

Rsample.txt

第1个问题：用克莱姆法则求解线性方程组

```
a=c(1,2,0,1,2,-1,3,-1,-1,3,-1,1,3,-2,1,4)
A=matrix(a,4,4)          ##创建一个线性方程组的系数矩阵A
b=c(2,7,6,-4)            ##创建一个线性方程组的常数列向量b
d0=det(A)                ##计算系数矩阵的行列式
for(i in 1:ncol(A)){      ##对A的每一列进行循环
  B=A                    ##另设一个矩阵B与A相同，以保证A不变
  B[,i]=b                ##将B的第i行替换为b
  d=det(B)               ##计算B的行列式
  x=d/d0                 ##计算解向量的第i个值
  cat("x",i,"=",x,"\n")  ##输出解向量
}
```

#####第2个问题：将一组基 $a_1=(1,1,1)$ $a_2=(1,2,1)$ $a_3=(0,1,1)$ 化为标准正交基

```
a1=c(1,1,1)
a2=c(1,2,1)
a3=c(0,1,1)              ##定义3个向量
b1=a1                    ##正交化
c1=crossprod(b1,a2)/crossprod(b1)
b2=a2-c1*b1
c2=crossprod(b1,a3)/crossprod(b1)
c3=crossprod(b2,a3)/crossprod(b2)
b3=a3-c2*b1-c3*b2
d1=sqrt(b1[1]^2+b1[2]^2+b1[3]^2)    ##求正交化后的向量的模
d2=sqrt(b2[1]^2+b2[2]^2+b2[3]^2)
d3=sqrt(b3[1]^2+b3[2]^2+b3[3]^2)
x1=b1/d1                  ##单位化
x2=b2/d2
x3=b3/d3
cat("x1=(",x1,") x2=(",x2,") x3=(",x3,")")    ##输出标准正交基
```

#####第3个问题：生成一个n阶Hilbert矩阵

```
F=function(n){
  a=rep(NA,n^2)          ##创建一个含有n^2个元素的空向量
  A=matrix(a,n,n)        ##创建一个n维空矩阵
  for(i in 1:n){
    for(j in 1:n){
      A[i,j]=1/(i+j-1)    ##定义矩阵中每个数字
    }
  }
  print(A)               ##输出该矩阵
}
```

Rsamle.txt

#####第4个问题：用泰勒展开逼近 $\cos(x)$

```
cos.taylor=function(x,k){
  n=1:k
  z=(-1)^n*x^(2*n)/factorial(2*n)
  y=1+sum(z)
  return(y)
}
```

#####第5个问题：正态分布的点估计

```
norm.point_estim=function(sample.norm,estim.way=0,display=0){
  n=length(sample.norm)
  norm.mean=mean(sample.norm)
  norm.sd=sqrt(var(sample.norm)*(n-1)/n)
  if(display!=0){
    print(paste("服从正态分布的样本的平均值点估计为：",norm.mean))
    print(paste("服从正态分布的样本的标准差点估计为：",norm.sd))
  }
  result=data.frame(norm.mean=norm.mean,norm.sd=norm.sd)
  result
}
```

#####第6个问题：求置信区间

```
F=function(x,sigma=-1,alpha=0.05){
  n=length(x)      ##求数据个数
  xb=mean(x)        ##求数据均值
  if(sigma>=0){
    tmp=sigma/sqrt(n)*qnorm(1-alpha)      ##方差已知
    df=n
  }else{
    tmp=sd(x)/sqrt(n)*qnorm(1-alpha)      ##方差未知
    df=n-1
  }
  data.frame(mean=xb,df=df,a=xb-tmp,b=xb+tmp)
}
```

#####第7个问题：求函数 $f(x)=x^2+1$ 在区间 $[-2,1]$ 上按间隔0.001平分的黎曼和

```
Int=function(f,xmin,xmax,dx){
  x=seq(xmin,xmax,by=dx)      ##将区间 $[-2,1]$ 按间隔0.001平分
  y=f(x)                      ##每个点的函数值
  y=y[-length(y)]
  z=y*dx                      ##每个小区域的面积
  s=sum(z)                    ##黎曼和
}
```

Rsample.txt

```

        return(s)
    }
    F=function(x){
        y=x^2+1
        return(y)
    }
    Int(F,-2,1,0.001)

```

#####第8个问题：用for语句求斐波那契数列的前n项之和

```

F=function(n){
    a=rep(NA,n)      ##定义一个空向量
    for(i in 1:n){
        if(i==1) a[i]=0
        if(i==2) a[i]=1
        if(i>=3) a[i]=a[i-1]+a[i-2]  ##计算斐波那契数列第i项的值
    }
    s=sum(a)          ##计算向量a各项之和s
    print(s)          ##输出s
}

