网页爬虫利器

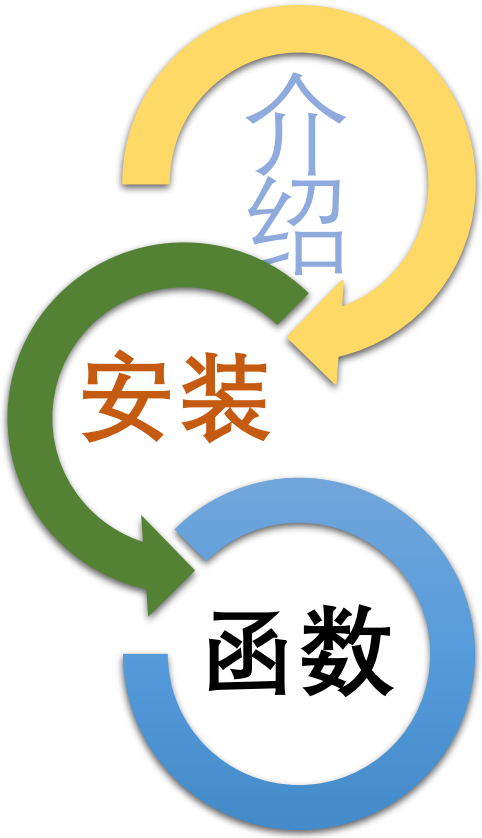
——R-Vest

小组成员：

王萌麟 1600022714

张书萍 1600022721

R-VEST



R-VEST介绍

R-vest 是R语言一个用来做网页数据抓取的包，包的介绍就是“更容易地收割（抓取）网页”。

R-VEST安装

安装rvest包：

```
install.packages("rvest")
```

加载rvest包：

```
library(rvest)
```

R-VEST函数

encoding
html
html_nodes
html_session
Html_text

Jump_to
Pluck
scrape_history
set_values
submit_form
Xml
xml_structure

R-VEST函数

encoding调整字符编码

描述：

这些函数能帮助那些有错误编码声明的网页进行调整.
你可以使用guess_encoding 得到正确编码, 或者用repair_encoding来休整字符型向量

使用:

```
guess_encoding(x)
repair_encoding(x, from = NULL)
```

参数:

参数	说明
x	字符型向量
from	字符的实际编码

R-VEST函数

html：解析HTML网页

什么是html？html是由哪些部分组成？

什么是html标签？

如何查看？

HTML

HTML(Hyper Text Markup Language,超文本标记语言）是WWW中最基本的网页构建语言，用于标注文档或给文档添加标签，使得文档可以在WEB浏览器中显示。

一个HTML文档，通常由3个部分组成：
1.prologue（序）：指出该HTML文档的HTML语言版本号，它是HTML文档开头的一行，用来通知浏览器该文档所遵循的HTML版本。
2.head（头部）：用于存放该HTML文档的属性内容。
3 body（主体）：定义Web页面的具体内容。

HTML

以下表格列出了 HTML head 元素： [3]

标签	描述
<head>	定义了文档的信息
<title>	定义了文档的标题
<base>	定义了页面链接标签的默认链接地址
<link>	定义了一个文档和外部资源之间的关系
<meta>	定义了HTML文档中的元数据
<script>	定义了客户端的脚本文件
<style>	定义了HTML文档的样式文件

HTML

如何查看？

两种方法：
1、打开一个网页，在该网页中点击右键，点击查看源代码，出现一大堆字母就是这个网页的源代码；
2、打开开发者工具也可以看到源代码。

HTML

函数

html(x, ..., encoding = NULL)

参数	说明
x	可以是url，本地路径，包含html的字符串，或者来自httr的请求如果x是URL参数就传递给GET()
Encoding	文档的编码形式,查看iconvlist()有完整列表,他如果不能正确确定encoding方式可以尝试stri_enc_detect

例子

```
1、# From a url:
2、google <- html("http://google.com")
3、google %>% xml_structure()
4、google %>% html_nodes("p")
5、# From a string: (minimal html 5 document)
6、# http://www.brucelawson.co.uk/2010/a-minimal-html5-document/
7、minimal <- html("<!doctype html>
8、<meta charset=utf-8>
9、<title>blah</title>
10、 <p>I'm the content")
11、minimal
12、 # From an httr request
13、 google2 <- html(httr::GET("http://google.com"))
```

R-VEST函数

Html_node()：收集HTML中的节点

描述：

使用XPath和css更简单的分片提取HTML， 结合selectorgadget，能够简单地找到应该使用哪个css selector

函数：

html_nodes(x, css, xpath)

html_node(x, css, xpath)

参数：

参数	说明
x	完整的文档(XMLInternalDocument)，tags列表(XMLNodeSet)，单一的tag(XMLInternalElementNode)
css，xpath	要收集的节点。css和xpath 两种selector方式可选

R-VEST函数

html_text()

从html抽取属性、文本和标签

函数：

```
html_text(x, ···)
html_tag(x)
html_children(x)
html_attrs(x)
html_attr(x, name, default =
NA_character_)
```

