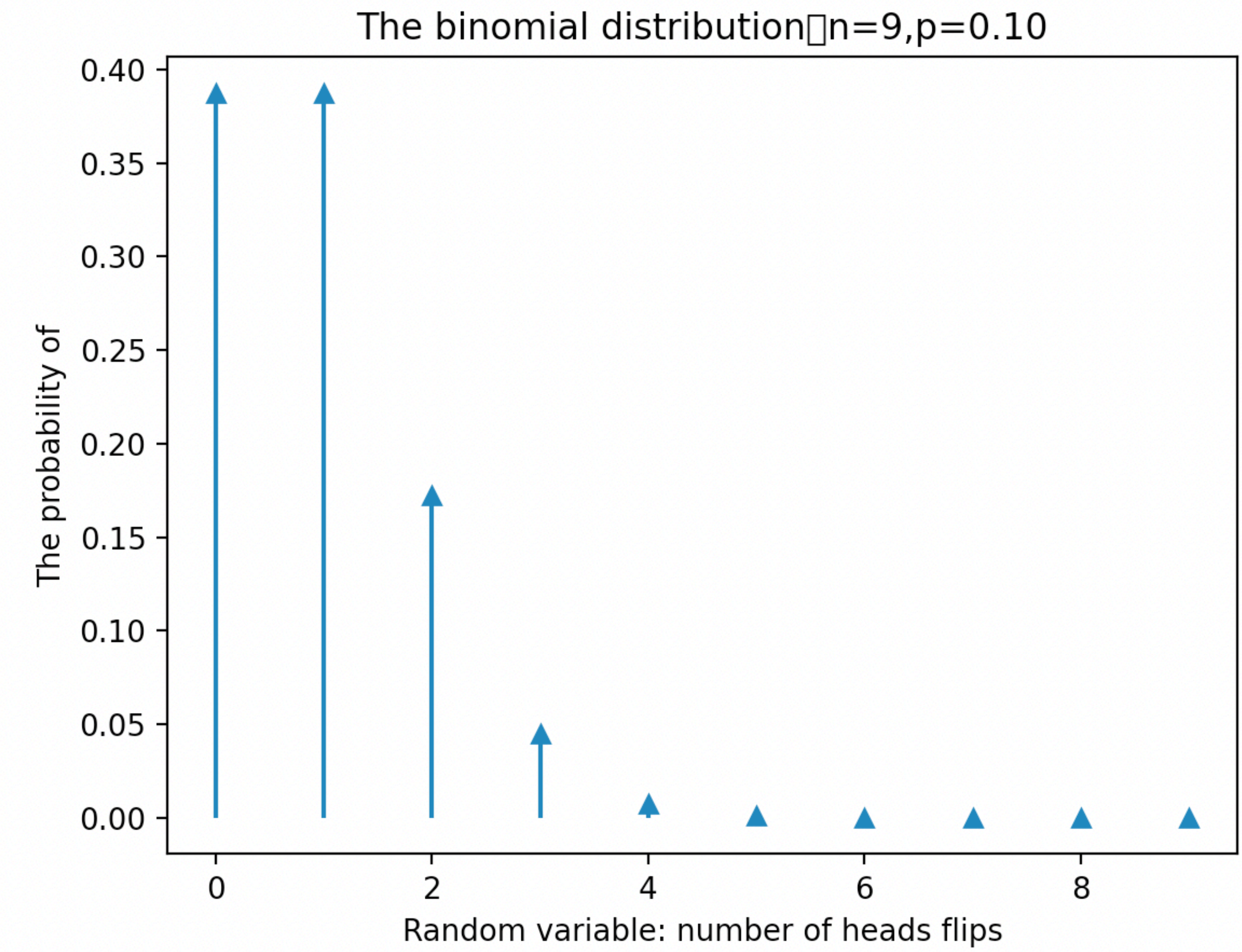
```
### https://oomake.com/question/5068173
## 方法1:mapply函数
graph <- function(n,p){
  x <- dbinom(0:n,size=n,prob=p)
  barplot(x,names.arg=0:n,
         main=sprintf(paste('bin. dist. ',n,p,sep=':')))
}
op = par(mfcol=c(2,5))
mapply(graph,9,seq(0.1,1,0.1))
par(op)
## 方法2:for循环
par(mfrow=c(2, 5))
for(p in seq(0.1, 1, len=10))
{
    x <- dbinom(0:9, size=9, p=p)
    barplot(x, names.arg=0:9, space=0)
}
```



Python实现

概率精确值

```
### https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.binom.html
# from scipy import stats
import scipy.stats as stats #导入统计计算包的统计模块
import numpy as np #数组包
import matplotlib.pyplot as plt #绘图包
### arange用于生成一个等差数组，arange(start, stop, step)
n = 9    # 做某件事情的次数
p = 0.1 # 做某件事情成功的概率
X = np.arange(0, n+1, 1)
X
### 求对应分布的概率:概率质量函数 (PMF)
### 它返回一个列表，列表中每个元素表示随机变量中对应值的概率
pList = stats.binom.pmf(X, n, p)
pList[0] # X=0(即所有勘探失败)的概率
### 绘图
plt.rcParams['font.sans-serif']=['SimHei'] #python 绘图中文显示
plt.plot(X, pList, marker='^',linestyle='None')
plt.vlines(X, 0, pList)
plt.xlabel('随机变量：抛硬币正面朝上次数') #x轴文本
plt.ylabel('概率') #y轴文本
plt.title('二项分布：n=%i,p=%.2f' % (n,p)) #标题
plt.show() #显示图形
### 更多绘图
# https://blog.csdn.net/weixin_42695959/article/details/84037126
```



随机数模拟值

```
import numpy as np
n, p = 9, 0.1
a = sum((np.random.binomial(n, p, size=30000)==0))/30000
print(a)
```

```
##0.3853666666666667
```

参考

- 何书元. 概率论[M]. 北京: 北京大学出版社, 2006.
- 二项分布和多项分布
- 概率分布的介绍和使用
- Python统计分析(3)——几种重要的概率分布：二项式分布

超几何分布

超几何分布是统计学上一种离散概率分布。它描述了从有限N个物件（其中包含M个指定种类的物件）中抽出n个物件，成功抽出该指定种类的物件的次数（不放回）。称为超几何分布，是因为其形式与“超几何函数”的级数展式的系数有关。
超几何分布中的参数是M,N,n，上述超几何分布记作$X \sim H(n, M, N)$。

概率一般记作：$\frac{C_m^a C_n^b}{C_{n+m}^{a+b}}$

例

在一个口袋中装有30个球，其中有10个红球，其余为白球，这些球除颜色外完全相同。游戏者一次从中摸出5个球。恰好摸到4个红球就中一等奖，那么获一等奖的概率是多少?

