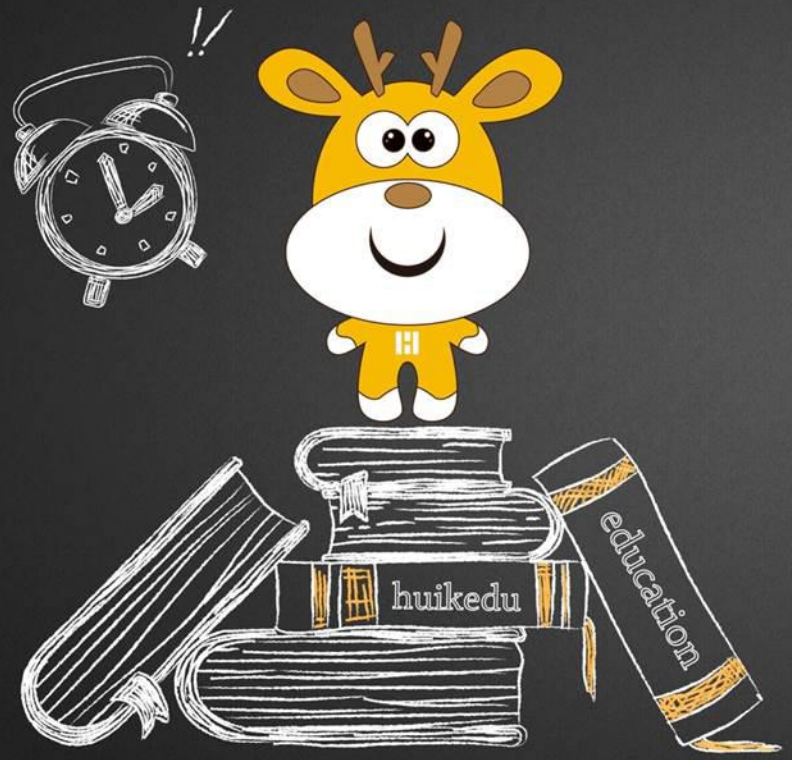
慧科集团
Huikedu Group



《数据分析与R语言》

创建数据集

内容回顾

慧科集团 Huikedu Group

- R的安装
- 熟悉R语言
- 运行R程序

什么是数据集

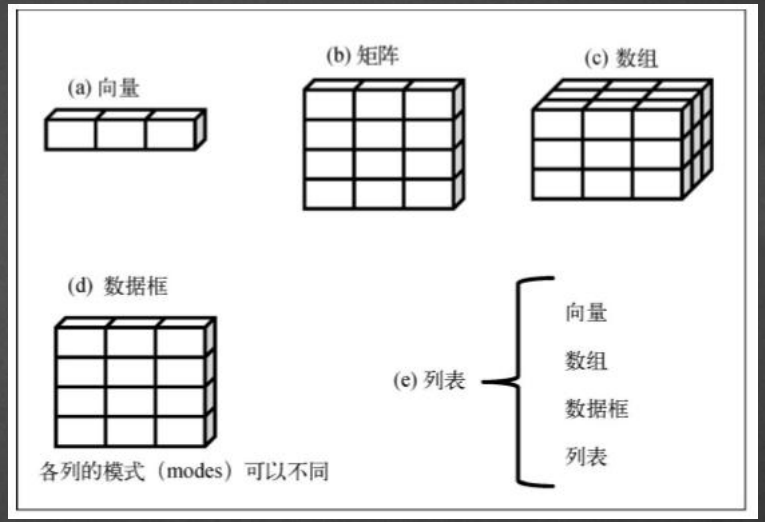
- 数据集通常是由行和列构成的一个矩形数组，行表示观测，列表示变量。

表2-1　病例数据

病人编号（PatientID）	入院时间（AdmDate）	年龄（Age）	糖尿病类型（Diabetes）	病情（Status）
1	10/15/2009	25	Type1	Poor
2	11/01/2009	34	Type2	Improved
3	10/21/2009	28	Type1	Excellent
4	10/28/2009	52	Type1	Poor

