

掌握点R语言 | 丕子

* [R for Beginners](#): 一本不错的 R 入门教材,例子比较通俗易懂. (已有中文版)

* [simpleR](#): 我早期看过的一个例子比较多的文档, 个人推荐。

* [Using R for Data Analysis and Graphics - Introduction, Examples and Commentary](#): 写的通俗易懂, 可操作性强, 比 simpleR 略为复杂一点, 个人推荐。

* [An Introduction to R](#): R 官方文档中面向一般用户的, 内容较全, 可能针对性不强, 可以当手册查。 (已有中文版)

* [Practical Regression and Anova using R](#): 用 R 做回归分析, 写得有点涩, 需要一些数学功底。 (该文档有点老)

* [Statistics with R](#): 除了 R 官方的Manual, 这个文档可能是我看到的对 R传统统计方面描述的最全面的一个。

建议

初学者学习 R 语言的文档 (教材) 阅读顺序可以参考如下:

[R for Beginners](#)--> [simpleR](#)/ [Using R](#)--> [An Introduction to R](#)

常用函数参考

<http://www.stathome.cn/manual/s/14.html>

这一节分类列出常用的函数, 需要时可以参看帮助。

基本

一、数据管理

vector : 向量

numeric : 数值型向量

logical : 逻辑型向量

character ; 字符型向量

list : 列表

data.frame : 数据框

c : 连接为向量或列表

length : 求长度

subset : 求子集

seq , from:to , sequence : 等差序列

rep : 重复

NA : 缺失值

NULL : 空对象

sort , order , unique , rev : 排序

unlist : 展平列表

attr , attributes : 对象属性

mode , typeof : 对象存储模式与类型

names : 对象的名字属性

二、字符串处理

character : 字符型向量

nchar : 字符数

substr : 取子串

format , formatC : 把对象用格式转换为字符串

paste , strsplit : 连接或拆分

charmatch , pmatch : 字符串匹配

grep , sub , gsub : 模式匹配与替换

三、复数

complex , Re , Im , Mod , Arg , Conj : 复数函数

四、因子

factor : 因子

codes : 因子的编码

levels : 因子的各水平的名字

nlevels : 因子的水平个数

cut : 把数值型对象分区间转换为因子

table : 交叉频数表

split : 按因子分组

aggregate : 计算各数据子集的概括统计量

tapply : 对“不规则”数组应用函数

数学

一、计算

+, -, *, /, ^, %, %/, %%% : 四则运算

ceiling , floor , round , signif , trunc , zapsmall : 舍入

max , min , pmax , pmin : 最大最小值

range : 最大值和最小值

sum , prod : 向量元素和, 积

cumsum , cumprod , cummax , cummin : 累加、累乘

sort : 排序

approx和approx fun : 插值

diff : 差分

sign : 符号函数

二、数学函数

abs , sqrt : 绝对值, 平方根

log, exp, log10, log2 : 对数与指数函数

sin , cos , tan , asin , acos , atan , atan2 : 三角函数

`sinh` , `cosh` , `tanh` , `asinh` , `acosh` , `atanh` : 双曲函数

`beta` , `lbeta` , `gamma` , `lgamma` , `digamma` , `trigamma` , `tetr`
`agamma` , `pentagamma` , `choose` , `lchoose` : 与贝塔函数、伽玛函
数、组合数有关的特殊函数

`fft` , `mvfft` , `convolve` : 富利叶变换及卷积

`polyroot` : 多项式求根

`poly` : 正交多项式

`spline` , `splinefun` : 样条差值

`besselI` , `besselK` , `besselJ` , `besselY` , `gammaCody` : Bessel函
数

`deriv` : 简单表达式的符号微分或算法微分

三、数组

`array` : 建立数组

`matrix` : 生成矩阵

`data.matrix` : 把数据框转换为数值型矩阵

`lower.tri` : 矩阵的下三角部分

`mat.or.vec` : 生成矩阵或向量

`t` : 矩阵转置

`cbind` : 把列合并为矩阵

`rbind` : 把行合并为矩阵

`diag` : 矩阵对角元素向量或生成对角矩阵

`aperm` : 数组转置

`nrow` , `ncol` : 计算数组的行数和列数

`dim` : 对象的维向量

`dimnames` : 对象的维名

`row/colnames` : 行名或列名

%*% : 矩阵乘法

crossprod : 矩阵交叉乘积 (内积)

