R language and data analysis: strings

Qiang Shen

Jan.2,2018

Paste

```
paste('Welcome','Qiang')
paste('Welcome','Qiang','and his friends.')
paste0('Christ','mas',sep="") #equivalent to paste0
paste('Welcome','Qiang','and his friends.',sep=' | ')
##vectorization
paste('Welcome',c('Qiang','jack','Lucy'),
c('and his friends.','and his classmates'),sep=" ")
# result<-paste('Welcome',c('Qiang','jack','Lucy'),
# c('and his friends.','and his classmates'),sep=" ")
#   result
#   cat(result,sep='\n')
```

Paste: collapse for vectors.

```
vectorOfText<-c('Hello','everyone','out there',".")
vectorOfText
paste(vectorOfText,collapse="")

paste(c('a','b','c','d'),collapse=" + ")
## "PC1 + PC2 + PC3 + PC4 + PC5 + PC6 + PC7 + PC8
# + PC9 + PC10"
PCs<-paste0('PC',1:100)
PCs
paste(PCs,collapse=" + ")
paste0("PC", 1:100,collapse="+")#strjoin
```

Sprintf

```
"Hello Qiang,the game will start at eight at
Zhaohui campus"
person='Ming';time='six';location='Pingfeng'

paste('Hello ',person,",the game will start at ",
      time,' at ', location,' campus',sep="")

sprintf("Hello %s,the game will start at %s at %s campus",
        person,time,location)

#vectoriztion
person=c('Ming','Jack','Linda','Tracy');time=
  c('six','nine');
location=c('Pingfeng','Zhaohui','Deqing','Keqiao')
sprintf("Hello %s,the game will start at %s at %s campus",
         person,time,location)
```

nchar/toupper/tolower

```
value<-c('gender','general','requirement','ecosystem')
value[nchar(value)>8]
length(value)
nchar("");length("")
DNA<-"ATGCtttACC"
toupper(DNA)
tolower(DNA)
DNA
chartr("Tt", "Uu", DNA)##character translation
chartr("Tt", "UU", DNA)
```

String manipulation

- substr: element extraction
- grep(l): vector search
- (g)regexpr: position search
- (g)sub: element replacement
- strsplit: string split

String manipulation

- **substr**
- grep(l)
- (g)regexpr
- (g)sub
- strsplit

Substr

```
str(substr)
value<-c('a1243_v','b3934_d','c1723_t')
substr(value,start=2,stop=5)
substr(value,2,5)<-'1234'
value
```

String manipulation

- substr
- **grep(l)**
- (g)regexpr
- (g)sub
- strsplit

Grep

```
str(grep) ##pattern is regular expression
grep('A',c('ABC','DEF','XAT','ATZ'))
grep('A',c('ABC','DEF','XAT','ATZ'),value=T)
grepl('A',c('ABC','DEF','XAT','ATZ'))
c('ABC','DEF','XAT','ATZ')[grepl('A',c('ABC','DEF','XAT','A
##regular expression.
grep('^A',c('ABC','DEFA','XAT','ATZ'),value=T)
```

Grep example

```
# only works on Qiang's computer but you can revise
# it accordingly for your purpose.
files<-list.files('~/desktop/test')
# pattern=".*stata[0-9]*_[0-9]*\\.dta$"
files
files[grep("^e.*\\.csv$", files)]
##"\\"escape character(转义字符).
```

Escape character

```
paste('I\'m', 'a', 'big  fan of R')
# cat('Hello','World!')
# text<-  c("Hellow, Adam!", "Hi, Adam!",
# "How are you, Adam.")
# sub(pattern=".*(Adam).*",
# replacement="\\1", text)
```

String manipulation

- substr
- grep(l)
- **(g)regexpr**
- (g)sub
- strsplit

String manipulation

- How many ABs is available for each element in the following vector?

```
vec<-c('ABAAAABBBAAAAB','AAABBBBBBBAABBB',
       'AAAAAAAABBBBBAA','AABBBABBBAABBAB')
```

String manipulation

```
vec<-c('ABAAAABBBAAAAB','AAABBBBBBBBAABBB',
       'AAAAAAAABBBBBAA','AABBBABBBAABBAB')
str(regexpr)
regexpr('AB',vec)
gregexpr('AB',vec)
sapply(gregexpr('AB',vec),length)
```

String manipulation

- substr
- grep(l)
- (g)regexpr
- **(g)sub**
- strsplit

Sub and gsub

```
str(sub)
sub('a','x',c('abcda','bcda','xdg_a','xgh_a'))
gsub('a','x',c('abcda','bcda','xdg_a','xgh_a'))

##blank:space and tab
sub('\\s','|',c('abcd','hello world!','jack Ma'))
sub('_.','_x',c('abcda','bcda','xdg_a','xgh_a'))
```