数据结构

- 向量
- 矩阵
- 数组
- 数据框
- 列表



向量

慧科集团 Huikedu Group

- 向量是用于存储数值型、字符型或逻辑型数据的一维数组，函数c()可用来 创建向量

```
a <- c(1,2,3,4,5,-6)
b <- c("one","two","three")
c <- c(TRUE,FALSE,TRUE,TRUE)
```

```
#访问向量的元素
a[3]
a[c(1,3,5)]
a[2:6]
```

- 注意：单个向量中的数据必须拥有相同的类型或模式（数值型、字符型或逻辑型）。同一向量中无法混杂不同模式的数据。
- 访问元素：通过在方括号中给定元素所处位置的数值。例如，a[c(2, 4)] 用于访问向量a中的第二个和第四个元素。

矩阵

慧科集团 Huikedu Group

- 矩阵是一个二维数组，只是每个元素都拥有相同的模式（数值型、字符型或逻辑型）。
- 可通过函数matrix创建矩阵。
- 可使用下标和方括号来选择矩阵中的行、列或元素。X[i,]指矩阵X中的第i 行，X[,j] 指第j 列，X[i, j]指第i 行第j 个元素。选择多行或多列时，下标i 和j 可为数值型向量。

```
myymatrix <- matrix(vector, nrow=number_of_rows, ncol=number_of_columns,
                    byrow=logical_value, dimnames=list(
                    char_vector_rownames, char_vector_colnames))
```

数组

慧科集团 Huikedu Group

- 数组（array）与矩阵类似，但是维度可以大于2
- 数组可通过array函数创建
- 数组是矩阵的一个自然推广。在编写新的统计方法时可能很有用。像矩阵一样，数组中的数据也只能拥有一种模式。
- 从数组中选取元素的方式与矩阵相同

```
myarray <- array(vector, dimensions, dimnames)
```

数据框

- 不同的列可以包含不同模式（数值型、字符型等）的数据，是R中常处理的数据结构
- 数据框可通过函数data.frame()创建

```
mydata <- data.frame(col1, col2, col3,…)
```

- 选取数据框中元素的方式有若干种，可以使用前述下标记号，亦可直接指定列名

表2-1　病例数据

病人编号（PatientID）	入院时间（AdmDate）	年龄（Age）	糖尿病类型（Diabetes）	病情（Status）
1	10/15/2009	25	Type1	Poor
2	11/01/2009	34	Type2	Improved

因子

慧科集团 Huikedu Group

- 类别（名义型）变量和有序类别（有序型）变量在R中称为因子（factor）
- 名义型变量是没有顺序之分的类别变量
- 有序型变量表示一种顺序关系，而非数量关系

因子

- 函数factor()以一个整数向量的形式存储类别值，整数的取值范围是[1... k]（其中k 是名义型变量中唯一值的个数），同时一个由字符串（原始值）组成的内部向量将映射到这些整数上。

  diabetes <- c("Type1", "Type2", "Type1", "Type1")

  diabetes <- factor(diabetes)将此向量存储为(1, 2, 1, 1)，并在内部将其关联为 1=Type1和2=Type2

因子

慧科集团 Huikedu Group

- 函

status <- c("Poor", "Improved", "Excellent", "Poor")

语句status <- factor(status, ordered=TRUE)会将向量编码为(3, 2, 1, 3)，并在内部将这些值关联为1=Excellent、2=Improved以及3=Poor

status <- factor(status, order=TRUE,

levels=c("Poor", "Improved", "Excellent"))

因子

慧科集团 Huikedu Group

```
1  > patientID <- c(1, 2, 3, 4)
2
3  > age <- c(25, 34, 28, 52)
4
5  > diabetes <- c("Type1", "Type2", "Type1", "Type1")
6
7  > status <- c("Poor", "Improved", "Excellent", "Poor")
8
9  > patientdata <- data.frame(patientID, age, diabetes, status)
10
11 > patientdata
  patientID age  diabetes     status
1         1  25     Type1       Poor
2         2  34     Type2   Improved
3         3  28     Type1  Excellent
4         4  52     Type1       Poor
```