outer : 数组外积

kronecker : 数组的

Kronecker积

apply : 对数组的某些维应用函数

tapply : 对 “不规则” 数组应用函数

sweep : 计算数组的概括统计量

aggregate : 计算数据子集的概括统计量

scale : 矩阵标准化

matplot : 对矩阵各列绘图

cor : 相关阵或协差阵

Contrast : 对照矩阵

row : 矩阵的行下标集

col : 求列下标集

四、线性代数

solve : 解线性方程组或求逆

eigen : 矩阵的特征值分解

svd : 矩阵的奇异值分解

backsolve : 解上三角或下三角方程组

chol : Choleski分解

qr : 矩阵的QR分解

chol2inv : 由Choleski分解求逆

五、逻辑运算

< , > , <= , >= , == , != : 比较运算符

!, &, &&, |, ||, xor() : 逻辑运算符

logical : 生成逻辑向量

all, any : 逻辑向量都为真或存在真

ifelse() : 二者择一

match, %in% : 查找

unique : 找出互不相同的元素

which : 找到真值下标集合

duplicated : 找到重复元素

六、优化及求根

optimize, uniroot, polyroot : 一维优化与求根

程序设计

一、控制结构

if, else, ifelse, switch : 分支

for, while, repeat, break, next : 循环

apply, lapply, sapply, tapply, sweep : 替代循环的函数。

二、函数

function : 函数定义

source : 调用文件

call : 函数调用

.C, .Fortran : 调用C或者Fortran子程序的动态链接库。

Recall : 递归调用

browser, debug, trace, traceback : 程序调试

options : 指定系统参数

missing : 判断虚参是否有对应实参

nargs : 参数个数

stop : 终止函数执行

on.exit : 指定退出时执行

eval , expression : 表达式计算

system.time : 表达式计算计时

invisible : 使变量不显示

menu : 选择菜单 (字符列表菜单)

其它与函数有关的还有 : delay , delete.response , deparse , do.call , dput , environment , , formals , format.info , interactive , is.finite , is.function , is.language , is.recursive , match.arg , match.call , match.fun , model.extract , name , parse , substitute , sys.parent , warning , machine。

三、输入输出

cat , print : 显示对象

sink : 输出转向到指定文件

dump , save , dput , write : 输出对象

scan , read.table , load , dget : 读入

四、工作环境

ls , objects : 显示对象列表 rm, remove : 删除对象

q , quit : 退出系统

.First , .Last : 初始运行函数与退出运行函数。

options : 系统选项

? , help , help.start , apropos : 帮助功能

data : 列出数据集

统计计算

一、统计分布

每一种分布有四个函数 :

d—density (密度函数) ,

p—分布函数 ,

q—分位数函数 ,

r—随机数函数。

比如 , 正态分布的这四个函数为dnorm , pnorm , qnorm , rnorm。下面我们列出各分布后缀 , 前面加前缀d、p、q或r就构成函数名 :

norm : 正态 , t : t分布 , f : F分布 , chisq : 卡方 (包括非中心)

unif : 均匀 , exp : 指数 , weibull : 威布尔 , gamma : 伽玛 , beta : 贝塔

lnorm : 对数正态 , logis : 逻辑分布 , cauchy : 柯西 ,

binom : 二项分布 , geom : 几何分布 , hyper : 超几何 , nbinom : 负二项 , pois : 泊松

signrank : 符号秩 , wilcox : 秩和 , tukey : 学生化极差

二、简单统计量

sum, mean, var, sd, min, max, range, median, IQR (四分位间距) 等为统计量 , sort , order , rank与排序有关 , 其它还有ave , five num , mad , quantile , stem等。

