R\_Basics

#生成一个正态分布的随机数矩阵100\*100，用于后续使用  
d1<-matrix(rnorm(10000),ncol=10,nrow=1000)  
ncol(d1)

## [1] 10

dim(d1)

## [1] 1000 10

#用字母给每列命名  
colnames(d1)<-LETTERS[1:10]  
#变成data.frame  
d1<-data.frame(d1)  
#查看df的列名  
names(d1)

## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"

#查看df的结构  
str(d1)

## 'data.frame': 1000 obs. of 10 variables:  
## $ A: num -0.424 1.713 -2.182 -0.54 -1.817 ...  
## $ B: num 0.971 -2.163 -0.97 -0.764 1.573 ...  
## $ C: num -0.27 -1.26 1.23 1.11 -1.22 ...  
## $ D: num -0.178 -1.289 -1.728 0.164 0.266 ...  
## $ E: num -0.1425 0.0318 -1.1525 0.8486 -0.2632 ...  
## $ F: num -0.558 0.812 0.875 -0.216 -0.472 ...  
## $ G: num -0.782 -1.473 -0.645 1.197 0.343 ...  
## $ H: num 0.645 -1.527 -1.601 2.327 0.761 ...  
## $ I: num 1.7189 0.0657 -0.2836 0.0579 0.3926 ...  
## $ J: num -0.5036 -0.7641 -0.9302 -0.0676 1.2094 ...

#列出大于2.5的第一列数值的位置  
which(d1$A>2.5)

## [1] 14 54 188 246 430 500 520 804 869

#NA和任何量运算结果都是NA，要特别注意,以第一列的前5行为例  
NA & d1$A[1:5]

## [1] NA NA NA NA NA

#判断是否都满足或者任意满足  
d1$A[1:10]

## [1] -0.42414949 1.71252442 -2.18188101 -0.54025522 -1.81700316  
## [6] 1.67022150 -0.09109736 0.04002646 0.75596368 1.49278811

all(d1$A[1:10]>0)

## [1] FALSE

any(d1$A[1:10]>0)

## [1] TRUE

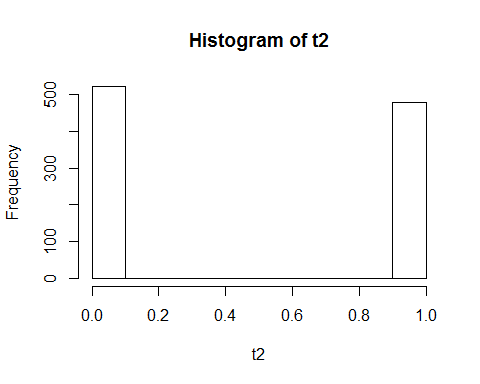
#生成一个长度为1000的空向量，这个技巧用于循环通常大幅提高性能  
t1<-rep(NA,times=1000)  
#整体复制3次和逐个复制3次的区别  
rep(c(1,2,3),times=3) #整体

## [1] 1 2 3 1 2 3 1 2 3

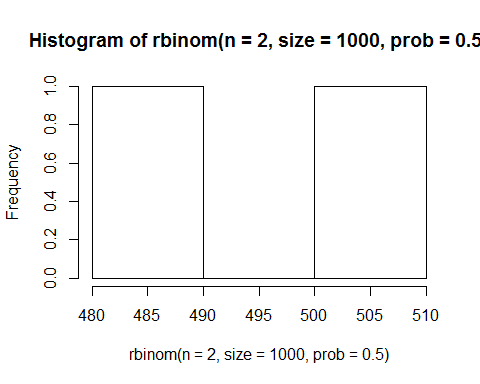
rep(c(1,2,3),each=3) #逐个

## [1] 1 1 1 2 2 2 3 3 3

#replicate是对某个函数运行多少次，如模拟扔硬币  
t2<-replicate(1000,rbinom(n=1,size=1,prob=0.5))  
hist(t2) #0,1频数



hist(rbinom(n=2,size=1000,prob=0.5)) #次数



#对t1进行条件赋值，大于0的t2赋给t1,用ifelse进行向量化操作  
t1<-ifelse(t2>0,t2,NA)  
#判断是否是NA缺失值  
is.na(t1[1:10])

## [1] TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE

#查看不重复的唯一值  
unique(t1)

## [1] NA 1

#直接对有缺失值的向量计数得到的也是NA，所以得选择出非缺失的数值  
sum(t1)

## [1] NA

#注意常见的错误是直接用==或者!=判断是否是NA，这样得不到结果  
t1!=NA

## [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [24] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [47] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [70] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [93] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [116] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [139] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [162] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [185] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [208] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [231] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [254] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [277] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [300] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [323] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [346] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [369] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [392] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [415] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [438] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [461] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [484] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [507] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [530] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [553] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [576] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [599] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [622] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [645] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [668] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [691] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [714] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [737] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [760] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [783] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [806] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [829] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [852] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [875] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [898] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [921] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [944] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [967] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [990] NA NA NA NA NA NA NA NA NA NA NA

sum(t1[t1!=NA])

## [1] NA

#正确做法是用is.na  
sum(t1[!is.na(t1)])

## [1] 478

#放回抽样和不放回抽样  
t3<-1:5  
#不放回抽样  
sample(t3,size=4,replace=FALSE,prob=c(0.2,0.1,0.1,0.3,0.3))

## [1] 3 5 1 2

#放回抽样  
sample(t3,size=4,replace=TRUE,prob=c(0.2,0.1,0.1,0.3,0.3))

## [1] 4 2 2 5

#列出当前所有的对象，返回一个字符向量  
ls()

## [1] "d1" "t1" "t2" "t3"

#把字符向量的内容连接起来  
s1<-c("Hi","this","is","Tom")  
s2<-paste(s1,collapse = "-")  
s2

## [1] "Hi-this-is-Tom"

#有趣的向量连接  
paste(s2,s1,sep="^")

## [1] "Hi-this-is-Tom^Hi" "Hi-this-is-Tom^this" "Hi-this-is-Tom^is"   
## [4] "Hi-this-is-Tom^Tom"

#显示一个对象占用的内存，以字节计算  
object.size(d1)

## 81704 bytes

#把字符变成时间，lubridate包更好，但是一般的as.Date()和as.POSIXlt也还可以  
tt1<-as.Date(Sys.Date())  
tt2<-as.POSIXlt(Sys.Date())  
#从1970-01-01至今的天数  
unclass(tt1)

## [1] 16457

#具体的时间列表  
unclass(tt2)

## $sec  
## [1] 0  
##   
## $min  
## [1] 0  
##   
## $hour  
## [1] 0  
##   
## $mday  
## [1] 22  
##   
## $mon  
## [1] 0  
##   
## $year  
## [1] 115  
##   
## $wday  
## [1] 4  
##   
## $yday  
## [1] 21  
##   
## $isdst  
## [1] 0  
##   
## attr(,"tzone")  
## [1] "UTC"

tt2$year

## [1] 115