我们手动用公式计算得：$\frac{C_{10}^4 C_{20}^1}{C_{30}^5} = 0.029$

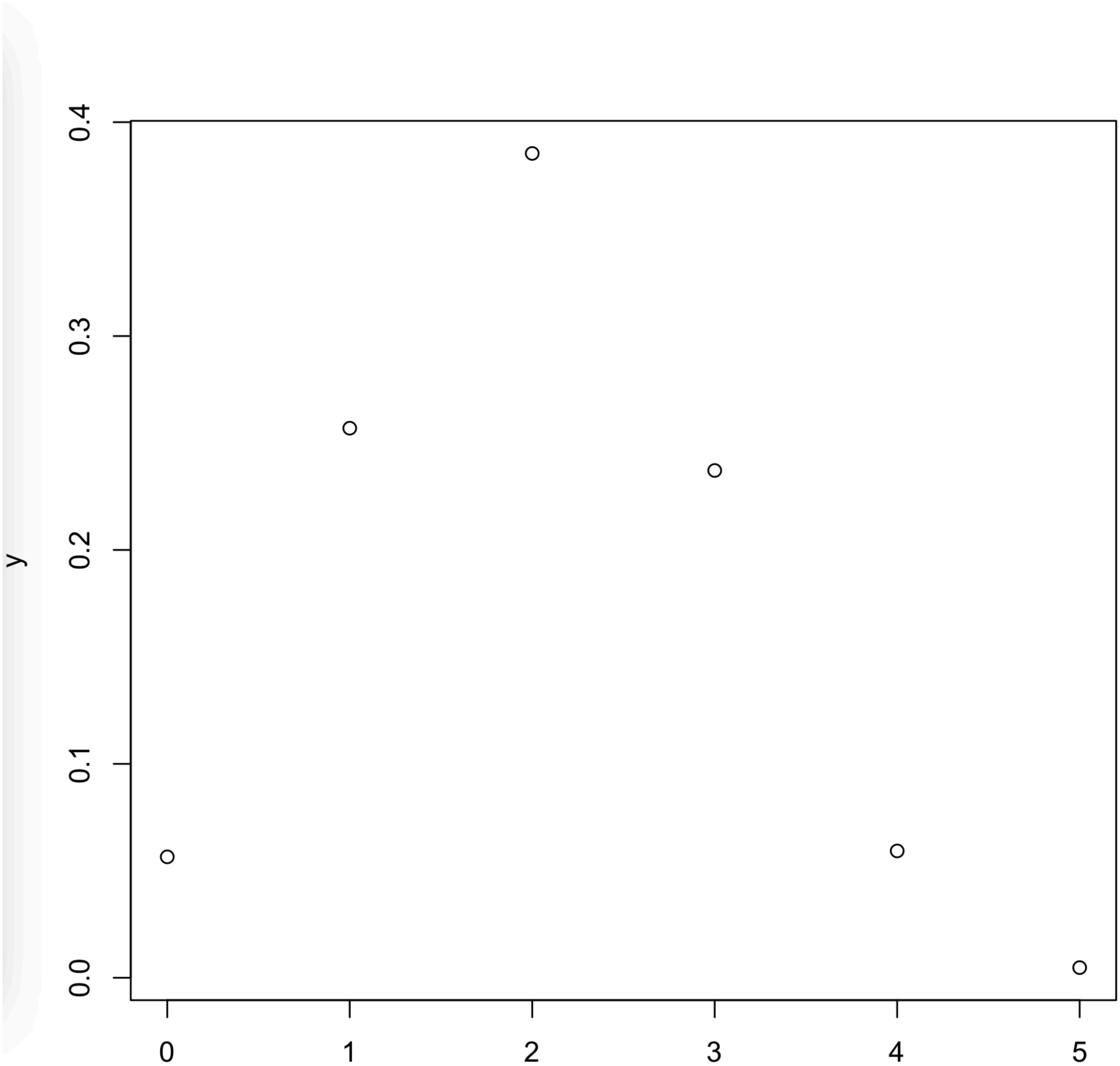
R实现

```
n<-10          #有十个红球
m<-30-n        #白球的数量
t<-5           #摸球数量
dhyper(4, n, m, t)
```

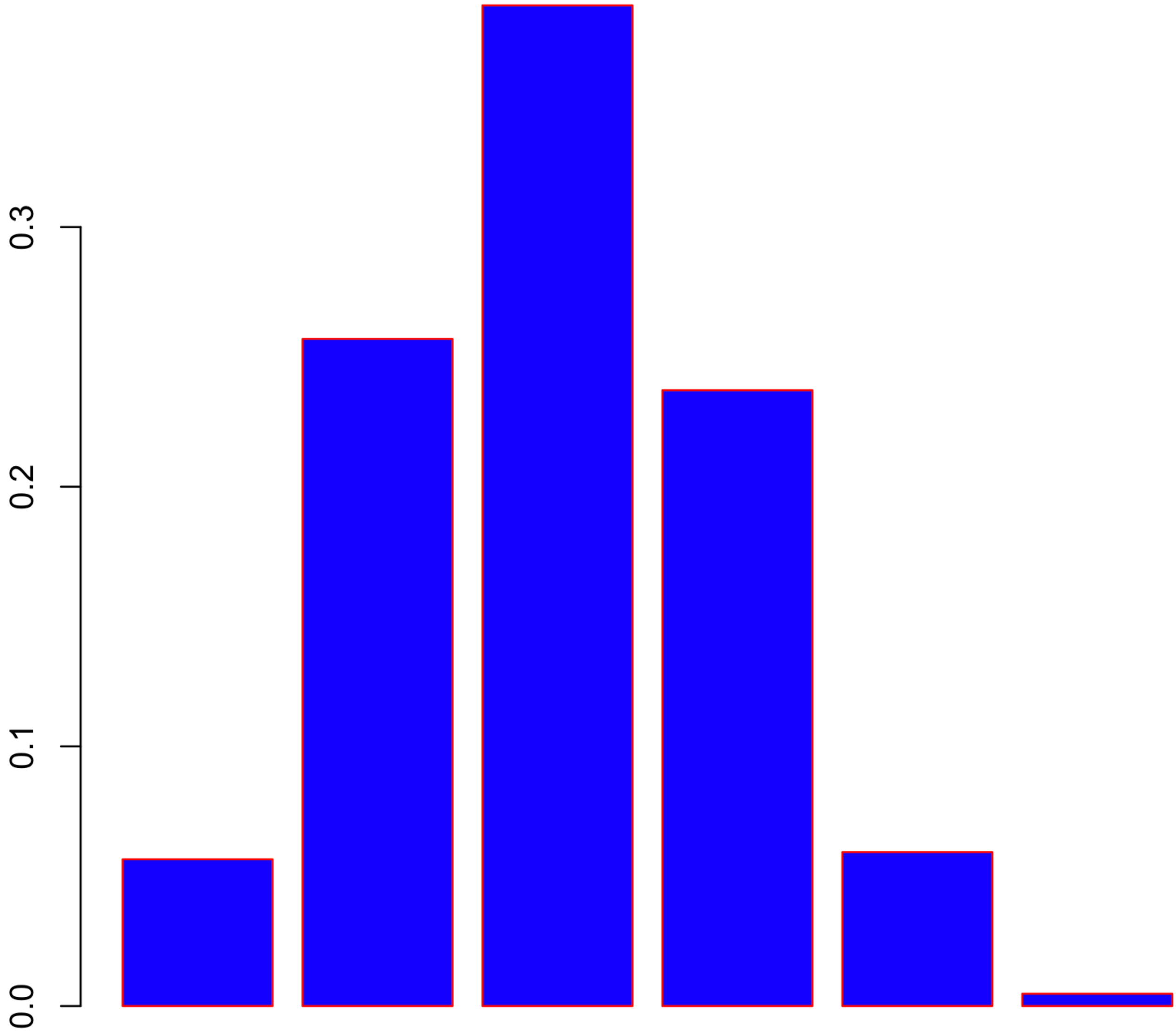
```
##[1] 0.02947244
```

画图

```
x = seq(0,5,by = 1) # 创建一个 0 到 5 的数组间隔为 1
y = dhyper(x,5,20,10) # 分别求出实现实验成功从 0 到 5 次的概率
plot(x,y) # 画出超几何分布的图像
```

