# Ch2.5数据的存储与读取

本章概要

*♢* **数据的存储**

*♢***数据的读取**

**数据文件的读取与写入都是在工作目录中进行的。掌握两个函数：**

* getwd() #查询当前工作目录
* setwd("E:/Rprogram") #设定工作目录

注：R中路径分隔符是一个正斜杠“/”或两个反斜杠“\\”

2.5.1**数据的存储**

1.函数**write.table**()和**write.csv(**):在文件中写入一个对象，并保存为文本文件

**一般用法：**

write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",

eol = "\n", na = "NA", dec = ".", row.names = TRUE,

col.names = TRUE, qmethod = c("escape", "double"),

fileEncoding = "")

write.csv(...)

write.csv2(...)

d<-data.frame(obs=c(1,2,3),treat=c("A","B","C"),weight=c(2.3,NA,9))

d

#保存为简单的文本文件

write.table(d,file="E:/Rprogram/data/foo2.txt") #尝试：不要行名和字符串的双引号

#保存为逗号分隔的文本文件

write.csv(d,file="E:/Rprogram/data/foo3.csv") #用记事本打开

2.函数**save**():将给定对象保存为R数据文件

save(d,file="E:/Rprogram/data/foo4.Rdata")

save.image() #保存工作空间的映像，等价于save(list=ls(all=TRUE),file=".Rdata")

2.5.2 **数据文件的读取**

**1.** **文本文件的读取**

**1）函数read.table():读取表格形式的简单文本文件(即空格隔开的)，建立数据框**

一般用法：read.table(file, header = FALSE, sep = "", quote = "\"'", dec = ".",numerals = c("allow.loss", "warn.loss", "no.loss"), 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, skipNul = FALSE)

变形用法：逗号分隔的数据文件用read.csv（）,分号分隔用read.csv2（），制表符分隔时用read.delim()

**注：数值项被当作数值变量读入，非数值项被当作因子读入，如果header=TRUE，表示数据文件中有表头**

例如：在E:/Rprogram/data /houses.txt中数据文件内容为下，读取数据并建立数据框。

Price Floor Area Room Age Cent.heat

01 52 110 830 5 6.2 no

02 54.75 128.0 710 5 7.5 no

03 57.50 101.0 1000 5 4.2 no

04 57.50 131.0 690 6 8.8 no

05 59.75 93.0 900 5 1.9 yes

> setwd("E:/Rprogram/data")

> HousePrice1<-read.table(file="houses.txt",header=TRUE) #尝试：如果原文本文件中没有行号可以吗？读取后会自动添加

> HousePrice1

Price Floor Area Room Age Cent.heat

01 52.00 110 830 5 6.2 no

02 54.75 128 710 5 7.5 no

…..

05 59.75 93 900 5 1.9 yes

> class(HousePrice1) #查询对象的属性类别

[1] "data.frame"

> class(HousePrice1$Cent.heat)

[1] "factor"

#函数read.csv():读取逗号分隔的文本文件（不使用行号）

> HousePrice2<-read.csv(file="houses.csv",header=TRUE)

> HousePrice2

Price Floor Area Room Age Cent.heat

1 52.00 110 830 5 6.2 no

2 54.75 128 710 5 7.5 no……

5 59.75 93 900 5 1.9 yes

**2）.函数read.fwf():读取固定宽度格式的数据文件**

**例：**从当前工作目录的文件data.txt中读取数据建立数据框，内容如下：

**A1.501.2**

**A1.551.3**

**B1.601.4**

**B1.651.5**

**C1.701.6**

**C1.751.7**

> mydata<-read.fwf("data.txt",widths=c(1,4,3),col.names=c("X","Y","Z"))

Warning message:

In readLines(file, n = thisblock) :

incomplete final line found on 'data.txt'

> mydata

X Y Z

1 A 1.50 1.2

2 A 1.55 1.3

3 B 1.60 1.4

4 B 1.65 1.5

5 C 1.70 1.6

6 C 1.75 1.7

> class(mydata)

[1] "data.frame"

**3）函数scan()：从控制台或文件中读取数据,建立向量或列表**

mydata<-scan() #默认从控制台中读取数值型数据建立向量，等价于scan(what=0)，0可以改为其它数值

mydata

mydata<-scan(what="") #从控制台中读取字符型数据建立向量,尝试what=TRUE,建立逻辑向量

mydata

**例：**从当前工作目录的文件data.dat中读取数据建立列表，内容如下：

M 65 168

M 70 172

F 54 156

F 58 163

注意到：数据文件中无表头（变量名）

> mydata<-scan("data.dat",what=list("",0,0)) #指定有三个变量，第一个字符型，第二个和第三个变量都为数值型

Read 4 records

> mydata

[[1]]

