R中的数据结构

2023年9月8日 13:32

R中的数据类型

数值型: as.numeric()将向量中的其它数据类型转换为数值型

字符串型 as.character()把对象转换成字符型

逻辑型 as.logical()把对象转换成逻辑型

日期型

R中的数据结构

向量

不同于数学中，这里指用于存储数值型、字符型或逻辑型的一维数组 (一个向量中的数据为同一类型)

向量创建:

用c()来创建，字符串需要用引号引出

创建1-100的：c(1:100)或seq(from=1,to=100)

创建1-100，间隔为2:seq(from = 1,to =100,by=2)

创建1-100，一共10个:seq(from = 1,to=100,length,out=10)

创建重复序列:rep()

统计向量中的个数length()

向量索引:

1.正(负整数索引):

x[1]，从1开始

x[1:18]

x[c(1,12,40)]

x[x>5]

x[x>5&&x<9]

2.逻辑向量索引:

```
> z
[1] "one"   "two"   "three" "four"  "five"
> one %in% z
Error in one %in% z : object 'one' not found
> "one" %in% z
[1] TRUE
> z["one" %in% z]
[1] "one"   "two"   "three" "four"  "five"
> z[z %in% c("one","two")]
[1] "one" "two"
> z %in% c("one","two")
[1]  TRUE  TRUE FALSE FALSE FALSE
> k <- z %in% c("one","two")
> z[k]
[1] "one" "two"
```

3.名称索引:

```
> names(y) <- c("one","two","three","four","fiv","six","seven","eight","nine","ten")
> y
  one   two three  four   fiv   six seven eight  nine   ten
    1     2     3     4     5     6     7     8     9    10
> names(y)
 [1] "one"   "two"   "three" "four"  "fiv"   "six"   "seven" "eight" "nine"  "ten"
> euro
      ATS       BEF       DEM       ESP       FIM       FRF       IEP       ITL
 13.76030  40.33990   1.95583 166.38600   5.94573   6.55957   0.78756 1936.27000
      LUF       NLG       PTE
 40.33990   2.20371 200.48200
> euro["ATS"]
 ATS
> y["one"]
one
```

屏幕剪辑的捕获时间: 2023/9/8 17:43

向量修改:

添加向量:

直接添加:

```
v<-c(1,2,3,4)
v[2]<-1
```

用append()

append(x=向量名,values=值,after=第几位后)

向量运算:

向量加减乘除为对每一个标量进行操作

数学运算

**或^幂计算

%%求余运算

%/%求商运算

逻辑运算

%in%左边的是否包含在右边

```
c(1,2,3)%in%c(1,2,2,4,5,6)
```

结果为:TRUE TRUE FALSE

<小于

<=小于等于

>大于

>=大于等于

==等于

!=不等于

x | y x或y

x & y x和y

取平方根sqrt()

取对数log(16,base=2)取以2为底，16的对数

ceiling()向上取整;floor()向下取整

round()四舍五入

signif(x，n)保留n位有效数字

sin(),cos()

sum()

max()

min()

mean()

var()求方差

sd()求标准差

prod()返回连乘的积

median()求中位数

quantile()求分位数

which()查看索引值

```
which.max(c(1,2,3,4))
which(c(1,2,3,4)==4)
```

查看变量类型： mode()

矩阵与数组

矩阵中每个元素的数据类型需要相同

创建矩阵

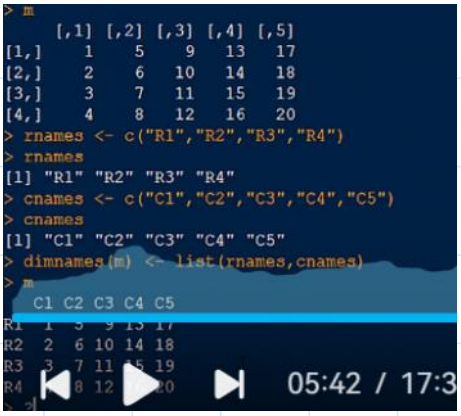
```
x<-c(1:20)
```

matrix(x,nrow=4,ncol=5)

默认按列分布

matrix(x,nrow=4,ncol=5,byrow=TRUE)按行分布

给矩阵添加名字



屏幕剪辑的捕获时间: 2023/9/8 21:31

创建数组

1.x<-c(1:20)

dim(x)<-c(2,2,5)将x排列为5个2行二列矩阵的数组

2.用array()创建

dim1<-c('A1','A2')
dim2<-c('B1','B2','B3')
dim3<-c('C1','C2','C3','C4')
z<-array(1:24,c(2,3,4),dimname=list(dim1,dim2,dim3))

