关于一元线性回归分析的代码实现：

**package** artificialIntelligence;

**import** java.util.Scanner;

**public** **class** linearRegression {

**static** **double**[] *x*=**new** **double**[100];

**static** **double**[] *y*=**new** **double**[100];

**static** **double** *count*=0;

**static** **double** *average*;

**public** **static** **void** main(String[] args) {

Scanner read=**new** Scanner(System.***in***);

System.***out***.println("请输入样本数：");

*count*=read.nextDouble();

System.***out***.println("请输入x:");

**for**(**int** i=0;i<*count*;i++)

{

*x*[i]=read.nextDouble();

}

System.***out***.println("请输入y:");

**for**(**int** i=0;i<*count*;i++)

{

*y*[i]=read.nextDouble();

}

linearRegression l=**new** linearRegression();

l.calculatXAverage();

**double** w=l.forW();

**double** b=l.forB(w);

System.***out***.println("y="+w+"x"+"+"+b);

System.***out***.println(*average*);

}

**public** **void** calculatXAverage() {

**double** ammount=0;

**for**(**int** i=0;i<*count*;i++) {

ammount=ammount+*x*[i];

}

*average*=cutThree(ammount,

*count*);

}

**public** **double** cutThree(**double** a,**double** b) {

**double** c=a/b\*1000;

**int** d=(**int**)c;

**double** e=(**double**)(d%1000)/1000.0;

**double** f=d/1000;

**double** g=f+e;

**return** g;

}

**public** **double** forW() {

**double** squareX=0;

**for**(**int** i=0;i<*count*;i++) {

squareX=squareX+Math.*pow*(*x*[i], 2);//求x的平方和

}

**double** addX=0;

**for**(**int** i=0;i<*count*;i++) {

addX=addX+*x*[i];

}

**double** whatever=Math.*pow*(addX, 2);

**double** xAddSA=cutThree(whatever, *count*);

**double** up=0;

**for**(**int** i=0;i<*count*;i++) {

up=up+*y*[i]\*(*x*[i]-*average*);

}

**double** w1=squareX-xAddSA;

**double** w=cutThree(up, w1);

**return** w;

}

**public** **double** forB(**double** w) {

**double** b;

**double** c=0;

**for**(**int** i=0;i<*count*;i++) {

c=c+(*y*[i]-w\**x*[i]);

}

b=cutThree(c, *count*);

**return** b;

}

}