列表

慧科集团 Huikedu Group

- 一般来说，列表就是一些对象的有序集合
- 列表允许你整合若干对象到单个对象名下。例如，某个列表中可能是若干向量、矩阵、数据框，甚至其他列表的组合
- 可使用函数list()创建列表，对象可以是目前为止讲到的任何结构

```
mylist <- list(object1, object2, …)
```

```
mylist <- list(name1=object1, name2=object2, …)
```

列表

- 可通过在双重方括号中指明代表某个成分的数字或名称来访问列表中的元素

```
> g <- "My First List"
> h <- c(25, 26, 18, 39)
> j <- matrix(1:10, nrow=5)
> k <- c("one", "two", "three")
> mylist <- list(title=g, ages=h, j, k)        <-- 创建列表
> mylist                                       <-- 输出整个列表
$title
[1] "My First List"

$ages
[1] 25 26 18 39

[[3]]
     [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10

[[4]]
[1] "one"   "two"   "three"

> mylist[[2]]                                  <-- 输出第二个成分
[1] 25 26 18 39
> mylist[["ages"]]
[1] 25 26 18 39
```

注意事项

- 对象名称中的句点（.）没有特殊意义
- R不提供多行注释或块注释功能
- 将一个值赋给某个向量、矩阵、数组或列表中一个不存在的元素时，R将自动扩展这个数据结构以容纳新值。

```
> x <- c(8, 6, 4)
> x[7] <- 10
> x
[1]  8  6  4 NA NA NA 10
```

注意事项

- R中的下标不从0开始，而从1开始
- 变量无法被声明，它们在首次被赋值时生成。

基本数据类型

慧科集团 Huikedu Group

- 常见的数据类型

---

1 字符（charactor）
2 数字（numeric）
3 整数（integer）
4 逻辑（logical）
5 复数（complex）

基本数据类型

慧科集团 Huikedu Group

```
#1 字符（charactor）单引号双引号都可以
a <- "哈哈哈哈"
a.a <- "二哈" #起名字字母之间可以有点
is.character(a)#数据类型的判断函数
mode(a)

#2 数字（numeric）实数
b <- 5.7
is.numeric(b)

#3 整数（integer）
c <- 50
is.integer(c)#false，语言bug，混乱性  可转换数据类型
is.numeric(c)
mode(c)
```