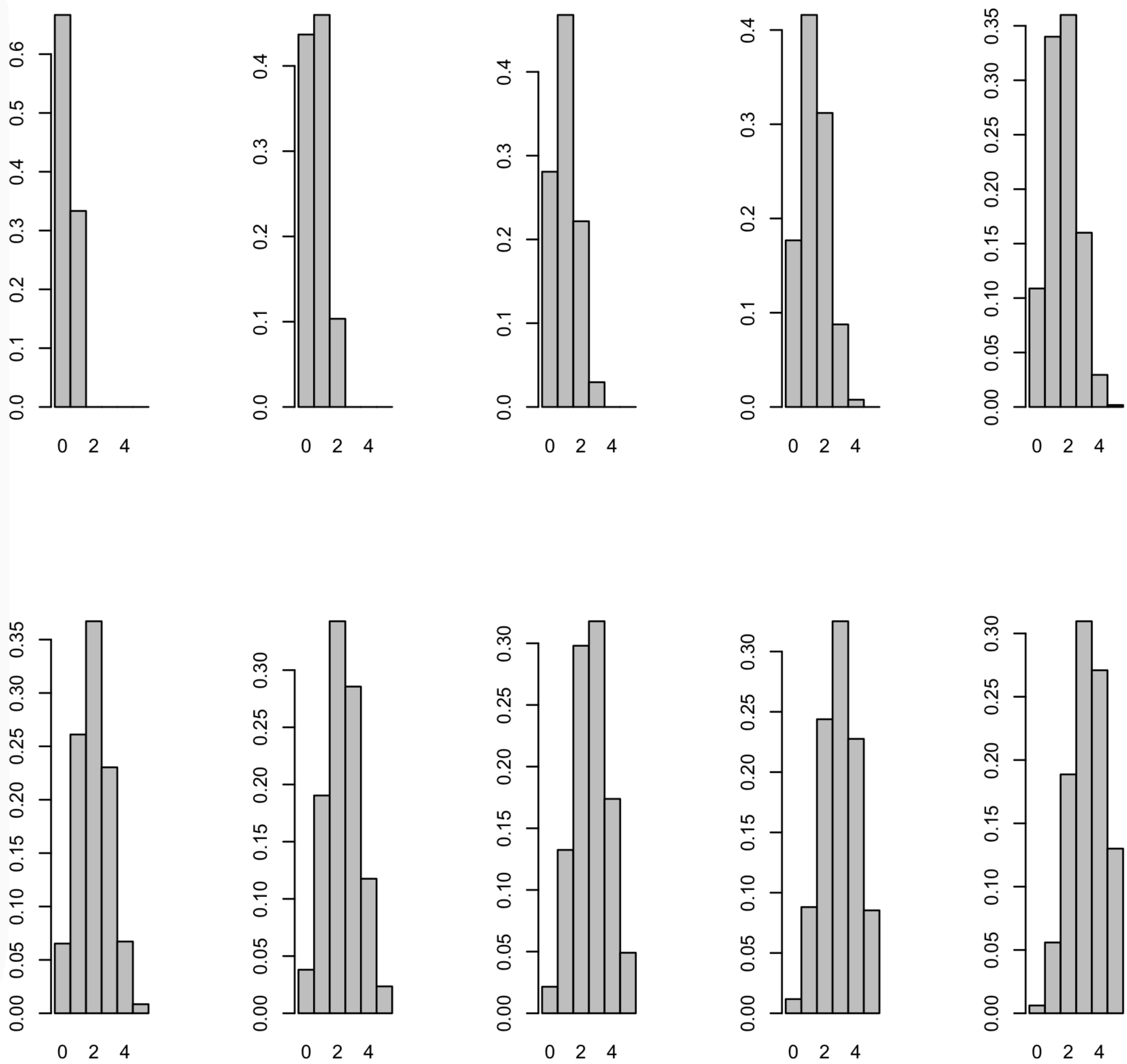



```
barplot(y,col="blue",border="red") # 柱状图
```



多图表示：

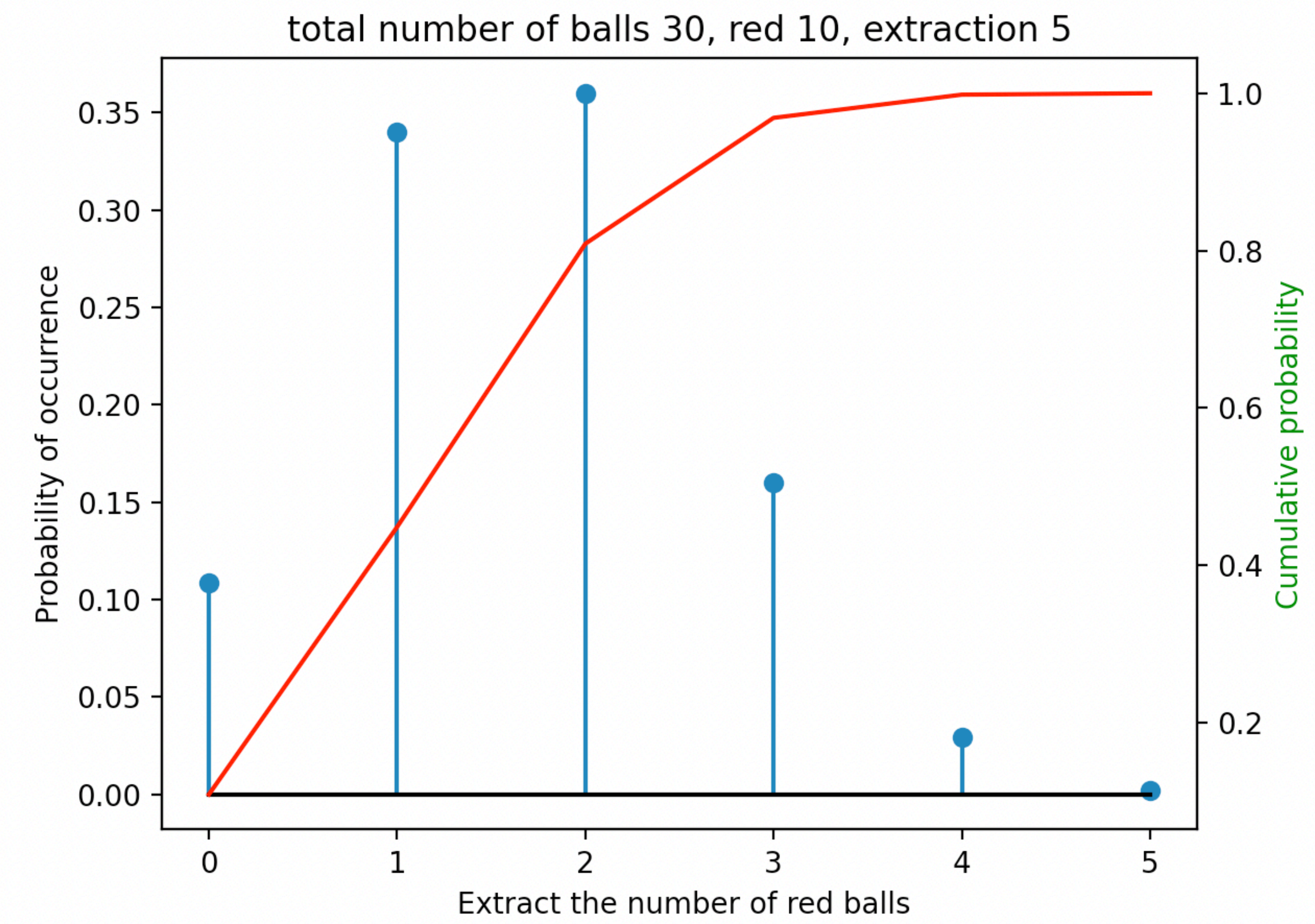
```
par(mfrow=c(2, 5))
for(p in seq(1,10,by=1))
{x <- dhyper(0:5,p,30-p,10)
barplot(x, names.arg=0:5,space=0)}
```



