

# Contents

<b>R 语言的数据读写</b>	<b>1</b>
使用键盘输入数据	1
读取 R 语言包中的数据	1
读取数据文件	2
读取 Excel 文件	3
保存数据到文件	4

## R 语言的数据读写

---

### 使用键盘输入数据

使用 `edit()` 或 `fix()` 函数会自动调用一个文本编辑器。

```
# 首先建立一个空的数据框, 包含 age、gender 和 weight 三个变量
mydata<-data.frame(age=numeric(0),gender=character(0),weight=numeric(0))
# 然后调用 edit() 或 fix() 函数来编辑和修改数据
mydata<-edit(mydata)
#fix(mydata)
mydata
```

```
## [1] age    gender weight
## <0 rows> (or 0-length row.names)
```

### 读取 R 语言包中的数据

R 中包含许多内置数据集。由于 R 提供了这些数据集，你可以应用它们来进行试验，所以它们对学习和使用 R 很有帮助。许多数据集都包含在名为 `datasets` 的 R 包中。该 R 包是 R 的基础包，它在 R 的搜索路径中，因此可以直接应用这些数据集。例如，可以调用内置数据集 `pressure`：

```
head(pressure)
```

```
##   temperature pressure
## 1           0    0.0002
## 2          20    0.0012
## 3          40    0.0060
## 4          60    0.0300
## 5          80    0.0900
## 6         100    0.2700
```

任何 R 包都可以选择包含数据集，来补充 R 包 `datasets` 中的数据。例如，`MASS` 包中包含了许多有趣的数据集。使用带有 `package` 参数的 `data` 函数可以访问相应的 R 包中的数据集。`MASS` 包中包含数据集 `Cars93`，可通过以下方法来访问：

```
data(Cars93, package="MASS")
head(Cars93)
```

```
##   Manufacturer   Model   Type Min.Price Price Max.Price MPG.city
## 1         Acura Integra  Small     12.9   15.9     18.8      25
## 2         Acura  Legend Midsize     29.2   33.9     38.7      18
## 3          Audi    90 Compact     25.9   29.1     32.3      20
## 4          Audi   100 Midsize     30.8   37.7     44.6      19
```

```
## 5      BMW      535i Midsize      23.7  30.0      36.2      22
## 6      Buick Century Midsize      14.2  15.7      17.3      22
##      MPG.highway      AirBags DriveTrain Cylinders EngineSize
## 1      31      None      Front      4      1.8
## 2      25 Driver & Passenger      Front      6      3.2
## 3      26      Driver only      Front      6      2.8
## 4      26 Driver & Passenger      Front      6      2.8
## 5      30      Driver only      Rear      4      3.5
## 6      31      Driver only      Front      4      2.2
##      Horsepower RPM Rev.per.mile Man.trans.avail Fuel.tank.capacity
## 1      140 6300      2890      Yes      13.2
## 2      200 5500      2335      Yes      18.0
## 3      172 5500      2280      Yes      16.9
## 4      172 5500      2535      Yes      21.1
## 5      208 5700      2545      Yes      21.1
## 6      110 5200      2565      No      16.4
##      Passengers Length Wheelbase Width Turn.circle Rear.seat.room
## 1      5      177      102      68      37      26.5
## 2      5      195      115      71      38      30.0
## 3      5      180      102      67      37      28.0
## 4      6      193      106      70      37      31.0
## 5      4      186      109      69      39      27.0
## 6      6      189      105      69      41      28.0
##      Luggage.room Weight Origin      Make
## 1      11      2705 non-USA Acura Integra
## 2      15      3560 non-USA Acura Legend
## 3      14      3375 non-USA Audi 90
## 4      17      3405 non-USA Audi 100
## 5      13      3640 non-USA BMW 535i
## 6      16      2880 USA Buick Century
```

## 读取数据文件

使用 R 语言的时候，如果是少量数据，不妨使用创建向量的语句 `c()` 或其他函数进行创建；但是对于大量数据，最好还是先通过其他更方便的软件创建数据文件，然后使用 R 读入这个文件。`.csv` 是非常好的数据文件格式，跨平台支持非常好。在 Excel 或者 SPSS 中创建的数据，只要存为 csv 格式，就可以使用几乎任何数据处理软件对这些数据进行处理了。使用通用格式在多人合作、不同版本兼容等常见行为中，优势十分明显。另外，之所以使用不同的数据处理软件，第一，可以取长补短。比如有些工作 SPSS 很复杂的，可以用 R 语言几行命令搞定。第二，可以进行软件间处理结果对照，发现问题。R 语言中读取外部文件的最基本函数是 `read.table()`，还有用来读 csv 的 `read.csv()`，`.csv` 是非常好的数据文件格式，跨平台支持非常好。。输入 `help ( read.table )` 命令，就看到了关于数据输入函数的说明。对 `read.table`，使用格式是这样的；

```
read.table(file, header = FALSE, sep = "", quote = "'",
  dec = ".", row.names, col.names,
  as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, nrows = -1,
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#",
  allowEscapes = FALSE, flush = FALSE,
  stringsAsFactors = default.stringsAsFactors(),
  fileEncoding = "", encoding = "unknown", text)
  参数很多，最常用的也就几个，重写如下：
read.table(file, header = FALSE, sep = "", quote = "'",
  dec = ".", skip = 0,
```

```
strip.white = FALSE, blank.lines.skip = TRUE,
comment.char = "#")
```