基本数据类型

慧科集团 Huikedu Group

#4 逻辑（logical）

```
d <- TRUE
is.logical(d)
```

#5 复数（complex）

```
e <- 5+7i
is.complex(e)
```

数据类型

慧科集团 Huikedu Group

- 注意：两种特殊的数据类型，即数据的缺失 NA 和 空值NULL

数据类型

慧科集团 Huikedu Group

- #特殊的数据类型，即数据的缺失 NA 和 空值NULL

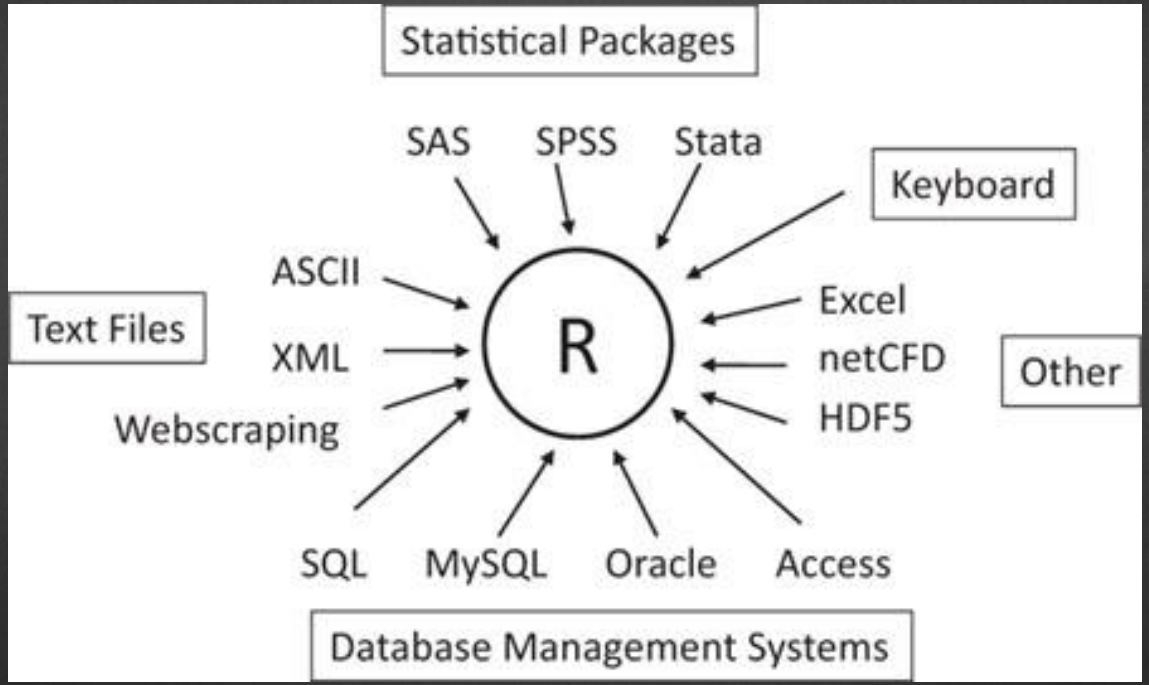
```
#数值缺失
f <- NA
is.na(f)
#空值NULL
g <- c(NA,2,5,NULL)

#数据类型的转化
o <- as.integer(c)
is.integer(o)
```

数据类型

- R常见数据类型的判断与转换函数表

数据类型	中文含义	判断函数	转换函数
character	字符串	is.character( )	as.character( )
numeric	数值	is.numeric( )	as.numeric( )
logical	逻辑	is.logical( )	as.logical( )
complex	复数	is.complex( )	as.complex( )
NA	缺失	is.na( )	as.na( )

