2021年数理统计上机课

–数据结构

黄启岳

北京师范大学统计学院

2021 年 3 月 16 日

目录

常见对象与属性

R语言的操作需要借助对象，这些对象主要通过名称、内容与数据类型刻画，包含“类型”与“长度”两种内在属性。

常见数据类型

数据类型常见为以下四种：

(1)数值型(numeric)：包括数值型(numeric，通常表达实数)、浮点型(float，采用浮点格式存储)、双精度型(double，固定长充浮点格式存储)、整型(int，仅表示整数)

(2)字符型(character)：类字形单位或符号，一组字符组成了字符串，输入时须加上双引号"

(3)逻辑型(logical)：FALSE、TRUE和NA

(4)复数型(complex)：包括实部(Re())与虚部(Im())

查看数据类型一般方法：typeof()与mode()

判断数据类型一般方法：is.X()，如is.numeric()

数值型数据计算

数值型数据无论是常亮还是变量都支持基本的数学运算与基本数学函数(例如exp())。最常用的类型为浮点型，如果使用整型需要在数字后加上L。

针对复数型数据，可以写作：

$$z = a + bi = Re(z) + iIm(z), \text{其中} i = \sqrt{-1}$$

注：部分数学函数不支持复数型数据，例如开方(sqrt())在默认状态下不支持复数运算，如sqrt(-17)；但输入sqrt(-17+0i)则输出$\sqrt{-17}$的计算值。

逻辑型变量与条件结构

一般用在条件结构以及数据筛选中，通过as.numeric(), 可以将一个逻辑数据变成一个0-1 数值数据。类似于概率论中的示性函数，将一个一般空间映射到$\mathbb{R}^1$空间。

R中常用的条件结构为if-else结构，与大多数编程语言结构类似，使用大括号判断条件语句，对缩进没有要求(区别于python使用缩进判断条件语句)。

判断语句常用格式：

if(条件1){

操作1}else(条件2){

操作2 }

如果判断的条件较多，可以改用elseif，也可以一直使用if。条件值即是需要返回逻辑值的命令。

无穷与缺失值

常见的无穷包括正无穷Inf(infinity)与负无穷-Inf。Inf事实上存在下界。

常见缺失值包括NA(Not Available)，NAN(Not a Number)和NULL。is.na()和is.nan()都可以判断缺失值，但只有is.nan()可以区分NA与NAN的区别。