矩阵索引

m<-matrix(1:20,4,5)
m[3,4]
m[1,c(2,3)]

m[2,]取第二行所有

矩阵运算

m<-matrix(1:20,4,5)

rowSums(m)计算每行和

colSums(m)计算每列和

colMeans()
rowMeans()
内积%*%
外积%o%
转置t()
determinant()求行列式

列表

列表是一些对象的有序集合，列表中可以储存若干相同或不同数据类型的数据，形式上类似向量，一维数据集合

创建列表

list()

列表的访问

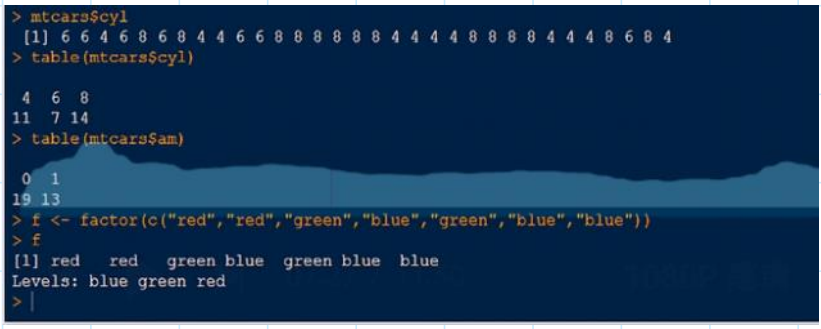
直接索引或用名字或数据框名$名字

单中阔号取子集，双中括号取其中的元素

若要取列表lst中的组件c,有3种方法

lst$c
lst[['c']]

lst[[i]],i是c再lst中的数字编号



★ 数据框

数据框是由数据构成的一个矩形数组，行表示观测，列表示变量

创建数据框

data.frame()

通过索引或名字

as.data.frame()把其它数据类型转换为数据框

因子

变量由三种组成:

名义型变量:如城市名

有序性变量:如好，一般，差

连续变量:1-100

名义型变量和有序性变量的分类可能指成为一个水平，由这些水平值构成的向量称为因子

屏幕剪辑的捕获时间: 2023/9/9 0:34

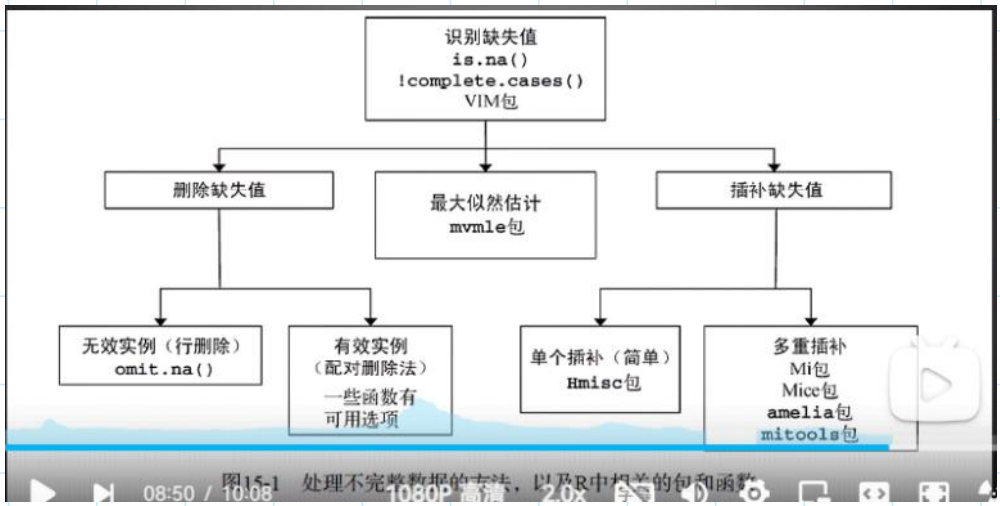
缺失数据

用NA表示缺失值

na.rm=TRUE参数来跳过NA

is.na()判断数据中是否有缺失值

na.omit()去除数据中含缺失值的行



屏幕剪辑的捕获时间: 2023/9/9 14:09

NaN是不存在的值

Inf无穷大或无穷小

字符串

字符串统计

nchar()统计每个元素中的字符串长度，空格算一个字符串

length()统计元素个数

字符串查找

grep()
match()

字符串替换

substr(x=提取的字符串,start=每个字符串提取开始点，stop=每个字符串提取结束点)

toupper()转换为大写

tolower()转换为小写

strsplit()将字符串分隔

```
path<-''user/local/bin'
strsplit(path,'/')
```

字符串连接

paste()将字符串组合，默认用空格分隔

日期和时间

字符串转换为时间序列

as.Date(字符串,format='%Y-%m-%d'),strftime查找所选日期参数