Python实现

```
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt
N = 30
M = 10
n = 5
po=stats.hypergeom(N,M,n)
x=np.arange(n+1)
px=po.pmf(x)
fig = plt.figure()
ax = plt.gca()
line1 = ax.stem(x,px,basefmt='k',label='The probability of picking a red ball')
ax.set_xlabel(' Extract the number of red balls')
```

```
ax.set_ylabel('Probability of occurrence')
ax.set_title('total number of balls 30, red 10, extraction 5')
ax2=plt.twinx()
y=po.cdf(x)
line2 = ax2.plot(x,y,'r',label='Cumulative probability')
ax2.set_ylabel('Cumulative probability',color='g')
plt.show()
```



随机数模拟值

```
import numpy as np
n, p = 9, 0.1
s = np.random.hypergeometric(10,20,5,size=1000000)
print(s)
```

```
##[3 2 0 ... 2 1 1]
```

参考

- 盛骤, 谢式千, 潘承毅. 概率论与数理统计[M]. 第4版. 北京: 高等教育出版社, 2008.
- 王学民. 应用多元统计分析[M]. 第5版. 上海: 上海财经大学出版社, 2017.

-http://www.py.cn/jishu/jichu/13376.html

多项分布

如果一个随机向量$X=(X_1,X_2,...,X_N)$满足下列条件：
$(1)X_i\geq(1\leq i\leq n)$,且$X_1+X_2+...+X_n=N$;
(2)设$m_1,m_2,...,m_n$为任意非负整数，且$m_1+m_2+...+m_n=N$，则事件
$X_1=m_1;X_2=m_2,...,X_n=m_n$的概率为
$P\{X_1=m_1,X_2=m_2,...,X_n=m_n\}=\frac{N!}{m_1!m_2!...m_n!}{p_1}^{m_1}{p_2}^{m_2}...{p_n}^{m_n}$
其中，$p_i\geq0(1\leq i\leq n),p_1+p_2+...+p_n=1$则称随机向量$X=(X_1,X_2,...,X_n)$服从多项分布，记作$X\text{~}PN(N:p_1,p_2,...,p_n)$。

例

西班牙足球甲级联赛(简称"西甲")已经成为世界上最高水平的足球联赛之一，吸引了世界球迷的目光，巴萨与皇马的德比之战更是令世界瞩目。截止目前，巴萨在历史上和皇马交手178次，拿到了71胜35平72负的战绩。如果接下来巴萨与皇马将进行6场比赛，求巴萨3胜1平2负的概率是多少?

我们手动用公式计算得：
$P\{X_1=m_1,X_2=m_2,...,X_n=m_n\}=\frac{N!}{m_1!m_2!...m_n!}{p_1}^{m_1}{p_2}^{m_2}...{p_n}^{m_n}$
$P\{X_1=\frac{71}{178},X_2=\frac{35}{178},X_3=\frac{72}{178}\}=\frac{6!}{3!1!2!}{p_1}^{m_1}{p_2}^{m_2}{p_3}^{m_3}$

其中：$p_1=\frac{71}{178},p_2=\frac{35}{178},p_3=\frac{72}{178},m_1=3,m_2=1,m_3=2$
得到概率为：0.1225009

R实现

```
multi <- function(p, x) {
  n <- sum(x)    ##试验次数

  f1 <- prod(p^x)
  f2 <- prod(factorial(x))
  factorial(n)*f1/f2       ##返回值
}
p <- c(71, 35, 72)/178
x <- c(3, 1, 2)
multi(p, x)
```

```
##[1] 0.1225009
```

R模拟：

```r
a <- c('胜', '平', '负')
p <- c(71, 35, 72)/178  ##巴萨胜、平、负各概率值
n <- 10000  ##比赛场数

set.seed(18)
result <- sample(a, n, replace=TRUE, prob=p)  ##replace=TRUE表示可重复抽样

head(result, 20)  ##前20场比赛的结果
table(result)  ## 计数
result
n*p[1]  ##巴萨获胜场数的理论值
n*p[2]  ##巴萨打平场数的理论值
n*p[3]  ##巴萨输皇马场数的理论值
```