NA默认储存为逻辑型，这也表示其不可以直接和数值型变量进行数值运算。指定NA的类型一般使用*NA_TYPE_*，例如：*NA_integer_*。

思考：length(c(NA,Inf,NULL)) = ?为什么？

常用数据结构

R中的常用数据结构为：向量(vector)，矩阵(matrix)，数组(array)，数据框(data.frame)，列表(list)。

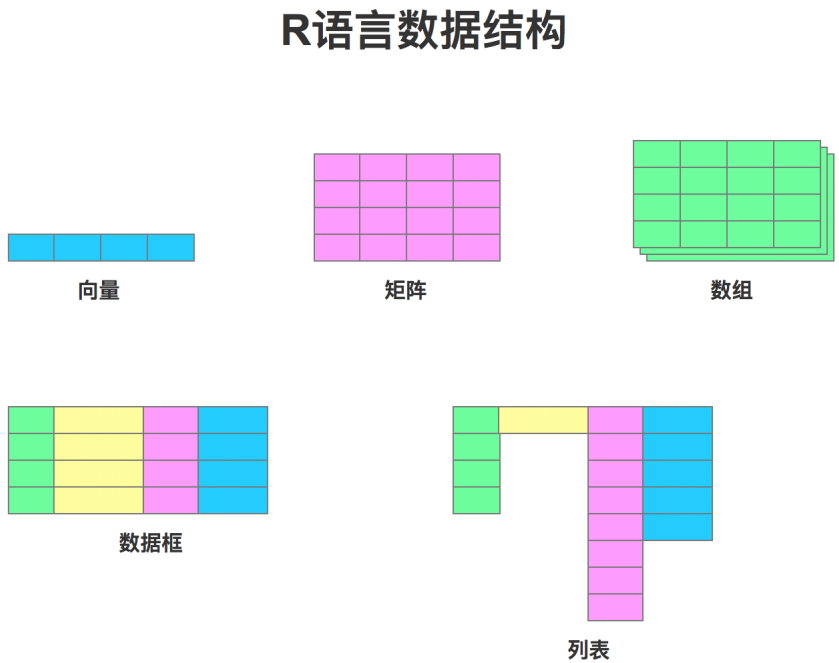

图 1: 常见数据结构图示

图1(来源：https://blog.51cto.com/h2appy/1862371)也反映了这几种数据结构的兼容方向，向量是特殊的矩阵，矩阵是特殊的数组，数据框是有名字的矩阵，列表则全部兼容。

在可以向下兼容的时候，as.TYPE()系列转换数据结构的函数，向量转矩阵会多出一个维度的长度信息(行为1), 矩阵转向量会按列拉直，数据框转矩阵时会只保留数据，列表在可以兼容的时候也会只保留数据。

判断数据结构类型可使用函数is.TYPE()；创建则可以直接使用TYPE()，如matrix()。

以下重点介绍矩阵与数据框。这是相对最常见的两种数据结构。

矩阵及其运算

首先提到向量，R中默认向量均为列向量。快速生成向量主要有以下方法：

(1)rep(n,m):生成长度为m，所有元素全部为n的向量。

(2)seq(m,n,by = v)：生成从m到n的间隔为v的向量，v对符号有要求。当v=1时，等价于m:n。

(3)c()：创建指定数据(可以有字符，缺失值和逻辑值) 的向量。没有数据则直接生成一个空向量，占据部分内存。常用于组合向量。

向量x的元素索引通过x[index]来实现,index是索引的下标，若对应一个整数(运算精度下的浮点也可以)向量，则直接显示对应下标元素，如x[1]，计数从1开始；若index是逻辑型向量，则选取位置对应为TURE的变量。

构建出向量后，即可根据向量构造矩阵。一般格式为：matrix(data = 数据, nrow = a, ncol = b, byrow = T/ F, dimnames = NULL)

如果行数和列数的乘积与数据的总长度不符，程序仍然可以运行，但产生的结果会略去一部分数据。

矩阵运算中加减乘除均逐点计算(数乘计算注意与线性代数中矩阵的数乘运算区分)。R中的所有向量视为列向量，视为单列的矩阵，矩阵和相同行数的向量的加减乘除为逐行运算，即矩阵的每行都与对应行的向量进行加减乘除。

矩阵与矩阵运算列举如下：矩阵之间的乘法为%*%，求逆为solve()，张量积为kronecker()，类外积运算为outer(),内积运算为crossprod()。

部分对矩阵操作函数：最大max()，最小min()，全平均mean()，全求和sum()，列数ncol()，行nrow()，行求和rowSum()，列求和colSums()，行平均rowMeans()，列平均colMean()，行拼接cbind()，列拼接rbind()，转置t()，下三角元素lower.tri()，上三角元素upper.tri()，对角线diag()，行列式det()。

部分常见的矩阵分解：谱分解eigen()，QR分解qr()，奇异值分解svd()，Choleskey分解chol()。

数据框及处理

数据框的创建为data.frame()，输入的数据如果是矩阵或向量，则返回同样维度的数据框，但是多了名称。形式为：data.frame(..., row.names = NULL, check.rows = FALSE,check.names = TRUE, fix.empty.names = TRUE,stringsAsFactors =default.stringsAsFactors())