```

#####第9个问题：输出所有的水仙花数(各位数字立方之和等于该数本身的三位数)

```

for(i in 100:999){
    ##循环开始
    b3=(i-i%100)/100      ##得到百位数
    b2=(i-b3*100-(i-b3*100)%10)/10  ##得到十位数
    b1=i-b3*100-b2*10     ##得到个位数
    if(b3^3+b2^3+b1^3==i)  ##判断是否为水仙花数并输出
        print(i)
}

```

#####第10个问题：计算半径为r的圆的周长和面积

```

F=function(r){
    c=2*pi*r      ##计算周长
    s=pi*r^2      ##计算面积
    cat("该圆的周长为",c)
    cat("该圆的面积为",s)
}

```

#####第11个问题：计算向量x=(23,41,35,67,37,57,86,23)中最大5个数的均值

```

F=function(x){
    x1=sort(x)      ##将向量x从小到大排序
    x2=rev(x1)      ##将x1逆序输出
}

```

```

                                Rsample.txt
    y=x2[1:5]                    ##取前5个数字
    mean=mean(y)                 ##求其均值
    return(mean)
}
x=c(23,41,35,67,37,57,86,23)
F(x)

```

#####第12个问题： 在一个长度为 n ($n < 1000$)的整数序列中，判断是否存在某两个元素之和为 k

```

F=function(x,k){
    n=length(x)                ##得到向量x的长度
    h=0                         ##用来计算和为k的数对的个数
    for(i in 1:n){
        for(j in 1:n){
            if(i!=j){
                a=x[i]+x[j]
            }else{
                next
            }
            if(a==k) h=h+1
        }
    }
    if(h>0){
        cat("yes")
    }else{
        cat("no")
    }
}

```

#####第13个问题： 求解一元二次方程 $ax^2+bx+c=0$

```

F=function(a,b,c){
    d=b^2-4*a*c
    if(d<0){
        print("此方程在实数范围内无解")
    }else{
        if(d==0){
            x1=-b/(2*a)
            cat("x1=x2=",x1)
        }else{
            x1=(-b+sqrt(d))/(2*a)
            x2=(-b-sqrt(d))/(2*a)
            cat("x1=",x1,"\n")
            cat("x2=",x2)
        }
    }
}

```

}

#####第14个问题：用二分法求方程 $x^3-x-1=0$ 在区间 $[1,2]$ 上精确度为0.00001的近似根

```
F=function(f,a,b,dx){          ##f为一个函数，a、b分别为所求区间的左右端
点，dx为精确度
  if(f(a)*f(b)>0){              ##若区间端点函数值之积为正，不一定有解
    print("该区间内不一定有解")
  }else{
    repeat{
      if(b-a<dx) break
      c=(a+b)/2
      if(f(a)*f(c)<0){
        b=c
      }else{
        a=c
      }
    }
    x=(a+b)/2
    print(x)
  }
}
f=function(x){                  ##定义函数y=x^3-x-1
  y=x^3-x-1
  return(y)
}
F(f,1,2,0.00001)
```

#####第15个问题：判断今年是否为闰年

```
today=Sys.Date()               ##得到今天的日期
y=format(today,"%Y")           ##得到今年的年份
year=as.numeric(y)             ##将年份转化为数值型
if(year%%4==0&year%%100!=0|year%%400==0){      ##判断今年是否为闰年
  cat(year,"is a leap year","\n")              ##若是闰年，输出一条语句
}else{
  cat(year,"is not a leap year","\n")          ##若不是闰年，输出另一条语句
}
句
}
```

#####第16个问题：模拟一个银行账户，具有取款、存款、显示余额的功能，

```
open.account=function(total){
  list(                                ##创建一个包含三
个函数的列表
    deposit=function(amount){          ##存款并
```

Rsampl.txt

显示余额

```
if(amount<=0){  
  print("存款金额必须为正数")  
}else{  
  total=total+amount  
  cat("您的账户余额为",total)  
}
```

```
},
```

```
withdraw=function(amount){
```

##取款并

显示余额

```
if(amount>total){  
  print("账户余额不足")  
}else{  
  total=total-amount  
  cat("您的账户余额为",total)  
}
```

```
},
```

```
balance=function(){
```

##显示余

额

```
cat("您的账户余额为",total)
```

```
}
```

```
)
```

```
}
```

```
a=open.account(100)
```

##开设一个余额为100元的账户

```
a$deposit(30)
```

##进行存款操作

```
a$withdraw(50)
```

##进行取款操作

```
a$balance()
```

##显示余额