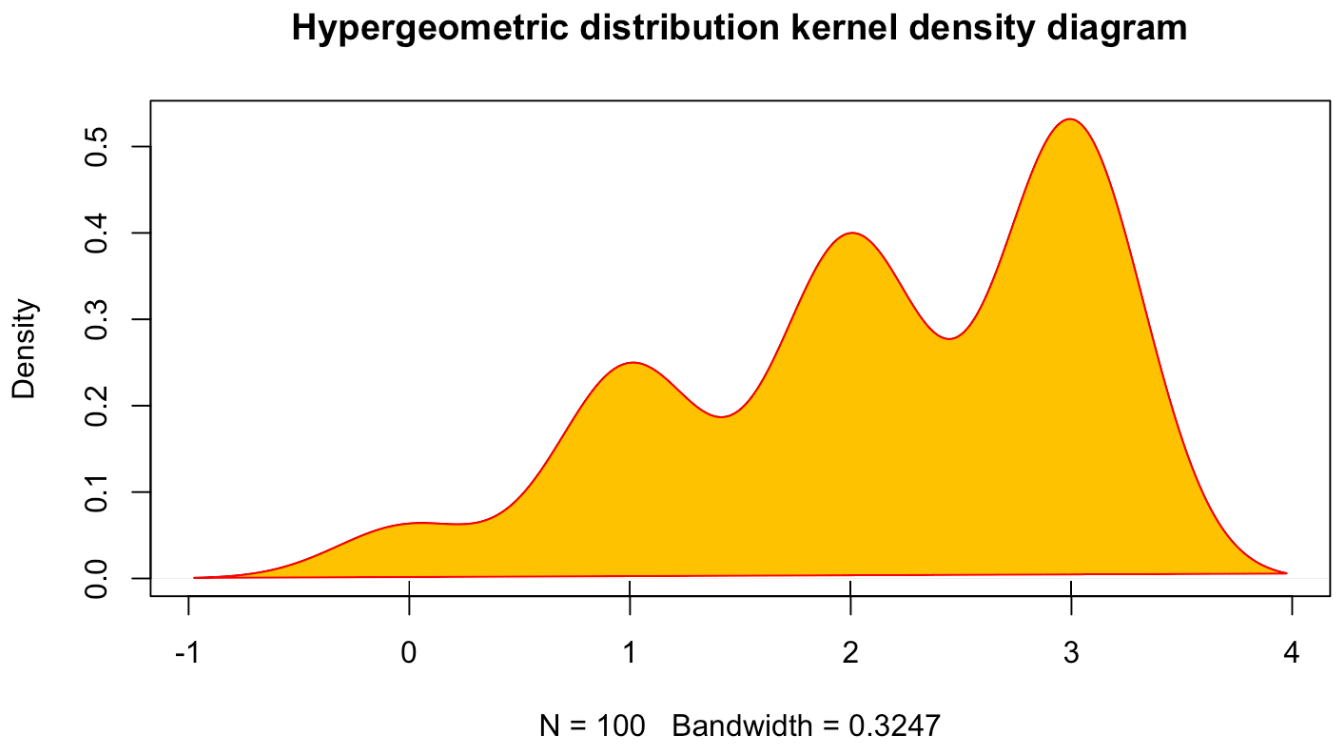
```
> a <- c('胜', '平', '负')
> p <- c(71, 35, 72)/178  ##巴萨胜、平、负各概率值
> n <- 10000  ##比赛场数
>
> set.seed(18)
> result <- sample(a, n, replace=TRUE, prob=p)  ##replace=TRUE表示可重复抽样
>
> head(result, 20)  ##前20场比赛的结果
 [1] "平" "胜" "平" "负" "负" "胜" "负" "胜" "负" "负" "胜" "胜" "负" "负" "平" "负"
[17] "胜" "负" "平" "负"
> table(result)  ## 计数
result
  负   平   胜
4022 1958 4020
> result
    [1] "平" "胜" "平" "负" "负" "胜" "负" "胜" "负" "负" "胜" "胜" "负" "负" "平" "负"
   [17] "胜" "负" "平" "负" "负" "负" "平" "平" "负" "胜" "胜" "胜" "负" "负" "平" "胜"
   [33] "负" "负" "负" "胜" "平" "胜" "胜" "胜" "负" "负" "平" "负" "负" "胜" "负" "负"
   [49] "胜" "胜" "负" "负" "负" "胜" "胜" "胜" "负" "负" "负" "负" "负" "胜" "胜" "平"
   [65] "胜" "胜" "平" "负" "负" "平" "平" "负" "胜" "胜" "负" "负" "胜" "胜" "胜" "胜"
   [81] "胜" "胜" "平" "胜" "胜" "平" "平" "负" "胜" "负" "胜" "胜" "胜" "负" "负" "胜"
   [97] "负" "胜" "负" "胜" "负" "负" "负" "平" "胜" "平" "负" "负" "胜" "负" "负" "负"
  [113] "负" "胜" "平" "平" "胜" "胜" "胜" "胜" "负" "胜" "胜" "负" "负" "负" "胜" "负"
  [129] "负" "负" "胜" "胜" "负" "胜" "胜" "平" "负" "胜" "胜" "胜" "负" "负" "胜" "负"
  [145] "负" "平" "负" "负" "负" "负" "胜" "平" "胜" "平" "负" "负" "胜" "负" "胜" "负"
  [161] "负" "负" "负" "胜" "负" "胜" "平" "胜" "胜" "平" "负" "胜" "负" "平" "负" "负"
  [177] "胜" "负" "胜" "胜" "胜" "负" "负" "平" "负" "胜" "负" "胜" "胜" "平" "负" "胜"
  [193] "平" "负" "负" "平" "负" "胜" "平" "平" "胜" "平" "胜" "平" "胜" "胜" "平" "胜"
  [209] "负" "胜" "平" "平" "胜" "胜" "胜" "胜" "胜" "负" "胜" "负" "胜" "胜" "负" "胜"
  [225] "负" "负" "胜" "平" "平" "负" "胜" "负" "平" "负" "负" "负" "负" "负" "胜" "胜"
  [241] "负" "胜" "胜" "胜" "胜" "胜" "胜" "胜" "胜" "平" "负" "平" "胜" "负" "胜" "负"
  [257] "负" "负" "胜" "负" "负" "负" "负" "胜" "负" "胜" "胜" "胜" "负" "胜" "胜" "负"
  [273] "平" "平" "负" "负" "平" "胜" "负" "平" "胜" "平" "胜" "胜" "胜" "负" "平" "负"
  [289] "负" "负" "胜" "负" "负" "胜" "胜" "胜" "负" "负" "平" "负" "平" "负" "负" "胜"
```

```
 [305] "胜" "胜" "平" "负" "负" "负" "负" "平" "胜" "胜" "负" "胜" "平" "胜" "负" "平"
 [321] "负" "负" "胜" "胜" "平" "负" "负" "胜" "胜" "负" "平" "负" "胜" "胜" "胜" "平"
 [337] "平" "负" "胜" "负" "负" "胜" "胜" "胜" "平" "负" "负" "胜" "胜" "平" "胜" "胜"
 [353] "负" "胜" "胜" "胜" "负" "胜" "平" "负" "负" "胜" "负" "胜" "胜" "负" "负" "胜"
 [369] "胜" "负" "平" "胜" "负" "胜" "胜" "胜" "负" "负" "负" "负" "胜" "负" "负" "负"
 [385] "平" "胜" "平" "负" "胜" "胜" "负" "胜" "负" "胜" "平" "负" "胜" "胜" "平" "负"
 [401] "胜" "平" "胜" "胜" "负" "平" "胜" "胜" "负" "平" "胜" "平" "胜" "负" "负" "平"
 [417] "负" "胜" "负" "胜" "胜" "负" "胜" "负" "胜" "负" "胜" "负" "平" "平" "负" "负"
 [433] "负" "负" "平" "胜" "胜" "胜" "负" "胜" "负" "负" "负" "胜" "胜" "胜" "负" "负"
 [449] "负" "胜" "负" "负" "平" "负" "负" "负" "负" "平" "负" "胜" "胜" "胜" "负" "胜"
 [465] "胜" "胜" "负" "负" "胜" "平" "负" "胜" "平" "胜" "胜" "胜" "平" "胜" "胜" "平"
 [481] "胜" "胜" "胜" "胜" "负" "负" "平" "胜" "胜" "胜" "负" "胜" "负" "胜" "平" "负"
 [497] "平" "平" "胜" "胜" "胜" "平" "平" "负" "平" "平" "负" "胜" "负" "负" "胜" "胜"
 [513] "负" "负" "负" "胜" "平" "胜" "负" "胜" "胜" "负" "胜" "胜" "胜" "平" "胜" "负"
 [529] "负" "负" "胜" "负" "胜" "胜" "负" "胜" "负" "负" "负" "负" "平" "负" "平" "平"
 [545] "负" "胜" "负" "负" "胜" "负" "胜" "胜" "胜" "负" "平" "胜" "胜" "负" "胜" "平"
 [561] "负" "平" "负" "负" "负" "平" "胜" "负" "胜" "平" "负" "胜" "负" "胜" "负" "负"
 [577] "胜" "负" "平" "负" "平" "胜" "负" "负" "胜" "平" "平" "平" "负" "胜" "负" "平"
 [593] "负" "负" "负" "胜" "负" "胜" "胜" "负" "平" "平" "胜" "胜" "负" "负" "胜" "负"
 [609] "平" "平" "胜" "负" "负" "胜" "平" "胜" "胜" "负" "平" "负" "胜" "胜" "胜" "负"
 [625] "胜" "胜" "负" "胜" "负" "胜" "平" "胜" "胜" "平" "胜" "平" "负" "胜" "胜" "平"
 [641] "胜" "胜" "负" "胜" "胜" "胜" "平" "负" "负" "负" "负" "平" "负" "胜" "平" "胜"
 [657] "平" "胜" "胜" "负" "胜" "负" "胜" "胜" "胜" "平" "胜" "胜" "平" "负" "负" "负"
 [673] "负" "负" "胜" "胜" "胜" "负" "胜" "胜" "负" "平" "负" "平" "胜" "平" "平" "胜"
 [689] "平" "胜" "平" "胜" "胜" "负" "胜" "胜" "胜" "胜" "胜" "胜" "胜" "平" "负" "平"
 [705] "胜" "胜" "胜" "负" "胜" "平" "平" "负" "负" "胜" "负" "胜" "胜" "平" "胜" "负"
 [721] "负" "胜" "负" "胜" "负" "平" "胜" "胜" "负" "胜" "负" "胜" "平" "负" "负" "胜"
 [737] "平" "负" "胜" "胜" "负" "负" "胜" "胜" "负" "负" "负" "胜" "负" "平" "胜" "平"
 [753] "胜" "胜" "胜" "胜" "负" "平" "平" "胜" "胜" "胜" "平" "胜" "胜" "负" "负" "胜"
 [769] "平" "胜" "负" "胜" "胜" "平" "负" "胜" "负" "负" "负" "胜" "胜" "负" "负" "负"
 [785] "负" "平" "胜" "负" "胜" "平" "负" "负" "负" "负" "胜" "负" "负" "平" "平" "负"
 [801] "负" "平" "负" "负" "负" "胜" "负" "负" "胜" "胜" "负" "胜" "胜" "负" "负" "负"
 [817] "负" "胜" "胜" "负" "负" "负" "负" "胜" "负" "平" "胜" "负" "负" "负" "负" "胜"
 [833] "平" "胜" "负" "负" "平" "负" "胜" "负" "负" "胜" "负" "胜" "平" "胜" "负" "负"
 [849] "负" "平" "胜" "平" "负" "负" "负" "负" "胜" "胜" "平" "胜" "平" "胜" "胜" "平"
 [865] "胜" "负" "平" "负" "平" "负" "胜" "平" "胜" "胜" "负" "胜" "胜" "平" "胜" "胜"
 [881] "胜" "负" "负" "负" "负" "负" "平" "平" "平" "胜" "平" "胜" "平" "负" "胜" "负"
 [897] "负" "胜" "负" "负" "平" "胜" "负" "负" "负" "负" "负" "平" "胜" "平" "负" "负"
 [913] "平" "胜" "平" "胜" "负" "平" "负" "平" "平" "负" "负" "负" "负" "胜" "负" "负"
 [929] "负" "胜" "胜" "负" "负" "胜" "平" "胜" "负" "负" "负" "胜" "平" "胜" "胜" "胜"
 [945] "平" "胜" "负" "胜" "负" "平" "负" "负" "胜" "负" "平" "平" "胜" "负" "负" "胜"
 [961] "负" "胜" "负" "平" "胜" "胜" "负" "负" "平" "胜" "负" "胜" "胜" "胜" "负" "胜"
 [977] "平" "平" "胜" "平" "胜" "平" "胜" "胜" "负" "平" "胜" "负" "负" "负" "平" "胜"
 [993] "平" "负" "胜" "胜" "负" "胜" "平" "平"
 [ reached getOption("max.print") -- omitted 9000 entries ]
> n*p[1]  ##巴萨获胜场数的理论值
[1] 3988.764
> n*p[2]  ##巴萨打平场数的理论值
[1] 1966.292
```

