正则表达式与Stringr包介绍

目录

正则表达式介绍

正则表达式是一种查找以及字符串替换操作，其常被用于检查文本中是否含有指定的特征词、找出文中匹配特征词的位置、从文本中提取信息，比如：字符串的子串等。

例：strings <- c(
"apple",
"219 733 8965",
"329-293-8753",
"Work: 579-499-7527; Home: 543.355.3679"
)

```
phone <- "([0-9]{2})[- .]([0-9]{3})[- .]([0-9]{4})  “
> grep(phone,strings)
[1] 2 3 4
```

一 匹配字符串

#一个中括号代表匹配一个位置，多个中括号连在一起就可以匹配一个字符串
例：[a]表示匹配字符'a',同理[\]表示匹配字符 '\'
[c][a][t]表示匹配"cat"字符串

#多个字符写入一个括号中，表达一种“或”的关系
例：[aoeiu]就表示匹配'a'或'o'或'i'或'o'或' u'
[0123456789]表示匹配0-9之间的任何一个数
简写：[0123456789]也可以简写为[0-9]
[d-f]表示匹配'd'或'e'或' f'
注意：正则表达式区分大小写！
[D-F]表示匹配的是'D'或'E'或'F
思考：匹配一个年份？
匹配任意一个大小写字符？

一 匹配字符串

#特殊字符'.','^'的用法
'.'表示匹配任何一个字符
'^'表示“非”，就是表示对后面内容的否定，即不包含哪个字符
例：[....]就表示匹配任何一个长度为4的字符串
[^0-9]就表示除了数字之外的任何字符

#特殊字母的用法
d等价为[0-9]
w等价为[^0-9a-zA-Z]
s等价为[.]
例：[0-9][0-9][0-9][0-9]也可写为dddd

思考：想要匹配日期格式，比如2017-05-03？

正则表达式介绍

正则表达式与Stringr包介绍

一 匹配字符串

答案：[0-9][0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]
或者dddd-dd-dd

#花括号内的数字告诉正则表达式我们想要重复的次数
[0-9]{4}-[0-9]{2}-[0-9]{2}
或d{4}-d{2}-d{2}实现
a{2-4}就表示匹配"aa","aaa","aaaa “
a{2,}表示匹配连续次数两次以上的'a',即"aa","aaa","aaaa"...
技巧：匹配双引号内文本：".{0,} “

#简化字符的用法
'*'表示{0,}
'?'表示{0,1}
'+'表示{1，}

二 正则表达式函数

如：text = c("to be", "or not to", "be it is", "a question")
pat = "be"

(1). grep()

```
> grep(pat, text)
[1] 1 3

> grep(pat, text, value=TRUE)
[1] "to be"    "be it is  "
```

(2). grepl()

```
> grepl(pat, text)
[1]  TRUE FALSE  TRUE FALSE
```

二 正则表达式函数

如：text = c("to be", "or not to", "be it is", "a question")
pat = "be “

(3). regexpr()

```
> regexpr(pat, text)
[1]  4 -1  1 -1
```

思考题：1.英文有些单词有两种写法比如“color”，“colour” ，如何匹配？
2.如何匹配任意文本？
3.如何保证双引号中没有双引号了？

三 Stringr包用途

如：爬取网页数据

例1：爬取豆瓣TOP250图书数据，每页含25本书，

第二页网址：https://book.douban.com/top250?start=25

第三页网址：https://book.douban.com/top250?start=50

爬虫代码：library(stringr)

```
for(i=0:9)
{

url<-"https://book.douban.com/top250?start="
num<-as.character(i*25)
url<-str_c(url,num)

pachong(url)##########pachong(url)
}
```

三 Stringr包用途

如：爬取网页数据

例2：> position

```
 [1] "[美] 卡勒德·胡赛尼 / 李继宏 / 上海人民出版社 / 2006-5 / 29.00元"
 [2] "[法] 圣埃克苏佩里 / 马振聘 / 人民文学出版社 / 2003-8 / 22.00元"
 [3] "钱锺书 / 人民文学出版社 / 1991-2 / 19.00"
 [4] "余华 / 南海出版公司 / 1998-5 / 12.00元"
 [5] "[日] 东野圭吾 / 刘姿君 / 南海出版公司 / 2008-9 / 29.80元"
```

需要取每本书的价格组成向量：

```
 bookInfo<-"[法] 圣埃克苏佩里 / 马振聘 / 人民文学出版社 / 2003-8 / 22.00元 “
 > str_split(bookInfo," / ")
 [[1]]
 [1] "[法] 圣埃克苏佩里"
 [2] "马振聘"
 [3] "人民文学出版社"
 [4] "2003-8"
 [5] "22.00元"
```

四 Stringr包函数

1.合并字符串

```
A="hello"
B="world"
> str_c(A,B)
"helloworld "
```

2.计算字符串长度

```
> str_length(c("i","like","programming R",123,res))
[1]  1  4 13  3  3
```

3.截取字符串

```
> str_sub("banana",1,3)
[1] "ban"
```

四 Stringr包函数

4.检测字符串

```
> str_detect("banana","ban")
[1] TRUE

strings<-c("i","like","programming R")
> str_detect(strings,"i")
[1] TRUE TRUE TRUE

# 根据正则表达式检验是否匹配
> str_detect("fruit","[aeiou]")
[1] TRUE
> str_detect("2012-01-02","[0-9]{4}")
[1] TRUE
```

四 Stringr包函数

5.找出匹配的字符串位置

```
strings<-c("i","like","programming R","abc")
> str_locate(strings,"i")
     start end
[1,]     1   1
[2,]     2   2
[3,]     9   9
[4,]    NA  NA
```

6.重复字符串

```
> str_dup("fruit",2)
[1] "fruitfruit"
```

四 Stringr包函数

7.提取匹配的部分与替换匹配的部分

```
strings<-c("i","like","programming R","abc")

> str_extract(strings,"[aoeiu]")
[1] "i" "i" "o" "a"

> str_match(strings,"[aoeiu]")
     [,1]
[1,] "i"
[2,] "i"
[3,] "o"
[4,] "a"

> str_replace(strings,"[aoeiu]","+")
[1] "+"              "l+ke"
[3] "pr+gramming R" "+bc"
```

四 Stringr包函数

8.分割字符串

```
> str_split(bookInfo," / ")
[[1]]
[1] "[法] 圣埃克苏佩里"
[2] "马振聘"
[3] "人民文学出版社"
[4] "2003-8"
[5] "22.00元 “
```

9.加空格和去除空格

```
> str_pad("fruit",10,"both")
[1] "  fruit   "
> str_pad("fruit",10,"right")
[1] "fruit     "

> str_trim(" fruit ")
[1] "fruit"
```

谢谢观看

五 思考题答案

答案：

1. colou{0,1}r表示匹配"color","colour “
   colou?r也可以匹配"color","colour “

2. .{0,}表示匹配任何文本
   解析：因为这是一个通配符 ‘.’ 重复任何次数，由于我们是从0到正无穷，所以即便是一个空文本也可以被匹配到。

3. 可以用“[^”]{0,}“保证找到的双引号里面没有包括其他任何双引号。