若按照向量名称逐向量输入数据(需要保证向量长度相同)，则可以指定变量名称。

有时对于一个数据集需要创建一些新变量或统计量，还可能对变量进行重命名或对观测值进行重新排序，以便数据更容易处理。

首先介绍R语言自带的一些常用于数据预处理的函数。假定数据集命名为data。

(1)head(data)：展示表头。

(2)str(data):逐行显示数据中列的内容。

(3)summary(data)：提供最小值、最大值、四分位数和数值型变量的均值，以及因子向量和逻辑型向量的频数统计。

这里额外介绍几个其他包中的函数，用于进行对数据基本的分析，相当于“强化版”summary。

(1)psych::describe(data):计算非缺失值的数量、平均数、标准差、中位数、截尾均值、绝对中位差、最小值、最大值、值域、偏度、峰度和平均值的标准误

(2)pastecs::stat.des(data):格式为stat.desc(data,basic=TRUE,desc=TRUE,norm=FALSE,p=0.95)。其中的data是一个数据框或时间序列。若basic=TRUE(默认值)，则计算其中所有值、空值、缺失值的数量，以及最小值、最大值、值域，还有总和。若desc=TRUE(默认值)，则计算中位数、平均数、平均数的标准误、平均数置信度为95%的置信区间、方差、标准差以及变异系数。最后，若norm=TRUE(非默认)，则返回正态分布统计量，包括偏度和峰度(以及统计显著程度)和Shapiro–Wilk正态检验结果。

除了获取最基本的描述性统计信息，我们有时还需要对数据进行适当的分组。R语言自带函数有table(data)与aggregate(data,by = type,FUN = function)函数，分别统计不同元素的频数以及对数据进行依照函数的分类计算。

最后介绍一个专门用于处理数据框的程序包dplyr。dplyr是一个利用pipline语法快速处理数据的R语言包。详细信息可以参考`https://github.com/tidyverse/dplyr`。dplyr包含非常多的功能，这里介绍几个较为常用的函数。

(1)filter(data,条件)：filter()可以按逻辑条件筛选出符合要求的子数据集，返回与data相同类型的对象。原数据集行名称会被过滤掉。

(2)arrange(data,变量(列名))：按给定的列名依次对行进行排序。默认为升序，加上desc(列名)则依指定列降序。

(3)select(data,变量特征)：用列名作参数来选择子数据集。搭配dplyr中starts_with，ends_with，contains，matches，one_of，num_range和everything等使用，也可以指定相关列。

(4)mutate(data,新列= f(原有列))：对已有列进行数据运算并添加为新列，且保留原有变量。

(5)summarise(data,f(指定列))：对数据框调用函数进行汇总操作，返回一维的结果。这里的f可以为mean，sd，max等。写作summarize()亦可，效果与summarise()相同，之后不再区分两者。

(6)group_by(data,条件)：对数据集按照给定变量分组，返回分组后的数据集。

上述几个函数也可以考虑联合使用，如summarise()经常结合group_by使用：summarise(group_by(data,TYPE1,...)，newstat1, newstat2, newstat3...)。这样函数计算的仅仅是分组后的对应新建统计量。

