字符串操作

2023年12月4日 12:17

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()

grep(pattern,x,ignore.case=FALSE,value),返回x字符串向量中含pattern的索引号

ignore.case参数，是否忽略大小写

value参数：是否输出值而不是索引

grepl()用法同grep，返回逻辑值

区分大小写

regexpr()

regexpr(pattern,text)在字符串text中寻找pattern，返回与pattern匹配的第一个子字符串的起始字符位置，如下面这个例子：

```
> regexpr("uat","Equator")
[1] 3
```

gregexpr()

gregexpr(pattern,text)的功能与regexpr()一样，不过它会寻找与pattern匹配的全部子字符串的起始位置，下面是个例子：

```
> gregexpr("iss","Mississippi")
[[1]]
[1] 2 5
```

“Mississippi”中“iss”出现了两次，分别起始于第2个字符和第5个字符的位置。

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)

gsub(pattern, replacement, x, ignore.case = FALSE, perl = FALSE,

fixed = FALSE, useBytes = FALSE)从x中寻找含pattern的并替换为replacement的内容

toupper()转换为大写

tolower()转换为小写

字符串拆分连接

paste()将字符串组合，默认用空格分隔，seq参数指定连接方式

strsplit()将字符串分隔，返回由这些分割后的字符串组成的列表

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

正则表达式

一个任意字符：.(点号)

表示数量：

*任意个数包括零

+任意个数不包括零

？零或1个

{m,n}m到n个

表示位置：

^x以x开头的

x$以x结束的

方括号[],列出出现的集合

[adfg]出现a或d或f或g

或者：|

转义字符：\

\s一个空格

\S一个非空格

\w一个构成文字的字符

\W一个非文字

\d数字

\D非数字

[A-Za-z0-9]*任意数量字母和数字

(?<组名>内容)命名捕获内容的组名

★ R比较特殊一点是用\\表示转义

stringr

字符串处理扩展包：

stringr

字符串连接、重复

1. str_c(data):

用来把多个输入自变量按照元素对应组合为一个字符型向量，用 sep 指定分隔符，默认为不分隔。类似于 R 中向量间运算的一般规则，各自变量长度不同时短的自动循环使用。非字符串类型自动转换为字符型

str_c('a','b')

seq参数指定连接符

collapse参数 输出多个已连接的字符串时，指定多个结果之间的连接符

综合用法：

```
str_c("data", 1:3, ".txt", sep="", collapse=";")

## [1] "data1.txt;data2.txt;data3.txt"
```

2. str_dup(string,times)

类似基础包中的rep()函数，将字符型向量的元素按照times指定的次数在同一字符串内重复

```
str_dup(c("abc", " 长江"), 3)

## [1] "abcabcabc"    " 长江长江长江"
```

也可以针对每个元素指定不同的重复次数

```
str_dup(c("abc", " 长江"), c(3, 2))

## [1] "abcabcabc" " 长江长江"
```

格式化输出

1. format()

将一个数值型向量的各个元素按照统一格式转换为字符型

2. 字符串插值函数,在字符串的内容中插入变量值

a. str_glue()

在字符串内用大括号写变量名，函数可以将字符串中的变量名替换成变量值

seq参数指定分隔符

```
name <- " 李明"
tele <- "13512345678"
str_glue(" 姓名：{name}, ", " 电话号码：{tele}")

## 姓名：李明, 电话号码：13512345678
```

3. 字符串长度

str_length()

```
str_length(c("a", "bc", "def", " 北京"))

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

4. 取子串

str_sub(string,start,end)

```
str_sub(c("term2017", "term2018"), 5, 8)

## [1] "2017" "2018"
```

字符串替换还是用自带的gsub()或str_replace()

gsub(pattern, replacement, x, ignore.case = FALSE, perl = FALSE,

fixed = FALSE, useBytes = FALSE)从x中寻找含pattern的并替换为replacement的内容

5. 字符串变换

基本R中，字符变换表chartr(old,new,x)

```
基本 R 的 chartr(old, new, x) 函数指定一个字符对应关系，旧字符在 old 中，新字符在 new 中，x 是一个
要进行替换的字符型向量。比如，下面的例子把所有! 替换成.，把所有; 替换成,：

chartr("!;", ".,", c("Hi; boy!", "How do you do!"))

## [1] "Hi, boy."       "How do you do."
```

str_trim(string,side)删去字符型向量string每个元素首位空格

side参数'both','left','right'指定删除首尾或开头或末尾

6. 简单匹配与查找

grep()

grep(pattern,x,ignore.case=FALSE,value),返回x字符串向量中含pattern的索引号

ignore.case参数，是否忽略大小写

value参数：是否输出值而不是索引

7. 字符串替换

str_replace_all(string, pattern, replacement)在字符串向量string的每个元素中查找子串pattern，并以replacement替换

pattern支持正则表达

```
str_replace_all(c("New theme", "Old times", "In the present theme"),
  fixed("the"), "**")

## [1] "New **me"          "Old times"         "In ** present **me"
```

8. 字符串拆分

str_split(string,pattern)

string:字符型向量

pattern:将string中的每个元素按分隔符pattern拆分，每个元素拆分为一个字符型向量，结果是一个列表，pattern是正则表达

```
x <- c("11,12", "21,22,23", "31,32,33,34")
res1 <- str_split(x, fixed(","))
res1

## [[1]]
## [1] "11" "12"
##
## [[2]]
## [1] "21" "22" "23"
##
## [[3]]
## [1] "31" "32" "33" "34"
```

文本文件读写

文本文件时内容为普通文字，用换行分割成多行的文件

lines<-readLines('文件名.txt')按行读入为一个字符型向量

writeLines(lines,con='文件名.txt')将lines内容写入文本文件