使用 R 获得金融数据

石长顺

王亚南经济研究院厦门大学

深圳凌云至善科技有限公司

2016 年 11 月 15 日

1. 爬虫简介

1.1 什么是爬虫

- 网络爬虫，又被称为网页蜘蛛，网络机器人，它是指按照一定的规则，自动抓取万维网 (World Wide Web) 信息的程序或脚本。
- 网络爬虫的实质就是利用抽象化的程序模拟人访问网页的过程。

1.2 为什么要用爬虫

- 时效性
- 可复制性 (可重复性)
- 互联网数据非结构性的特点
- 节约人力、财力

爬虫能做什么？

- 批量下载、上传文件,
- 批量抓取网页内容,
- 模拟登陆 ...

1.3 爬虫需要哪些知识

- HTML
- Xpath
- HTTP
- 其它

1.3.1 爬虫需要那些知识: HTML 语言

1. 元素 (Element): HTML 的基本构成要素，定义了网页的各种对象。

```
<title>First HTML</title>
```

2. 属性 (Attribute): 属性是 HTML 元素提供的附加信息。

```
<img src="https://cran.r-project.org/Rlogo.svg"
alt="W3Schools.com" width="104" height="142">
```

3. CSS: 它主要用于美化网页，比如定义网页的背景颜色、字体的类型等。
4. JavaScript: 实现与用户之间的交互过程。

1.3.1 爬虫需要那些知识: HTML 语言 (Cont'd)

查看网页源代码

- Chrome: 右键 -> 选择 "显示源代码" 或右键 -> 检查 (inspect)
- Fiefox: 右键 -> "View Page Source"

1.3.2 爬虫需要那些知识: XPath 语言

Xpath

XPath 使用路径表达式在 XML 文档中选取节点。Xpath 语法同样适用 HTML 语言。

通俗地将，Xpath 相当于是网页的邮编地址，我们可以利用 XPath 从网页中提取出目标数据。

1.3.2 爬虫需要那些知识: XPath 语言 (Cont'd)

```xml
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book>
  <title lang="en">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="en">Learning XML</title>
  <price>39.95</price>
</book>

</bookstore>
```

- /bookstore/book 返回 bookstore 下所有的书的信息
- /bookstore/book[1] 返回第一本书
- //title[@lang='en'] 返回属性 lang 是 en 的书
- /bookstore/book[price>35.00]/title 返回 price 大于 35 的书的标题

详见 http://www.w3schools.com/xml/xpath_intro.asp

1.3.3 爬虫需要那些知识: HTTP 协议

- HTTP 是互联网上应用最为广泛的一种网络协议。HTTP 协议是基于请求响应模式的，客户端向服务器发送一个请求，服务器则以一个状态行作为响应，响应的内容通常为网页源码。
- HTTP 协议中定义了操作资源的八种不同方法，其中最基本的方法有 4 种：GET，POST，PUT，DELETE，它们分别表示这个资源执行查、改、增、删 4 个操作。

见 http://www.runoob.com/http/http-tutorial.html

1.3.4 编程语言

Figure 1

1.3.5 其他知识

- 正则表达式
- 数据库
- 文件管理
- 编码问题
- 反爬虫绕行

2. R 爬虫工具箱

2.1 R 基础包函数: dowload.file()

1. utils::downlod.file() 是 R 自带从互联网下载文件的函数:

```
download.file(url = "http://www.chinabond.com.cn/Info/24827119",
destfile = "file/2016 年第二期北京首都创业集团有限公司可续期公司债券簿
mode = "wb", quiet = TRUE)
```

download.file 对 https 支持不好，可能会造成下载失败，Rstudio 的Winston Chang写的dowloader可以作为替代。

2.1 R 基础包函数：read.csv()

2. read.csv 等函数，支持从互联网直接读取

```
read.csv(file, ...)
```

这里的 file 可以是一个 url 链接:

```
read.csv("http://chart.finance.yahoo.com/table.csv?s=MSFT&a=9&b=1
    stringsAsFactors = FALSE) %>% head()
```

```
##         Date  Open  High   Low Close   Volume Adj.Close
## 1 2016-11-11 58.23 59.12 58.01 59.02 35553100     59.02
## 2 2016-11-10 60.48 60.49 57.63 58.70 57796200     58.70
## 3 2016-11-09 60.00 60.59 59.20 60.17 48081600     60.17
## 4 2016-11-08 60.55 60.78 60.15 60.47 22862000     60.47
## 5 2016-11-07 59.78 60.52 59.78 60.42 31264400     60.42
## 6 2016-11-04 58.65 59.28 58.52 58.71 28619500     58.71
```

