Basic of R

Data Types

Objects

five basic objects

- character
- numeric(实数)
- integer
- complex
- logical(True/False)

向量vector 是最基本的对象但只能包含一种形式的对象。使用vector()创建
列表list 可以包含多种对象

Attributes

R objects can have attributes

- names，dimnames
- dimensions
- class
- length
- metadata
- etc

使用attributes()创建属性

Vectors

c() function to create vectors of objects

```
x<-(0.5,0.6)
```

vector() function to create vectors of objects too

```
>x<-vector("numeric",length=10)
>x
 [1]0000000000
```

implicit coercion:
mix different types of objects, every element of the vector will be transfer to the same class

```
y<-c(1.7,"a")  ##character
y<-c(True,2)  ##numeric
y<-c("a",TRUE)  ##character
```

explicit coercion:
as.* function :transfer one class to another

```
x<-0:6 ##integer
as.numeric(x) ##0123456
as.logical(x) ## FALSE TRUE TRUE TRUE TRUE ...
as.character(x) ## "0" "1" "2" ...
```

List

list can contain elements of different class

```
x<-list(1,"a",TRUE,1+4i)
```

Matrices

a special vector with dimension attribute

```
> m<-matrix(nrow = 2,ncol = 3)
> m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
> dim(m)
[1] 2 3
> attributes(m)
$dim
[1] 2 3
```

all the numbers are inserted into the matrix , by column

```
> m<-matrix(1:6,nrow = 2,ncol = 3)
> m
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

the other ways to create vector

```
> m<-1:10
> m
 [1]  1  2  3  4  5  6  7  8  9 10
> dim(m)<-c(2,5)
> m
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

```
> x<-1:3
> y<-10:12
> cbind(x,y)
     x  y
[1,] 1 10
[2,] 2 11
[3,] 3 12
> rbind(x,y)
  [,1] [,2] [,3]
x    1    2    3
y   10   11   12
```

Factors

factor is a special type of vector, which is used to create, to represent categorical data.

```
> x<-factor(c("yes","yes","no","yes","no"))
> x
[1] yes yes no  yes no
Levels: no yes
> table(x)
x
 no yes
  2   3
```

Missing Values

Missing values in R are denoted by either NA or NAN which we talked about before. NAN is used for undefined mathematical operations. And NA is pretty much used for everything else.

```
is.na() ##test objects if they are NA
is.nan() ##test for NaN
```

Data Frames

data frames are used to store tabular data.
every element has the same length
every element can store different classes

a special attributes

```
row.names
```

generate a data frame

```
> x<-data.frame(foo=1:4,bar=c(T,T,F,F))
> x
  foo   bar
1   1  TRUE
2   2  TRUE
3   3 FALSE
4   4 FALSE
> nrow(x)
[1] 4
> ncol(x)
[1] 2
```

Names

R objects can also have names. This can be very useful for writing readable code and self describing objects.

```
> x<- 1:3
> names(x)
NULL
> names(x) <- c("foo","bar","norf")
> x
 foo  bar norf
   1    2    3
> names(x)
[1] "foo"  "bar"  "norf"
```