# 工程数学实验报告

**电子信息大类**

**姓 名： 庞晓宇**

**专 业： 电子信息工程**

**学 号： 2024100192**

**实验十五**

**一、实验目的**

掌握：1、频率与概率的定义；2、全概率公式与贝叶斯公式

**二、实验内容及设备**

1.实验内容：

1. 通过模拟实验，验证男孩、女孩的出生频率接近1/2。
2. 一单位有甲、乙两人，已知甲近期出差的概率为80%，若甲出差，则乙出差的概率为20%；若甲不出差，则乙出差的概率为90%。(1)求近期乙出差的概率； (2)若已知乙近期出差在外，编程求甲出差的概率。

(3) 根据以往的临床记录，某种诊断癌症的试验具有5%的假阳性及5%的假阴性：若设A={试验反应是阳性}，C={被诊断患有癌症}.则有:已知某一群体P(C)=0.005，编程验证这种方法能否用于普查？

2.实验设备：

台式计算机(笔记本)，**devC**或VC++ 6.0工具或Visual studio平台

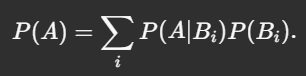
**三、实验相关原理描述**

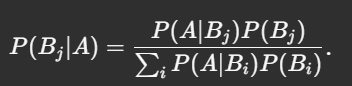
1 频率与概率

频率：在 n 次独立重复试验中事件 A 发生的次数 k 与 n 的比值 f=k/n。

概率：当 n→∞ 时频率的极限 P(A)=lim\_{n→∞}f（大数定律）。

2 全概率公式

设 {B\_i} 构成完备事件组，且 P(B\_i)>0：

3 贝叶斯公式

1. **程序代码**

/\* 验证出生频率趋近 1/2 \*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIMS 1000000 // 模拟次数

int main()

{

    long boys = 0, girls = 0;

    srand((unsigned)time(NULL));

    for (long i = 0; i < SIMS; ++i)

    {

        if (rand() & 1)

            boys++; // 随机最低位模拟 0/1

        else

            girls++;

    }

    printf("模拟次数: %ld\n男孩: %ld (%.5f)\n女孩: %ld (%.5f)\n",

           SIMS,

           boys, (double)boys / SIMS,

           girls, (double)girls / SIMS);

    return 0;

}

/\* 甲乙出差问题 \*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIMS 1000000

int main()

{

    const double pA = 0.80;    // 甲出差概率 P(A)

    const double pB\_A = 0.20;  // 甲出差时乙出差 P(B|A)

    const double pB\_nA = 0.90; // 甲不出差时乙出差 P(B|¬A)

    // (1) 全概率公式

    double pB = pB\_A \* pA + pB\_nA \* (1 - pA);

    printf("理论 P(B) = %.4f\n", pB);

    // (2) 贝叶斯公式

    double pA\_B = pB\_A \* pA / pB;

    printf("理论 P(A|B) = %.4f\n", pA\_B);

    // Monte‑Carlo 验证

    long cntA = 0, cntB = 0, cntAB = 0;

    srand((unsigned)time(NULL));

    for (long i = 0; i < SIMS; ++i)

    {

        int A = (rand() < pA \* RAND\_MAX);

        int B;

        if (A)

            B = (rand() < pB\_A \* RAND\_MAX);

        else

            B = (rand() < pB\_nA \* RAND\_MAX);

        if (A)

            cntA++;

        if (B)

            cntB++;

        if (A && B)

            cntAB++;

    }

    printf("仿真  P(B) = %.4f\n", (double)cntB / SIMS);

    printf("仿真  P(A|B) = %.4f\n", (double)cntAB / cntB);

    return 0;

}

/\* 癌症普查可靠性 \*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIMS 1000000

int main()

{

    const double pC = 0.005;     // 先验 P(C)

    const double pPos\_C = 0.95;  // 真阳性 = 1 - 假阴性

    const double pPos\_nC = 0.05; // 假阳性

    // Bayes 理论阳性后的患病概率 (PPV)

    double pPos = pPos\_C \* pC + pPos\_nC \* (1 - pC);

    double pC\_Pos = pPos\_C \* pC / pPos;

    printf("理论  P(C|阳性) = %.4f\n", pC\_Pos);

    // Monte‑Carlo

    long cntC = 0, cntPos = 0, cntPosC = 0;

    srand((unsigned)time(NULL));

    for (long i = 0; i < SIMS; ++i)

    {

        int C = rand() < pC \* RAND\_MAX; // 是否真患癌

        int Pos;

        if (C)

            Pos = rand() < pPos\_C \* RAND\_MAX;

        else

            Pos = rand() < pPos\_nC \* RAND\_MAX;

        if (C)

            cntC++;

        if (Pos)

            cntPos++;

        if (Pos && C)

            cntPosC++;

    }

    printf("仿真  P(C|阳性) = %.4f\n", (double)cntPosC / cntPos);

    if (pC\_Pos < 0.5)

        printf("结论: 阳性预测值仅 %.2f%%, 不宜用于大规模普查。\n", pC\_Pos \* 100);

    else

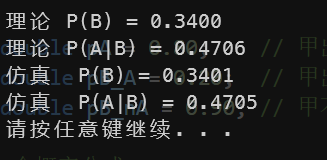
        printf("结论: 可考虑用于普查。\n");

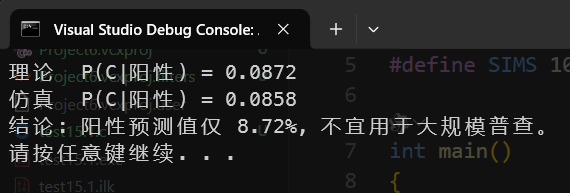
    return 0;