Practical exmample:revisit the export example

```
library(rmatio)
data(iris)
names(iris)
names(iris)<-sub("\\.", "_", names(iris))
names(iris)
```

Practical exmample:revisit the export example

```
data(iris);names(iris)
sub(".", "_", names(iris))
sub("\\.", "_", names(iris))
names(iris)<-sub(".", "_", fixed = T,names(iris))
names(iris)
sub("[.]", "_", names(iris))

## with pipeline.
library(dplyr);library(rmatio)
iris %>% setNames(sub("[.]", "_", names(.)))
names(iris)
out<-split(iris[,c(1:4)],f=iris$Species)
write.mat(out,'iris_nested.mat')
```

String manipulation

- substr
- grep(l)
- (g)regexpr
- (g)sub
- **strsplit**

Strsplit

How to split the first and last name in the following vector?

```
namelist<-c('Jack Ma','Bruce Lee','Jackie Chan',
            'Jet Lee','Yunfat Chow')
namelist
```

```
[1] "Jack Ma"     "Bruce Lee"   "Jackie Chan" "Jet Lee"
```

Strsplit

```
str(strsplit)
namelist<-c('Jack Ma','Bruce Lee','Jackie Chan',
            'Jet Lee','Yunfat Chow')
y<-strsplit(namelist," ")
y
# sapply(y,'[')[1,]
do.call('rbind',y)
```

String function

- substr
- grep(l)
- (g)regexpr
- (g)sub
- strsplit

Regular expression(正则表达式)

- 12: [ ] \ ^ $. | ? * + ( )
- ^ and $
- ., ?, +, * and {n}
- [a-z],[0-9], [a-zA-Z], [ab|ba] and (ab)
- \d(D:digit) ,\w(W:word) and \s(S:space)

Regular expression

- ^ and $

```
grep('^A',c('ABC','DEF','XAT','ATZ','aBC'))
```

```
[1] 1 4
```

```
grep('C$',c('ABC','DEF','XAT','ATZ','aBC'))
```

```
[1] 1 5
```

Regular expression

X*	0 or more repetitions of X
X+	1 or more repetitions of X
X?	0 or 1 instances of X
X{m}	exactly m instances of X
X{m,}	at least m instances of X
X{m,n}	between m and n (inclusive) instances of X

Regular expression

- .,?,+,*

```
# var<-c('a123','bcda','a456','xdfa','xddga')
# grep('^a',var)
# grep('a$',var)
# sub('^a','',var)
# sub('a$','',var)
var0<-c('aa123b_a','bcdab_bc','a45b6_c',
        'xdafa_d','xgaab_e')
sub('_.','',var0)
sub('_.?','',var0)
grep('aa?b',var0,value=T)
grep('aa+b',var0,value=T)
grep('aa*b',var0,value=T)
```

Regular expression

- [] and | and ()

```
var0<-c('aa123b_a','bcdab_b','ba45b6_c',
        'xdafa_d','xgaab_e')
sub('_[a-e]','',var0)
```

```
[1] "aa123b" "bcdab"  "ba45b6" "xdafa"  "xgaab"
```

```
grep('ab|ba',var0,value=T)
```

```
[1] "bcdab_b"  "ba45b6_c" "xgaab_e"
```

```
grep('[Aa$]',c('ABC','DEF','XAT$','ATZ','aBC'),value=T)
```

```
[1] "ABC"  "XAT$" "ATZ"  "aBC"
```

```
grep("(ab)?c",c("ababc","ac","cde"))
```

Example

```
variable<-c('A1234','A1234M6','A1234X5','A1565',
            'A2456Z4','1245')
var1<-c('data','hi14','history','hi2','thim','hi5')
grep('^h.*[0-9]+$',var1,value = T)


var2<-c(5,9,1,4,5,2)
var3<-rep(1:3,2)
data<-data.frame(var1,var2,var3)
data
data[grep('^hi[0-9]+$',data[,1]),]
grep('^A[0-9]+$', variable, value=TRUE)
variable[grepl('^A.',variable)]
```

Example

```
sub('a$','x',c('abcd','bcda'))
sub('a.c','',c('abcd','bcda'))
var0<-c('aa123b_a','bcdab_b','a45b6_c','xdafa_d',
        'xgaab_e')
sub('a*b','',c('akbcd','dcaaaba'))
sub('a*d','',c('abcd','bcda'))
sub('a.*e','',c('abcde','edcba'))
sub('ba|ab','',c('abcd','cdba'))
sub('[^ab]',"",var0)
```

String manipulation

- substr: element extraction
- grep(l): vector search
- (g)regexpr: position search
- (g)sub: element replacement
- strsplit: string split