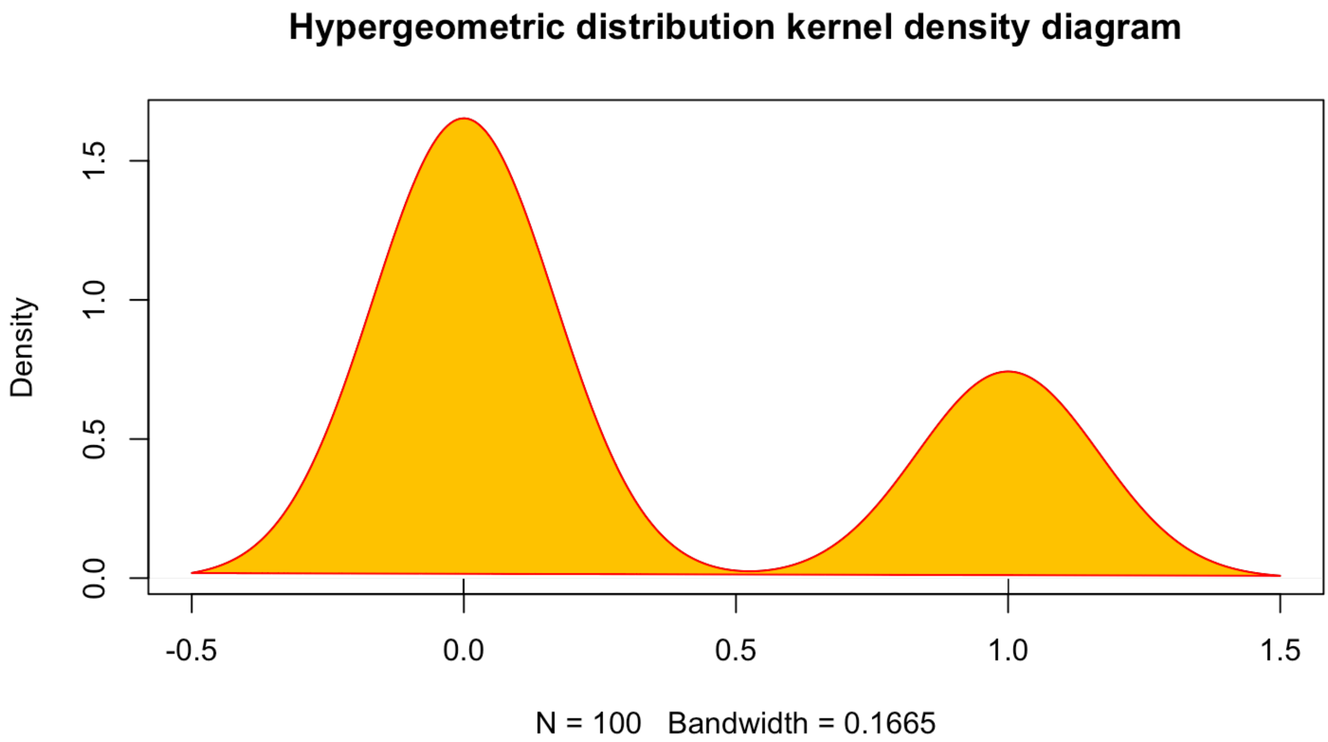
```
> n*p[3]   ##巴萨输皇马场数的理论值
[1] 4044.944
```