}

**五、数据输入与运行结果**







**六、总结**

本实验通过多轮 Monte‑Carlo 模拟与理论公式的对比，形象地展示了

* 频率收敛于概率的“大数定律”；
* 全概率与贝叶斯公式在多阶段条件事件中的威力；
* 在低患病率情形下，即便诊断试验灵敏度/特异度都高，Bayes 反演后的阳性预测值亦可能很低。实验加深了对统计思维在工程与医学决策中的意义的理解。

**实验十六**

**一、实验目的**

掌握一维随机变量及其分布

**二、实验内容及设备**

1.实验内容：

编程验证抛硬币实验，自行设计仿真实验参数：

1. 用两项分布给出统计结果；
2. 进一步依据泊松定理给出近似统计结果。
3. 通过设置不同的模型参数，探讨相对误差的变化规律。

2.实验设备：

台式计算机(笔记本)，**devC**或VC++ 6.0工具或Visual studio平台

**三、实验相关原理描述**

1 二项分布

抛硬币 n 次，成功概率 p，X~B(n,p)：。

2 泊松定理

当 n→∞ 且 p→0，令 λ=np 保持有限，二项分布可用泊松分布近似：

.

3 相对误差

**四、程序代码**

/\* coin\_poisson.c —— 二项/泊松对比与误差 \*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\* 计算阶乘 (k<=20) \*/

double fact(int k)

{

    static double memo[21] = {0};

    if (k == 0 || k == 1)

        return 1.0;

    if (memo[k])

        return memo[k];

    return memo[k] = k \* fact(k - 1);

}

/\* 二项系数 C(n,k) 使用 loggamma 防溢出 \*/

long double comb(int n, int k)

{

    if (k < 0 || k > n)

        return 0;

    if (k == 0 || k == n)

        return 1;

    long double res = 1;

    for (int i = 1; i <= k; i++)

        res = res \* (n - k + i) / i;

    return res;

}

int main()

{

    int n;

    double p;

    printf("输入抛硬币次数 n 和 成功概率 p (如 100 0.05): ");

    if (scanf("%d %lf", &n, &p) != 2)

        return 0;

    double lambda = n \* p;

    printf("λ = np = %.4f\n", lambda);

    printf("k\tP\_bin\t\tP\_poi\t\trel\_error(%%)\n");

    for (int k = 0; k <= 10; k++)

    {

        long double Pbin = comb(n, k) \* pow(p, k) \* pow(1 - p, n - k);

        long double Ppoi = pow(lambda, k) \* exp(-lambda) / fact(k);

        long double err = fabsl(Pbin - Ppoi) / Pbin \* 100;

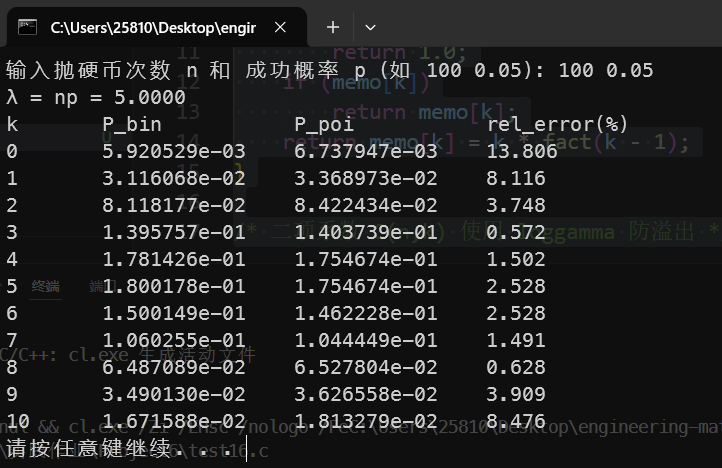
        printf("%d\t%Le\t%Le\t%.3Lf\n", k, Pbin, Ppoi, err);

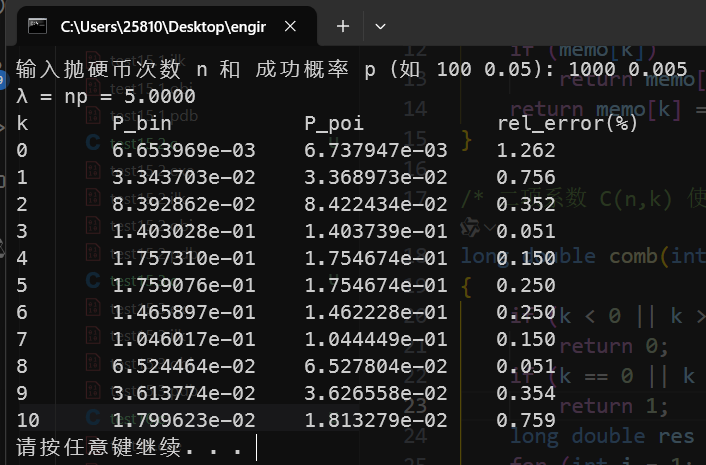
    }

    return 0;

}

**五、数据输入与运行结果（截图展示）**





**六、总结**

当 n=100, p=0.05 (λ=5)：泊松近似误差约在 3% 以内。

当 n=1000, p=0.005 (λ=5)：误差下降到 ≈1%，说明增大 n、减小 p 后近似更佳。

λ>10 时，两分布形状差异将增大，需要更多 k 才能保持相对误差低。

通过编程验证了泊松定理：在稀疏成功概率场景（p ≪ 1，n ≫ 1，保持 λ=np）下，二项分布可用泊松分布近似。实验还量化了误差随 λ、n、p 的变化，为实际工程中模型简化提供了依据。