2.1 R 基础包函数：readLines

```
readLines("http://www.baidu.com")[138:145]
```

```
## [1] "<head>"
## [2] "    "
## [3] "   <meta http-equiv=\"content-type\" content=\"text/html;ch
## [4] "   <meta http-equiv=\"X-UA-Compatible\" content=\"IE=Edge\"
## [5] "\t<meta content=\"always\" name=\"referrer\">"
## [6] "    <meta name=\"theme-color\" content=\"#2932e1\">"
## [7] "   <link rel=\"shortcut icon\" href=\"/favicon.ico\" type=\"
## [8] "   <link rel=\"search\" type=\"application/opensearchdescri
```

2.2 RCurl + XML

RCurl 这个程序包提供了由 R 到 libcurl 库的接口，从而实现 HTTP 的一些功能。可以实现保持连接、采用二进制格式读取、句柄重定向、伪装登陆等等。

XML 用于解析 ‘XML‘(‘HTML‘) 文档。

Figure 2: Duncan Temple Lang

2.2 RCurl + XML

使用 RCurl+XML 套件抓取数据的最简单爬虫的步骤: 获取 ft 官网 fastFt 最新的三条新闻 1. 读取网页 (optial):

```
respose <- getURL("https://www.ft.com/fastft")
```

2. 解析网页:

```
pageParsed <- htmlParse(respose) # OR
# pageParsed <- htmlParse("https://www.ft.com/fastft")
```

3. **提取信息**:

```
xpathSApply(doc = pageParsed,'//*[@id="stream"]//h3/a', xmlValue)
```

```
## [1] "China's bid to curb coal-price rise slow to take effect"
## [2] "Gold falls to 5-month low as dollar strengthens"
## [3] "Russneft to float stake on Moscow Exchange by year's end"
```

```
xpathSApply(doc = pageParsed,'//*[@id="stream"]//h3/a',
            xmlGetAttr,"href")%>%head(3)
```

```
## [1] "/content/8354268e-e95d-3109-bb79-2a844fba3ed3"
## [2] "/content/43026cda-032a-338f-8a5a-db00d9f96712"
## [3] "/content/1fa83fb4-8847-373c-9bef-869036c1cfd8"
```

案例一: WISE Paper Keywords

WISE Journalhttp://121.192.176.75/index.php?ser_id=2

```
library(RCurl)
library(XML)
library(stringr)
pageLinks <- str_c("http://121.192.176.75/index.php?ser_id=2&search=&ye
                   seq(0, 240, 20)) # 生成分页地址
# 提取每个分页老师论文的链接
paperlinks <- lapply(pageLinks, getHTMLLinks)%>% unlist()%>%
  str_extract("\\?ser_id=2&p_id=\\d+")%>% na.omit()%>%
  str_c("http://121.192.176.75/index.php",.)
head(paperlinks)
```

```
## [1] "http://121.192.176.75/index.php?ser_id=2&p_id=2349"
## [2] "http://121.192.176.75/index.php?ser_id=2&p_id=2348"
## [3] "http://121.192.176.75/index.php?ser_id=2&p_id=2350"
## [4] "http://121.192.176.75/index.php?ser_id=2&p_id=2262"
## [5] "http://121.192.176.75/index.php?ser_id=2&p_id=2261"
```

案例一: WISE Paper Keywords (Con'd)