画图

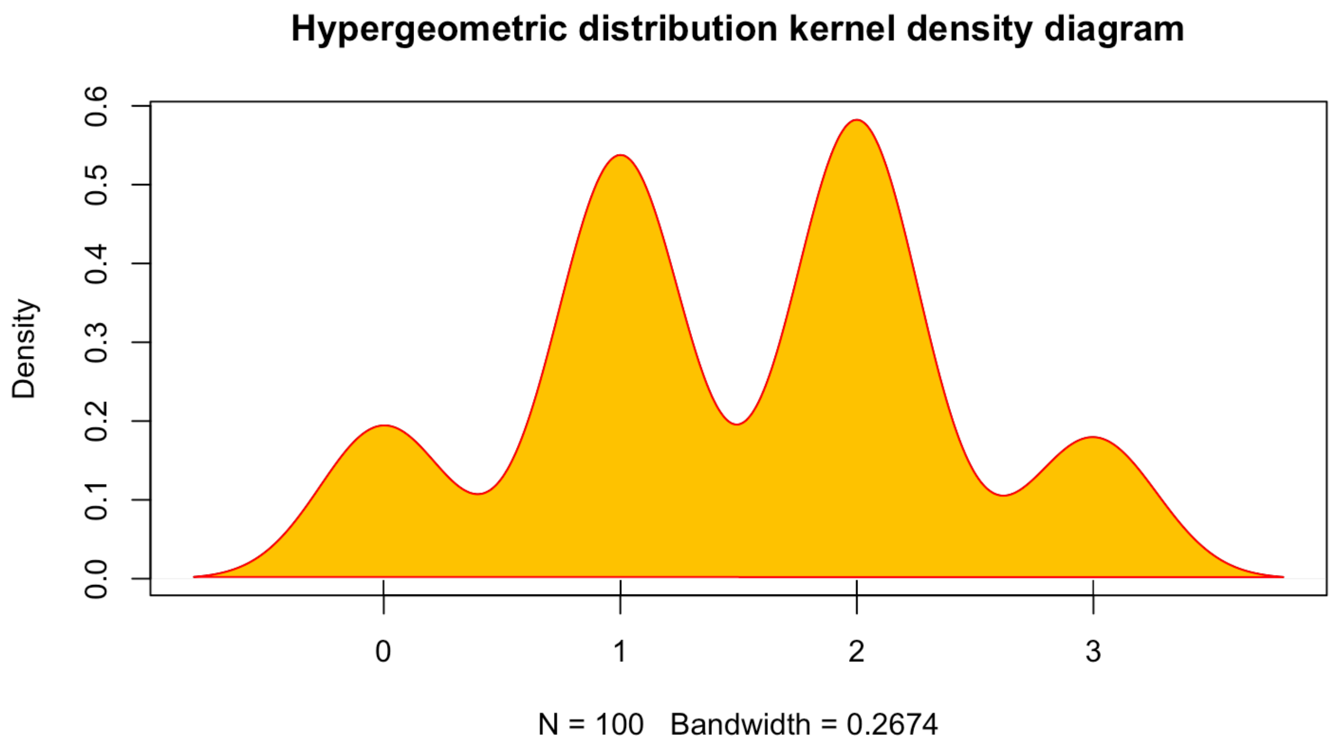
```
x1<-round(rhyper(nn=100,m=71,n=35,k=3));x1
plot(density(x1),main = "Hypergeometric distribution kernel density diagram")
polygon(density(x1),col=7,border = "red")
rug(x1,col="black")
```




```
x2<-round(rhyper(nn=100,m=35,n=72,k=1));x2
plot(density(x2),main = "Hypergeometric distribution kernel density diagram")
polygon(density(x2),col=7,border = "red")
rug(x2,col="black")
```




```
x3<-round(rhyper(nn=100,m=71,n=72,k=3));x3
plot(density(x3),main = "Hypergeometric distribution kernel density diagram")
polygon(density(x3),col=7,border = "red")
rug(x3,col="black")
```



Python实现

```
from functools import reduce
a = reduce(lambda x,y:x*y,range(1,6+1))
b = reduce(lambda x,y:x*y,range(1,3+1))
c = reduce(lambda x,y:x*y,range(1,1+1))
d = reduce(lambda x,y:x*y,range(1,2+1))
p = ((a)/(b*c*d))*((71/178)**3)*(35/178)*((72/178)**2)
print(p)
```

```
##0.1225009114171614
```

python模拟

```
import numpy as np
from numpy import random
for i in range(1):
    a = np.random.multinomial(10000, [71/178,35/178,72/178], size=1)
    print(a)
```

```
##[[3961 1942 4097]]
```

参考

- -王学民. 应用多元统计分析[M]. 第5版. 上海: 上海财经大学出版社, 2017. -https://zhuanlan.zhihu.com/p/52481385

多元超几何分布

多元超几何分布亦称阶乘多项分布。超几何分布的多元推广。
有一批产品共N件，其中一等品M_1件，二等品M_2件，从此批产品中任意取n件，求恰取出k_1件一等品，求k_2件二等品的概率。
设抽出的n件产品，有一等品的X件，二等品Y件，从N件产品中取出n件，共有取法C_N^n种。$(X=k_1,Y=k_2)$这件事相当于M_1件一等品中取k_1件，从M_2件二等品中取k_2，再从余$N-M_1-M_2$件产品中取出$n-k_1-k_2$件。
由乘法原理得：$P(X=k_1,Y=k_2)\frac{C_{M_1}^{k_1}C_{M_2}^{k_2}C_{N-M_1-M_2}^{n-k_1-k_2}}{C_N^n}$
$k_1=0,1,2,\ldots min\{n,M1\}$,$k_2=0,1,2,\ldots min\{n,M_2\}$

例

有一批产品100件，其中一等品70件，二等品30件，从此批产品中任意取10件，求恰好取出7件一等品，3件二等品的概率?
我们手动用公式计算得：
设：有一等品的X件，二等品Y件。
$P(X=7,Y=3)\frac{C_{70}^{7}C_{30}^{3}C_{100-70-30}^{10-7-3}}{C_{100}^{10}}=0.2811633943$

