



第02章 循环、 向量和应用

顾立平

019 循环



```
R Console

R是自由软件，不带任何担保。
在某些条件下你可以将其自由散布。
用'license()'或'licence()'来看散布的详细条件。

R是个合作计划，有许多人为之做出了贡献。
用'contributors()'来看合作者的详细情况
用'citation()'会告诉你如何在出版物中正确地引用R或R程序包。

用'demo()'来看一些示范程序，用'help()'来阅读在线帮助文件，或
用'help.start()'通过HTML浏览器来看帮助文件。
用'q()'退出R.

> setwd("C:/Users/user/Desktop/code")
> # 初始化total为0，用于后续的累加操作
> total=0
>
> # 使用for循环从1遍历到100
> for (i in 1:100){
+   # 每次循环，都将1/i^2加到total上
+   total=total + 1/i^2
+ }
> # 输出累加后的total值
> total
[1] 1.634984
> |
```



循环的作用



通过避免重复代码，循环体在满足条件时多次执行，减少代码量，提升编程效率。

”



提高代码效率的循环

对于需要重复相同操作的复杂任务，如数据处理，循环结构封装操作并按需控制执行，简化实现过程。

”



简化复杂任务的循环

循环允许根据运行时条件动态调整执行流程，如遍历数据结构或处理用户输入，增加程序灵活性。

”



实现动态功能的循环

循环是实现排序、搜索等算法的基础，有效组织循环以高效解决各种算法问题。

”



支持算法的循环结构



020 利用函数求和



```
R Console

> setwd("C:/Users/user/Desktop/code")
> # 初始化total为0, 用于后续的累加操作
> total=0
>
> # 使用for循环从1遍历到100
> for (i in 1:100){
+   # 每次循环, 都将1/i^2加到total上
+   total=total + 1/i^2
+ }
> # 输出累加后的total值
> total
[1] 1.634984
> # 定义一个函数f, 该函数接受一个参数x, 返回1/x^2
> f=function(x){1/x^2}
>
> # 使用sum函数对1到100的每一个数应用函数f, 并求和
> sum(f(1:100))
[1] 1.634984
>
> # 使用sapply函数对1到100的每一个数应用一个匿名函数(该函数返回1/x^2),
> # 然后用sum函数对结果求和
> sum(sapply(1:100,function(x){1/x^2}))
[1] 1.634984
> |
```



向量的作用（在编程中）



向量数据表示

在编程中，向量用于表示多维空间的点、颜色、尺寸，如图形学的顶点位置、颜色，或机器学习的特征向量。

向量操作简化

向量支持加法、减法、点积等操作，简化数学计算，提高代码可读性和可维护性。

向量性能优化

向量化操作利用底层优化和并行计算，提高效率，减少循环处理带来的计算开销和内存占用。



021 向量里的元素



```
R Console

> for (i in 2:100){
+   # vector1的第i个元素是前一个元素加上1/i^2
+   vector1[i]=vector1[i-1] + 1/i^2
+ }
> # 输出vector1的前3个和最后3个元素
> vector1[c(1:3,98:100)]
[1] 1.000000 1.250000 1.361111 1.634782 1.634884 1.634984
>
> # 重新定义函数f（与前面的定义相同）
> f=function(x){1/x^2}
>
> # 对1到100的每一个数应用函数f，然后使用cumsum函数计算累积和，
> # 结果保存在vector2中
> vector2=cumsum(f(1:100))
> # 输出vector2的前3个和最后3个元素
> vector2[c(1:3,98:100)]
[1] 1.000000 1.250000 1.361111 1.634782 1.634884 1.634984
>
> # 使用sapply函数对1到100的每一个数应用一个匿名函数（返回1/x^2），
> # 然后用cumsum函数计算累积和，结果保存在vector3中
> vector3=cumsum(sapply(1:100,function(x){1/x^2}))
> # 输出vector3的前3个和最后3个元素
> vector3[c(1:3,98:100)]
[1] 1.000000 1.250000 1.361111 1.634782 1.634884 1.634984
> |
```



循环和向量的应用



循环与向量的科学商业应用

科学领域：数据处理、模拟实验、算法开发，如物理模拟、图形渲染和机器学习。



向量在科学与商业中的应用

科学领域包括物理学的力、速度计算，以及信号处理中的向量应用。商业中，向量运用于数据分析、财务计算、库存管理和销售分析。



矩阵向量在科商中的应用

科学领域用矩阵描述量子力学和结构动力学，商业中则应用于PCA数据降维、市场预测矩阵模型及风险管理的协方差矩阵。



022 矩阵里的向量



```
R Console
[2,] 5 2 4 2 5 3 6 6 8 1
[3,] 1 10 9 3 5 10 10 2 2 5
[4,] 9 1 5 9 4 1 8 5 3 10
[5,] 10 8 4 9 2 10 4 4 8 2
>
> # 初始化一个长度为10的数值型向量count1
> count1=numeric(10)
>
> # 使用双层for循环遍历矩阵A的每一个元素
> for(col.j in 1:10){
+   for(row.i in 1:5) {
+     # 如果A的第row.i行、第col.j列的元素是偶数，则count1的第col.j个元素加1
+     count1[col.j]=count1[col.j] + ifelse(A[row.i,col.j]%%2==0,1,0)
+   }
+ }
> # 输出count1
> count1
[1] 1 4 2 2 3 3 5 4 4 2
>
> # 使用apply函数对矩阵A的每一列应用一个匿名函数，
> # 该函数计算每列中偶数的个数，结果保存在count2中
> count2=apply(A,2,function(x){sum(ifelse(x%%2==0,1,0))})
> # 输出count2
> count2
[1] 1 4 2 2 3 3 5 4 4 2
```





谢谢

gulp@mail.las.ac.cn