三、统计检验

R中已实现的有chisq.test , prop.test , t.test。

四、多元分析

cor , cov.wt , var : 协方差阵及相关阵计算

biplot , biplot.princomp : 多元数据biplot图

cancor : 典则相关 princomp : 主成分分析

hclust : 谱系聚类 kmeans : k-均值聚类

cmdscale : 经典多维标度

其它有dist , mahalanobis , cov.rob。

五、时间序列

ts : 时间序列对象 diff : 计算差分 time : 时间序列的采样时间 w
indow : 时间窗

六、统计模型

lm , glm , aov : 线性模型、广义线性模型、方差分析

=====COS上的字符串操作=====

用于字符串分割的函数：

如

```
a=strsplit('123abcdefgabcdef','ab')
```

```
[[1]]
```

```
[1] "123" "cdefg" "cdef"
```

那么怎么把每一个数据取出来呢？

a[[1]][1]其实是 “123” ，同理取出其他的值。

#字符串连接：

```
paste() #paste(..., sep = " ", collapse = NULL)
```

#字符串分割：

```
strsplit() #strsplit(x, split, extended = TRUE, fixed = FALSE, perl = FALSE)
```

#计算字符串的字符数：

```
nchar()
```

#字符串截取：

```
substr(x, start, stop)
```

```
substring(text, first, last = 1000000)
```

```
substr(x, start, stop) <- value
```

```
substring(text, first, last = 1000000) <- value
```

#####例子说明

```
substr("abcdef",2,4)
substring("abcdef",1:6,1:6)## strsplit is more efficient ...
substr(rep("abcdef",4),1:4,4:5)
x <- c("asfef", "qwerty", "yuiop[", "b", "stuff.blah.yech")
substr(x, 2, 5)substring(x, 2, 4:6)
substring(x, 2) <- c("..", "+++")
```

x

#####

#字符串替换及大小写转换：

chartr(old, new, x)

tolower(x)

toupper(x)

casefold(x, upper = FALSE)

字符完全匹配

grep()

字符不完全匹配

agrep()

字符替换

gsub()

#以上这些函数均可以通过perl=TRUE来使用正则表达式。

```
    grep(pattern, x, ignore.case = FALSE, extended = TRUE,
perl = FALSE, value = FALSE, fixed = FALSE, useBytes = FALSE
)
```

```
sub(pattern, replacement, x,
```

```
ignore.case = FALSE, extended = TRUE, perl = FALSE,
```

```
fixed = FALSE, useBytes = FALSE)
```

```
gsub(pattern, replacement, x,  
ignore.case = FALSE, extended = TRUE, perl = FALSE,  
fixed = FALSE, useBytes = FALSE)  
regexpr(pattern, text, ignore.case = FALSE, extended = TRUE,  
perl = FALSE, fixed = FALSE, useBytes = FALSE)  
gregexpr(pattern, text, ignore.case = FALSE, extended = TRUE,  
perl = FALSE, fixed = FALSE, useBytes = FALSE)
```

See Also:

regular expression(aka 'regexp') for the details of the pattern specification.

'glob2rx' to turn wildcard matches into regular expressions.

'agrep' for approximate matching.

'tolower', 'toupper' and 'chartr' for character translations.

'charmatch', 'pmatch', 'match'. 'apropos' uses regexps and has

nice examples.

暂时就这么多吧。找到以后再粘贴上。

=====

if出现“参数长度为0”的错误：

下面贴几个语句的执行效果，供参考：

```
> if (NULL) cat("OK")
```

```
Error in if (NULL) cat("OK") : argument is of length zero
```

```
> if (TRUE) cat("OK")
```

```
OK> if (TRUE) cat("OK\n")
```

```
OK
```

```
> if (FALSE) cat("OK\n")
```

```
> if (c(TRUE, FALSE)) cat("OK\n")
```

OK

Warning message:

In if (c(TRUE, FALSE)) cat("OK\n") :

the condition has length > 1 and only the first element will be used

矩阵的合并用cbind() rbind()

公式的使用，符号是~。