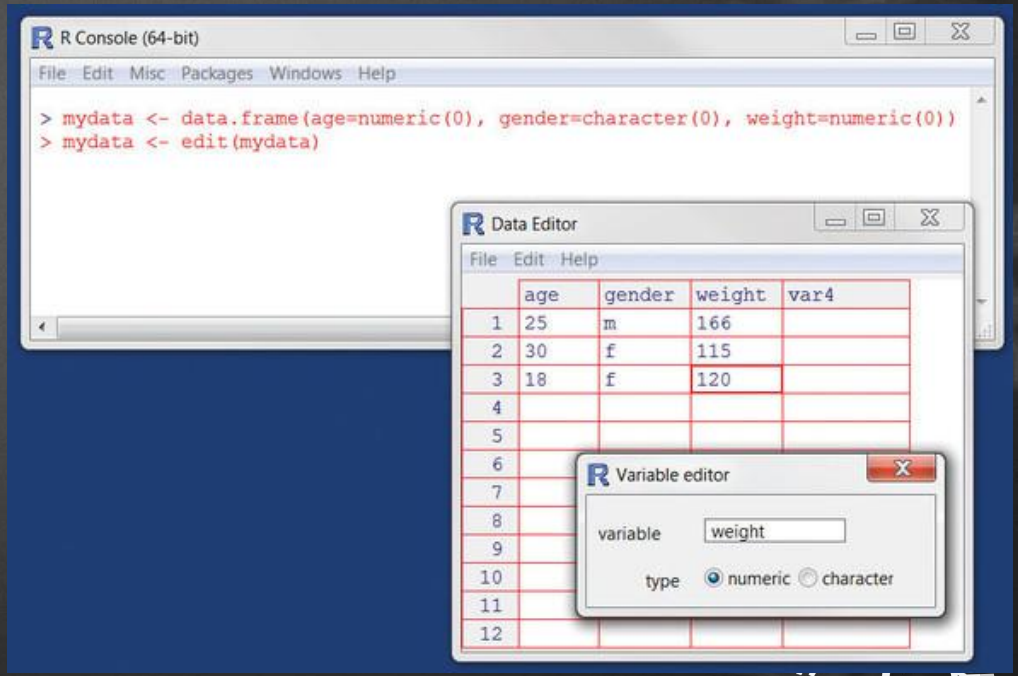
数据的输入


Statistical Packages
SAS
SPSS
Stata
Keyboard
ASCII
Text Files
XML
Webscraping
R
Excel
netCFD
HDF5
Other
SQL
MySQL
Oracle
Access
Database Management Systems

数据的输入

- 使用键盘输入数据

（函数edit()会自动调用一个允许手动输入数据的文本编辑器）



数据的输入

慧科集团 Huikedu Group

- 从带分隔符的文本文件导入数据（使用**read.table()**导入数据）

```
mydataframe <- read.table(file, header=logical_value,
sep="delimiter", row.names="name")
```

例如：

```
grades <- read.table("studentgrades.csv", header=TRUE, sep=",",
row.names="STUDENTID")
```

数据的输入

慧科集团 Huikedu Group

```
#从带分隔符的文本文件导入数据
grades <- read.table("test.csv", header=TRUE, sep=",",
                     row.names="Name")

grades1 <- read.table("test.csv", header=TRUE, sep=","
                      )
grades2 <- read.table("test.csv", header=FALSE, sep=","
)

str(grades) #性别为因子，若不想为因子，可以设置如下

grades <- read.table("test.csv", header=TRUE, sep=",",
                     row.names="Name",stringsAsFactors = FALSE)
str(grades)
```

数据的输入

- 导入Excel数据

将其导出为一个逗号分隔文件（csv）

数据的输入

慧科集团 Huikedu Group

- 导入XML数据

下载安装XML包

```
1 install.packages("XML")
```

导入数据

```
1 library(XML)
2 doc <- xmlRoot(xmlTreeParse("abc.xml"))
3 doc #显示xml数据
```

显示导入数据

```
> doc
<web>
 <servlet>
  <servlet-name>homesystem</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-cla$
 </servlet>
 <servlet-mapping>
  <servlet-name>homesystem</servlet-name>
  <url-pattern>/</url-pattern>
 </servlet-mapping>
</web>
```

数据的输入

慧科集团 Huikedu Group

- 访问数据库管理系统中的数据（RMySQL包）

```
install.packages("RMySQL")
library(RMySQL)

channel <- dbConnect(MySQL(),
                     user="root", #用户名
                     password="123456", #密码
                     dbname="test", #数据库名称
                     host="localhost") #主机地址
dbListTables(channel) #查看数据库中的所有表
```

数据集的标注

- 变量标签（names）

```
names(patientdata)[2] <- "Age at hospitalization (in years)"
```

数据集的标注

- 值标签（lables）

```
patientdata$gender <- factor(patientdata$gender,
                             levels = c(1,2),
                             labels = c("male", "female"))
```

处理数据对象的实用函数

慧科集团 Huikedu Group

表2-3　处理数据对象的实用函数

函　数	用　途
length(*object*)	显示对象中元素/成分的数量
dim(*object*)	显示某个对象的维度
str(*object*)	显示某个对象的结构
class(*object*)	显示某个对象的类或类型
mode(*object*)	显示某个对象的模式
names(*object*)	显示某对象中各成分的名称
c(*object*, *object*,…)	将对象合并入一个向量

处理数据对象的实用函数

慧科集团 Huikedu Group

函　数	用　途
cbind(*object*, *object*, …)	按列合并对象
rbind(*object*, *object*, …)	按行合并对象
Object	输出某个对象
head(*object*)	列出某个对象的开始部分
tail(*object*)	列出某个对象的最后部分
ls()	显示当前的对象列表
rm(*object*, *object*, …)	删除一个或更多个对象。语句rm(list = ls())将删除当前工作环境中的几乎所有对象*
newobject <- edit(*object*)	编辑对象并另存为newobject
fix(*object*)	直接编辑对象