file 表示要读取的文件。file 可以是 □ 绝对路径或者相对路径，但是一定要注意，在 R 语言中“”是转义符，所以路径分隔符必须写成“\”，比如“C:\myfile\myfile.txt”。□ 可以使剪切板的内容。□ 使用 file.choose()，弹出对话框，让你选择文件位置。强烈推荐使用第三种方法，免去了记忆和书写文件路径的麻烦，特别是能够避免因数据文件位置移动带来的错误！例如：read.table(file.choose(),...)。header 来确定数据文件中第一行是不是标题。默认 F，即认为数据文件没有标题，也即认为第一行就开始是数据了！例如：姓名年龄收入小六 12 350 如果 header=F，读进来的第一行数据是“姓名年龄收入”，header=T，读进来的第一行是“小六 12 350”。sep 指定分隔符，默认是空格。quote 是引号，默认就是双引号。dec 是小数点的表示，默认就是一个点。skip 是确定是否跳过某些行。strip.white 确定是否消除空白字符。blank.lines.skip 确定是否跳过空白行。comment.char 指定用于表示注释的引导符号。

和 read.table 有所不同的，是 read.csv 的默认参数有别。注意看，header 和 sep 的默认值。

```
read.csv(file, header = TRUE, sep = ",", quote="\"", dec=".",
fill = TRUE, comment.char="")
```

因为 csv 就是逗号分割的意思，当然 sep 必须是逗号。header 也是默认有标题的。fill 是默认填充的，即遇到行不相等的情况，空白域自动添加既定值。

举个例子，我们有一个记录金庸和古龙武侠小说人物的文件，分别保存为制表符分隔的 txt 文件和逗号分隔的 csv 文件。用 read.table() 读取 TXT 文件：

```
wuxia <- read.table("wuxia.txt", header = T)
wuxia
```

##	序号	姓名	小说名称
## 1	1	令狐冲	笑傲江湖
## 2	2	张无忌	倚天屠龙记
## 3	3	黄蓉	射雕英雄传
## 4	4	韦小宝	鹿鼎记
## 5	5	李寻欢	多情剑客无情剑
## 6	6	陆小凤	陆小凤传奇
## 7	7	乔峰	天龙八部
## 8	8	小龙女	神雕侠侣
## 9	9	胡一刀	雪山飞狐

用 read.csv() 读取 csv 文件：

```
wuxia2 <- read.csv("wuxia.csv", header = T, sep = ",")
wuxia2
```

##	序号	姓名	小说名称
## 1	1	令狐冲	笑傲江湖
## 2	2	张无忌	倚天屠龙记
## 3	3	黄蓉	射雕英雄传
## 4	4	韦小宝	鹿鼎记
## 5	5	李寻欢	多情剑客无情剑
## 6	6	陆小凤	陆小凤传奇
## 7	7	乔峰	天龙八部
## 8	8	小龙女	神雕侠侣
## 9	9	胡一刀	雪山飞狐

## 读取 Excel 文件

可以将 Excel 文件另存为 csv 格式，然后用 read.csv() 读取。或者安装 xlsx 包（需要安装 Java 虚拟机），使用 read.xlsx2() 直接读取。

```
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Loading required package: xlsxjars
```

```
wuxia3 <- read.xlsx2("wuxia.xlsx", sheetName="Sheet1")  
wuxia3
```

```
##      序号      姓名      小说名称  
## 1      1  令狐冲      笑傲江湖  
## 2      2  张无忌      倚天屠龙记  
## 3      3   黄蓉      射雕英雄传  
## 4      4  韦小宝      鹿鼎记  
## 5      5  李寻欢 多情剑客无情剑  
## 6      6  陆小凤      陆小凤传奇  
## 7      7   乔峰      天龙八部  
## 8      8  小龙女      神雕侠侣  
## 9      9  胡一刀      雪山飞狐
```

使用 openxlsx 包不需要 java 的支持，函数也是 read.xlsx()。

```
library(openxlsx)
```

```
##
```

```
## Attaching package: 'openxlsx'
```

```
## The following objects are masked from 'package:xlsx':
```

```
##
```

```
##      createWorkbook, loadWorkbook, read.xlsx, saveWorkbook,
```

```
##      write.xlsx
```

```
wuxia4 <- read.xlsx("wuxia.xlsx", sheet = 1)  
wuxia4
```

```
##      序号      姓名      小说名称  
## 1      1  令狐冲      笑傲江湖  
## 2      2  张无忌      倚天屠龙记  
## 3      3   黄蓉      射雕英雄传  
## 4      4  韦小宝      鹿鼎记  
## 5      5  李寻欢 多情剑客无情剑  
## 6      6  陆小凤      陆小凤传奇  
## 7      7   乔峰      天龙八部  
## 8      8  小龙女      神雕侠侣  
## 9      9  胡一刀      雪山飞狐
```

还可以将 Excel 内容复制到剪切板，然后使用 read.table("clipboard",header=TRUE,.....) 来读取剪切板的内容；或者使用 RODBC 包读取 ODBC 连接的方式等。

## 保存数据到文件

使用 write.table()、write.csv() 等函数都可以，或者调用 xlsx 包保存到 Excel 文件中。

```
mydata
```

```
## [1] age      gender weight
```

```
## <0 rows> (or 0-length row.names)
```

```
write.table(mydata, "mydata.txt", sep=",")  
library(xlsx)  
write.xlsx2(mydata, "mydata.xlsx")
```