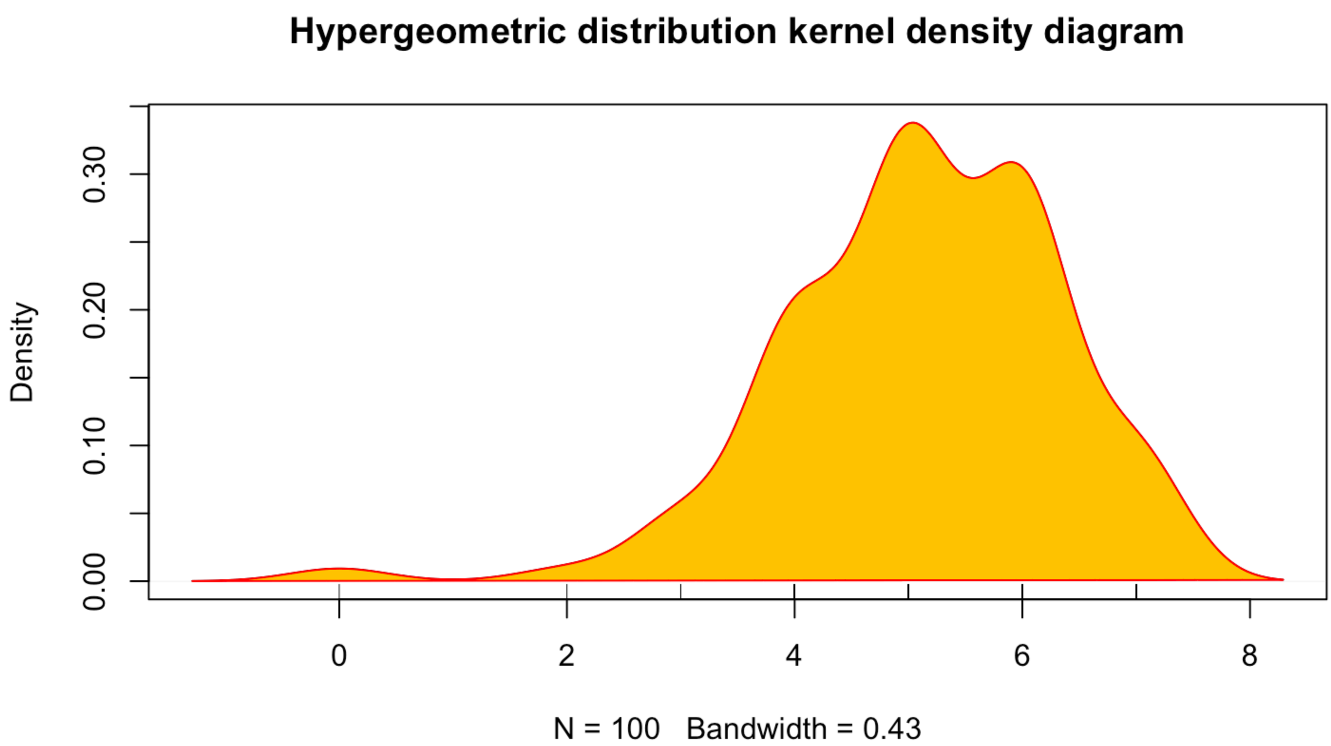
R实现

```
choose(70,7)*choose(30,3)/choose(100,10)
```

```
##[1] 0.2811634
```

画图

```
x1<-round(rhyper(nn=100,m=70,n=30,k=7));x1
plot(density(x1),main = "Hypergeometric distribution kernel density diagram")
polygon(density(x1),col=7,border = "red")
rug(x1,col="black")
```



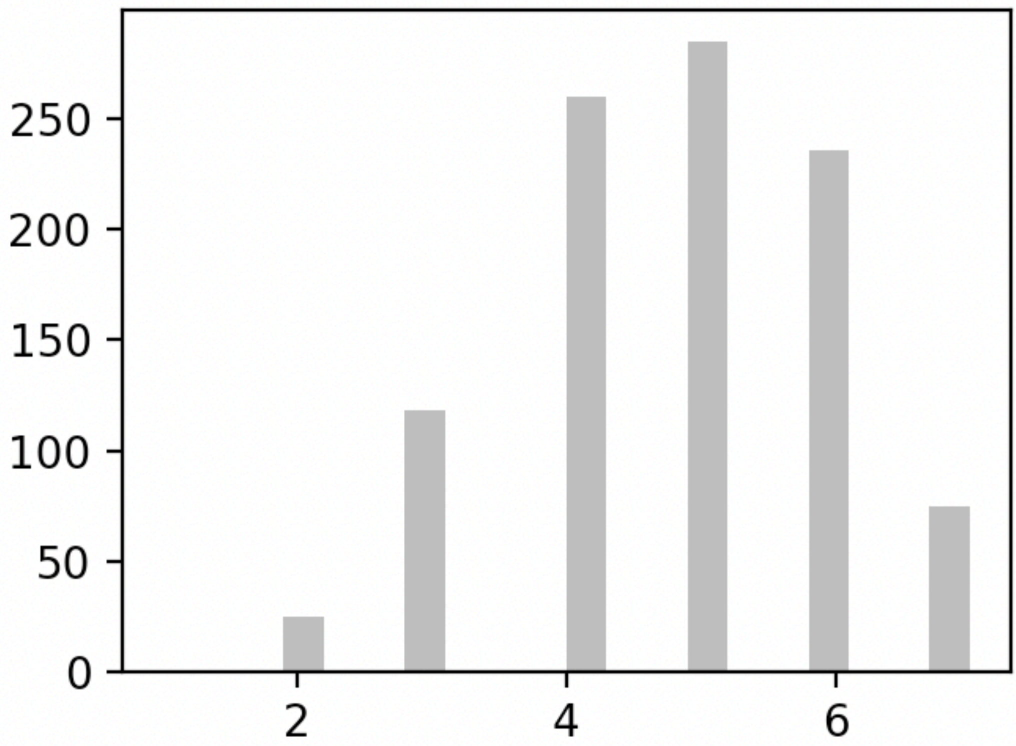
Python实现

```
from scipy.stats import multivariate_hypergeom
a = multivariate_hypergeom.pmf(x =[7,3], m =[70,30],n = 10)
#在70个一等品中取7个，30个二等品中取3个，总样本为10
print(a)
```

```
##0.2811633943025375
```

python模拟

```
import numpy as np
import matplotlib.pyplot as plt
s = np.random.hypergeometric(70,30,7,size=1000)
p =1- sum(s>=4)/1000.
fig = plt.figure(figsize=(8,6))
a1 = fig.add_subplot(2,2,1)
a1.hist(s ,bins=20,color='k',alpha=0.3)
plt.show()
```



随机数模拟

```
import numpy as np
n, p = 9, 0.1
s = np.random.hypergeometric(70,30,7,size=1000000)
print(s)
```

```
##[3 5 5 ... 6 6 6]
```

参考

- 王学民. 应用多元统计分析[M]. 第5版. 上海: 上海财经大学出版社, 2017.