更多功能可以参考dplyr的帮助，也可以参考文章**【R语言】必学包之dplyr包**。逻辑运算符可以参考**R：算术和逻辑运算符及数值**。

在summarise()与group_by的例子中，我们发现有时程序会很冗长。为了简化程序，dplyr包提供了一种简洁的写法，即使用管道函数%>%。

使用管道函数构造的程序可以理解为一条“生产线”，原始数据即生产线的“原料”，%>%为传输数据的“管道”，各个函数则相当于“加工车间”。

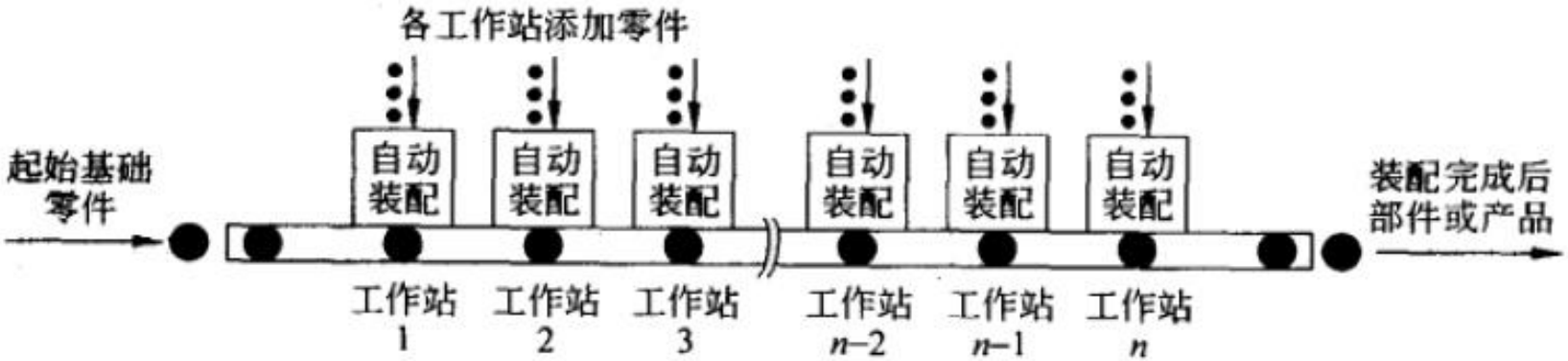

图 2: 流水线示意图

管道函数的作用是将前一个函数的返回值传递到后一个函数，作为下一个函数的第一个参数。这样可以避免反复的赋值、提取等操作，一方面避免了赋值名称混乱等情况，防止程序出错，提高程序的可读性，另一方面则节约了存储区的空间。

常用格式如下：

FUN1 %>% FUN2 %>% FUN3 %>%...

FUN1的位置也可以换成数据集。代码能够实现的前提是：下一个函数参数输入类型兼容前一个函数的返回结果，且管道函数只能传递至第一个参数，之后的参数需要设置。

思考：数据集名称为beijing，存在分类变量CATE，以及数值型变量price，如何计算CATE == "haidian" 且subway == 1的地区price的均值与方差？

更多有关包dplyr的信息，请参考文章：Data transformation，这是R for Data Science的一部分，作者正是tidyverse系列包的作者Hadley Wickham。

数据导入与输出

很多时候我们需要导入别人给的文件，R中支持直接导入多种格式的数据，包括“Excel”，“text”，“SPSS”，“SAS”，“Stata”等等(Rstudio 界面File中Import Dataset)。还有一些专业数据例如地理学常用的GIS数据，可以通过安装相关包(如maptools)导入。

另一种方法则是通过read.(读取)和write.(输出)，例如read.csv("路径/ 文件名.csv")，write.txt(("路径/ 文件名.txt")等。

注意：xlsx作为最常见的一种数据存储方式，R中没有直接读取的函数。可以加载readxl包，使用readxl::read_xlsx("地址")。更特殊的格式则只能使用专门的方法。

文件管理

首先需要给R设定路径，方法为setwd("路径")，注意地址中的“\”要用“/”代替。

指定目录下的文件可以使用dir函数，其格式是dir("目录","文件类型")，文件类型(.txt,.R,.pdf) 不填时则返回该目录下所有文件。另一种方式是list.files('目录')，返回结果类似。创建与移除文件分别可以使用dir.create("文件夹")与dir.remove("文件夹")

处理文件一般使用file.系列函数。常用以下的用法：

(1)file.create("文件名")：创建文件

(2)file.exists("文件名")：查询文件

(3)file.copy("原文件","新文件")：复制文件

(4)file.info("文件名")：文件信息

(5)file.rename("原文件名","新文件名")：重命名

(6)file.remove("文件名")：移除文件

执行后会自动返回逻辑值表示操作成功或失败。注意创建文件类型需要存在于电脑中。

总结

到此为止我们了解了数据类型与结构，以及对数据的基本运算、整理与管理方法。之后会介绍R语言的循环结构，并试着自行编写函数。