*以句点.开头的隐藏对象将不受影响。——译者注

小结

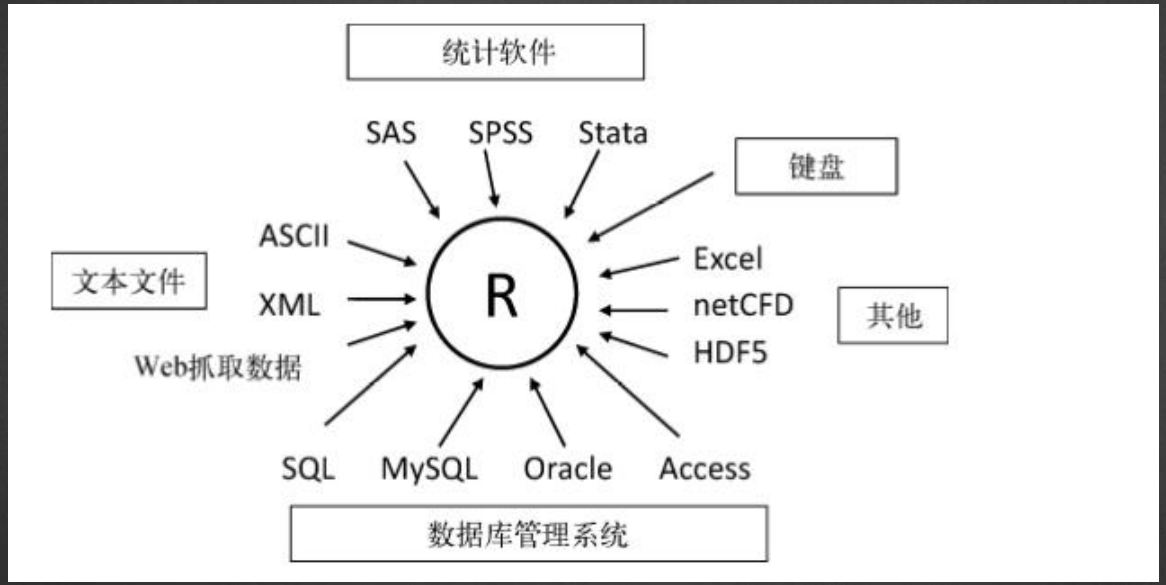
慧科集团 Huikedu Group

- 数据集的概念
- 数据结构
- 数据类型
- 数据的输入

慧科集团
Huikedu Group

Thankyou !

导入数据


统计软件
SAS
SPSS
Stata
键盘
ASCII
文本文件
XML
Web抓取数据
R
Excel
netCFD
HDF5
其他
SQL
MySQL
Oracle
Access
数据库管理系统

使用键盘输入数据

慧科集团 Huikedu Group

- 创建一个空数据框（或矩阵），其中变量名和变量的模式需与理想中的终数据集一致；
- 针对这个数据对象调用文本编辑器，输入你的数据，并将结果保存回此数据对象中。
- 此方式对于小数据集很有效，对于较大的数据集，采用接下来要介绍的方式：从现有的文本文件、Excel电子表格、统计软件或数据库中导入数据

从文本文件导入数据

慧科集团 Huikedu Group

- 从带分隔符的文本文件导入数据
- 带分隔符的文本文件，可以使用read.table()从带分隔符的文本文件中导入数据，此函数可读入一个表格格式的文件并将其保存为一个数据框。

```
mydataframe <- read.table(file, header=logical_value,
  sep="delimiter", row.names="name")
```

导入Excel数据

慧科集团 Huikedu Group

- 读取一个Excel文件的好方式，就是在Excel中将其导出为一个逗号分隔文件（csv），并使用前文描述的方式将其导入R中
- 在Windows系统中，你也可以使用RODBC包来访问Excel文件，电子表格的第一行应当包含变量/列的名称