

# R

---

数据整理、数据转化、可视化、建立模型、沟通交流、编程

假设生成和假设验证 验证性分析

R包是函数、数据和文档的集合，对R基础功能的扩展

## 基于S4的面向对象编程

---

S4对象系统有明显的结构化特征，更适合面向对象的程序设计

### S4对象简介

S4对象系统是一种标准的R语言面向对象实现方式。S4对象有明确的类定义、参数定义、参数检查、继承关系、实例化等面向对象系统的特征

由于S4对象是标准的面向对象实现方式，有专门的类定义函数 `setClass()` 和类的实例化函数 `new()`，我们看一下 `setClass()` 和 `new()` 是如何动作的。

#### `setClass()`

```
setClass(Class, representation, prototype, contains=character(),
         validity, access, where, version, sealed, package,
         S3methods = FALSE, slots)
```

参数列表：

- `Class`: 定义类名
- `slots`: 定义属性和属性类型
- `prototype`: 定义属性的默认值
- `contains=character()`: 定义父类，继承关系
- `validity`: 定义属性的类型检查
- `where`: 定义存储空间
- `sealed`: 如果设置 `TRUE`，则同名类不能被再次定义
- `package`: 定义所属的包
- `S3methods`: R3.0.0以后不建议使用
- `representation` R3.0.0以后不建议使用
- `access` R3.0.0以后不建议使用
- `version` R3.0.0以后不建议使用

S4 的意思是指 S 语言的版本 4。R 语言本身对面向对象编程的支持不是很理想，S4 类在一定程度上使得 R 语言更适合于面向对象编程。那么，为什么 R 也需要使用面向对象编程呢？我所理解的面向对象编程的好处在于，面向对象编程的方式可以很好地处理现实的问题。因为现实中的每一个需要处理的数据，都对应于一个背后的对象，例如，如果是工资数据，那么工资对应的是一个员工，这个员工会有种种属性，再例如，如果处理的数据是基因表达量，那么这个表达量背后对应的就是基因，基因有其位置信息，有其编码信息，

有其各种 ID 等. 我会需要使用 R 语言处理很多现实的问题. 如果只是简单地处理一个维度的数据, 那么就可以把这些数据看成简单的数字使用 R 语言处理. 但是, 如果要对数据所对应的对象进行多维度地分析和处理, 那么使用面向对象编程的方式就可以使得数据很规整, 并且减少错误的可能性. 如果简单地使用 list 来纵向地包装同一个对象的不同数据或者横向地包装不同对象地同类型数据自然也是可以的, 有的时候甚至很方便, 不过, 在更为复杂的地方, 使用更合适地面向对象的工具显然是个更优的选择.

## 面向对象

面向对象编程有三个重要特点: 封装, 继承和多态性. 封装的好处是把同属一个类的不同的属性能够包装在一起, 不会和外部的变量等相互影响. 那么在 R 的 S4 面向对象编程中, 也应该注意的是, 尽量不要直接引用类的 slot, 最好是通过方法来引用, 也就是通过按键来获得冰激凌, 而不是直接打开储藏盖去取. 不过, 直接引用确实是一个很方便的方式, 在能够保证正确的情况下也可以适当采用. 继承和多态性则有利于代码的重用, 减少由于采用重复相同的代码, 而在代码发生改变的时候造成不同的相关类之间出现不一致. 类的继承关系需要在编程之前妥善设计好, 这样才能有效地发挥面向对象编程的优势.