[1] "M" "M" "F" "F"

[[2]]

[1] 65 70 54 58

[[3]]

[1] 168 172 156 163

> mydata<-scan("data.dat",what=list(sex="",weight=0,height=0)) #直接命名列表的对象名称（即变量名称）

Read 4 records

> mydata

$sex

[1] "M" "M" "F" "F"

$weight

[1] 65 70 54 58

$height

[1] 168 172 156 163

> class(mydata) #查询对象的属性类别

[1] "list"

2.Excel数据的读取

方式一：另存转化为文本文件.csv或.txt文件后，分别用read.csv()或read.table()来读取

方式二：利用剪贴板，先复制Excel电子表格中的数据区域到剪贴板中，再使用命令:read.delim(“clipboard”)或read.table(“clipboard”),

> mydata<-read.delim("clipboard") #注意：复制数据时要带上变量名一起复制,因为默认header=TRUE

> mydata

Sex Weight Height

1 M 65 168

2 M 70 172

3 F 54 156

4 F 58 163

> mydata<-read.table("clipboard") #注意：只需复制数据，不能带上变量名一起复制,因为默认header=FALSE

> mydata

V1 V2 V3

1 Sex Weight Height

2 M 65 168

3 M 70 172

4 F 54 156

5 F 58 163

**方式三：使用程序包RODBC**

install.packages("RODBC")

library(RODBC)

z<-odbcConnectExcel("body.xls")

z

foo<-sqlFetch(z,"Sheet1")

foo

close(z)

3.R中数据集的读取

1)R中标准数据集包datasets

* **函数data():**显示内存中全部数据集
* **注：**除标准的数据集包datasets中包含的100多个自带的数据集（通常为数据框和列表），还包括后来library()加载的其它程序包中的数据集

data()

help(AirPassengers) #查看其中AirPassengers 的信息

AirPassengers

**2)专用程序包中数据集**

* **函数data(**datasetname, package=”pkname”**):**加载某个已安装的专用程序包中某个指定的数据集

注1： 显示已经安装的专用程序包中的所有数据集，数据集可以查看到，但没有加载到内存中

data(package="animation")

pollen #显示其中的一个数据集，不成功的因为？？？

注2：加载程序包后，其中的函数可以使用，但其中的数据集仍未载入内存

install.packages("animation")

library("animation")

data() #为什么不能查看到程序包中数据集pollen？？？

注3：函数data()的另一作用：载入专用程序包中的某个数据集（如pollen）

data(pollen,package="animation")

data()

pollen

WEIGHT #显示数据集中的某个变量WEIGHT能行吗？？？

注4：使用载入后的数据集中的变量有两种方式：

#方式一：数据集名$变量名

pollen$WEIGHT

mean(pollen$WEIGHT)

#方式二：attach()

attach(pollen) #挂接数据集成为当前数据集

WEIGHT

mean(WEIGHT)

detach(pollen) #数据集使用完毕后，需将其挂起

4．R格式的数据

R中的数据对象通过save()保存起来后，再使用load()可重新加载。

* 函数save():R中所有对象（向量、矩阵、数据框、列表、函数等）都可以保存起来，文件名以\*.Rdata为后缀
* 函数load():加载已用save()保存的R数据集\*.Rdata

例：将数据集mtcars中的变量mpg和hp生成数据框mtcar2，并保存在文件myR.Rdata中。

help(mtcars)

mtcars

attach(mtcars)

mtcar2<-data.frame(mpg,hp)

mtcar2

save(mtcar2,file="E:/Rprogram/data/myR.Rdata")

detach(mtcars)

quit()

load("E:/Rprogram/data/myR.Rdata")

5.其它统计软件的数据

R中可利用程序包foreign，读取其它统计软件的数据文件,如spss的数据文件kyphosis.sav。

library(foreign)

setwd("E:/Rprogram/data")

rs<-read.spss("kyphosis.sav")

rs #结果是一个列表

as.data.frame(rs) #函数as.data.frame()可将其它对象转化为数据框

本节学习的R函数小结

|  |  |  |
| --- | --- | --- |
| 第2章 2.5 R的数据存储与读取所介绍的R函数 | | |
| 函数 | 功能 | 示例 |
|  |  |  |
|  |  |  |

**作业：练习本节所有命令**