R-VEST函数

参数：

参数	说明
x	完整的文档(XMLInternalDocument)，标签列表(XMLNodeSet) 或 (XMLInternalElementNode)
…	其他参数传递给xmlValue().最有用的参数是trim = TRUE,他可以移除空白
name	提取的属性名
default	若任何一个节点的属性不存在则用这里设置的string参数

R-VEST函数

例子

```
movie <- html("http://www.imdb.com/title/tt1490017/")
cast <- html_nodes(movie, "#titleCast span.itemprop")
html_text(cast)
html_tag(cast)
html_attrs(cast)
html_attr(cast, "class")
html_attr(cast, "itemprop")
basic <- html("<p class='a'><b>Bold text</b></p>")
p <- html_node(basic, "p")
p
# Can subset with numbers to extract children
p[[1]]
# Use html_attr to get attributes
html_attr(p, "class")
```

R-VEST函数

小练习：

用rvest包抓取数据时如何查看html？

R-VEST函数

小练习：

使用WebBrowser控件

```
//通过WebBrowser空间访问网页，然后获取网页数据
WebBrowser web = new WebBrowser();
web.Navigate(tbUrl.Text);
web.DocumentCompleted += new
WebBrowserDocumentCompletedEventHandler(web_DocumentCompleted); //当网页加载完成时触发该事件，获取网页数据
void web_DocumentCompleted(object sender, WebBrowserDocumentCompletedEventArgs e)
{
//获取该html页面内的Table标签的内容
WebBrowser web = (WebBrowser)sender;
HtmlElementCollection ElementCollection = web.Document.getElementsByTagName_r("Table");
foreach (HtmlElement item in ElementCollection)
{
textBox1.AppendText(item.InnerText + "\n");
}
}
```

R-VEST函数

xml

与HTML操作相同．目前将XML解码成HTML.

函数:

xml(x, …, encoding = NULL)
xml_tag(x)
xml_attr(x, name, default = NA_character_)
xml_attrs(x)
xml_node(x, css, xpath)
xml_nodes(x, css, xpath)
xml_text(x, …)
xml_children(x)

R-VEST函数

参数:

参数	说明
x	可以是url,本地路径,包含html的字符串,或httr请求的结果
…	如果x是URl,就是GET()的附加参数
name	提取出的属性名
default	当属性名不存在,就用default作为属性名

```
1 search <- xml("http://stackoverflow.com/feeds")
2 entries <- search %>% xml_nodes("entry")
3 entries[[1]] %>% xml_structure()
4 entries %>% xml_node("author name") %>% xml_text()
5 entries %>% lapply(. %>% xml_nodes("category") %>% xml_attr("term"))# 提取category下的term值,
```

R-VEST函数

以豆瓣图书Top250为例，我们只需要以下几行代码就可以实现

R-VEST函数

```
library(rvest) #加载Rvest包

web<- read_html("https://book.douban.com/top250?icn=index-book250-all",encoding="UTF-8") #用read_html函数读取网页信息

position<-web %>% html_nodes ("p.pl") %>% html_text()#获取节点信息
```

用%>%符号进行层级划分。

html_nodes()函数获取网页里的相应节点。

R-VEST函数

原网页里的一个层级结构。
实际上我们要爬取的信息在25个class属性为pl的 标签里的文本

　　[清] 曹雪芹 著 / 人民文学出版社 /
1996-12 / 59.70元

在htmlnodes()函数里的写法就是简单的 "p.pl"，其中“.”表示class属性的值，如果是id属性则用“#”

R-VEST函数

```
position<-web %>% html_nodes("p.pl") %>%
html_text()
#html_text()函数表示获取文本信息，否则返回的是整个<p>标签
```

比较与XML获取节点的方法（如下行代码），其实二者是异曲同工的，只不过将“/”分隔换为了“%>%”，同时个别写法有些许调整。

```
node<-getNodeSet(pagetree,"//p[@class='pl']/text()")
```

R-VEST函数

```
> 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"
 [6] "[日] 村上春树 / 林少华 / 上海译文出版社 / 2001-2 / 18.80元"
 [7] "(日)东野圭吾 / 李盈春 / 南海出版公司 / 2014-5 / 39.50元"
 [8] "[捷克] 米兰·昆德拉 / 许钧 / 上海译文出版社 / 2003-7 / 23.00元"
 [9] "[清] 曹雪芹 著 / 人民文学出版社 / 1996-12 / 59.70元"
[10] "刘慈欣 / 重庆出版社 / 2008-1 / 23.00"
[11] "郭敬明 / 春风文艺出版社 / 2003-11 / 20.00元"
[12] "[美] 丹·布朗 / 朱振武 / 上海人民出版社 / 2004-2 / 28.00元"
[13] "[日] 东野圭吾 / 刘子倩 / 南海出版公司 / 2008-9 / 28.00"
[14] "韩寒 / 国际文化出版公司 / 2010-9 / 25.00元"
[15] "柴静 / 广西师范大学出版社 / 2013-1-1 / 39.80元"
```

R-VEST函数

? html_nodes()

例子

```
ateam <- read_html("http://www.boxofficemojo.com/movies/?id=ateam.htm")

ateam %>% html_nodes("center") %>% html_nodes("td")
ateam %>% html_nodes("center") %>% html_nodes("font")

library(magrittr)
ateam %>% html_nodes("table") %>% extract2(1) %>% html_nodes("img")
ateam %>% html_nodes("table") %>% `[[`(1) %>% html_nodes("img")

ateam %>% html_nodes("table") %>% `[`(1:2) %>% html_nodes("img")
ateam %>% html_nodes("table") %>% extract(1:2) %>% html_nodes("img")
```

R-VEST函数

```
ateam <-
read_html("http://www.boxofficemojo.com/movies/?id=ateam.htm")
```

所有的例子都是基于同一个网站，我们把这个网站存储在ateam变量里

```
ateam %>% html_nodes("center") %>% html_nodes("td")
ateam %>% html_nodes("center") %>% html_nodes("font")
```

获取了ateam这个网页里 标签里<td>的全部内容和 标签里<font>的全部内容

R-VEST函数

运行结果：

```
{xml_nodeset (7)}
[1] <td align="center" colspan="2">\n  <font size="4">Domestic Total Gross:
<b>$77,222, ...
[2] <td valign="top">Distributor: <b><a
href="/studio/chart/?studio=fox.htm">Fox</a></b  ...
[3] <td valign="top">Release Date: <b><nobr><a
href="/schedule/?view=bydate&amp;release  ...
[4] <td valign="top">Genre: <b>Action</b></td>
[5] <td valign="top">Runtime: <b>1 hrs. 57 min.</b></td>
[6] <td valign="top">MPAA Rating: <b>PG-13</b></td>
[7] <td valign="top">Production Budget: <b>$110 million</b></td>

{xml_nodeset (1)}
[1] <font size="4">Domestic Total Gross: <b>$77,222,099</b></font>
```

R-VEST函数

例子中还给出了获取特定序位的html标签的方法，用到了magrittr包里的extract2函数：

```
library(magrittr)
ateam %>% html_nodes("table") %>% extract2(1)
%>% html_nodes("img")
ateam %>% html_nodes("table") %>% `[[`(1)
%>% html_nodes("img")
```

R-VEST函数

上面两行代码都可以获得该网页中第一个<table>标签（由extract2(1)或`[[`(1)获取）中的所有<img>标签里的内容，运行结果如下：

```
{xml_nodeset (6)}
[1] <img src="https://images-na.ssl-images-
amazon.com/images/M/MV5BMTc4ODc4NTQ1N15BMl5B ...
[2] <img src="http://www.assoc-amazon.com/e/ir?t=boxofficemojo-
20&amp;l=as2&amp;o=1&amp ...
[3] <img src="http://www.assoc-amazon.com/e/ir?t=boxofficemojo-
20&amp;l=as2&amp;o=1&amp ...
[4] <img src="/img/misc/bom_logo1.png" width="245" height="56" alt="Box Office Mojo"/>
[5] <img src="/img/misc/IMDbSm.png" width="34" height="16" alt="IMDb"
valign="middle"/>
[6] <img
src="http://b.scorecardresearch.com/p?c1=2&amp;c2=6034961&amp;cv=2.0&amp;cj=1"/>
```

R-VEST函数

可以获得网页里前两个<table>标签储存的所有<img>标签里的内容：

```
ateam %>% html_nodes("table") %>% `[`(1:2) %>%
html_nodes("img")
ateam %>% html_nodes("table") %>% extract(1:2) %>%
html_nodes("img")
```

R-VEST函数

小练习：

1、来自豆瓣的2016—2022值得期待的美国影片

https://www.douban.com/doulist/1276354/?start=0&sort=seq&sub_type=2

R-VEST函数

小练习：

主要代码

```
library(rvest)
url=https://www.douban.com/doulist/1276354/?start=0&sort=seq&sub_type=2#赋值网址
web<-read_html(realurl,encoding="UTF-8") #读取网址
a<-web %>% html_nodes("div.bd.doulist-subject > div.title > a") %>% html_text()
#根据节点获得想要的信息
```

R-VEST函数

小练习：

全部代码：

```
library(rvest)
#library(data.table)
fun<-function(i,url){
#realurl<-paste(url,(i-1)*25)
#realurl<-sub(' ','',realurl)
realurl<-paste(url,(i-1)*25,sep='')
web<-read_html(realurl,encoding="UTF-8")
a<-web %>% html_nodes("div.bd.doulist-subject  > div.title > a") %>% html_text()
#a<-sub('\n        |\n      ','',a)
a<-sub('\n        ','',a)
a<-sub('\n       ','',a)
data.frame(a)
}
final<- data.frame()
url<-'https://www.douban.com/doulist/1276354/?start='
for (i in 1:4){
final<-rbind(final,fun(i,url))
}
```

谢谢！

2017/5/3