抓取论文信息的函数:

```
getPaperInfo <- function(paperlink){
  page_parse <- htmlParse(paperlink,encoding = "utf-8")
  # Get author name
  author <- xpathSApply(page_parse,'//*[@id="paged_list"]/dl/dd[1]/strong',xmlValue)
  # insert & between diff names
  author <- str_replace(author, ',','\\&')
  # paper title
  title <- xpathSApply(page_parse,'//*[@id="paged_list"]/dl/dt', xmlValue)
  # paper url
  journal <- xpathSApply(page_parse,'//*[@id="paged_list"]/dl/dd[2]/i', xmlValue)
  # File name include authoe title journal
  keyWords <- xpathSApply(page_parse,'//*[@id="paged_list"]/dl/dd[6]/text()', xmlValue)
  keyWords <- ifelse(length(keyWords)==0, NA, keyWords)
  updatedDate <- xpathSApply(page_parse,'//*[@id="paged_list"]/dl/dd[3]', xmlValue)%>%
    str_extract("20\\d{6}")
  paperInfo <- data.frame(author, title,journal,keyWords, paperlink,fullTextLink, updatedDate)
  return(paperInfo)
}
```

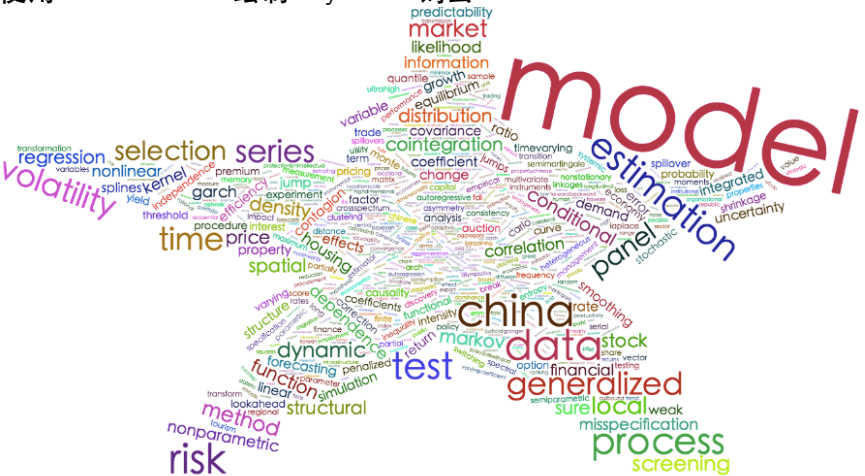
批量抓取:

```
papers.df <- lapply(paperlinks, getPaperInfo)%>%
  do.call("rbind",.)
papers.df <- papers.df[!duplicated(papers.df$title),]
head(papers.df)
```

author	title	journal	keyWords
Mouhua Liao	A Market Game with Symmetric Limit Orders	Journal of Mathematical Economics	Market game, Limit orders, Nash equilibriu
高翔& 龙小宁	省级行政区划造成的文化分割会影响区域经济吗?	经济学（季刊）	文化分割，区域经济，方言
Angelos Dassios& Hongbiao Zha	A Risk Model with Renewal Shot-Noise Cox Process	Insurance: Mathematics and Econom	Risk model; Ruin probability; Renewal sho
Mehmet Caner& Qingliang Fan	Hybrid Generalized Empirical Likelihood Estimators: Ins	Journal of Econometrics	Model selection, Near minimax risk bound
Seong Yeon Chang& Pierre Perr	Inference on A Structural Break in Trend with fractionall	Journal of Time Series Analysis	Fractionally integrated process; linear tren
陈海强& 张传海	股指期货交易会降低股市跳跃风险吗?	经济研究	股指期货；Levy跳跃；非参数方法；Gran
洪永淼	经济统计学与计量经济学等相关学科的关系及发展前景	统计研究	经济统计学、计量经济学、数理经济学、实
龙小宁& 王俊	法治与改革——基于中国法院系统的历史和实证研究	《经济社会体制比较》	NA

Figure 4:

使用 wordcloud2 绘制 keywords 词云



2.3 httr + rvest

httr Hadley 版本的 RCurl。

rvest Hadley 版本的 XML。

RCurl 和 XML 的组合功能强大，但是不够 user-friendly, hadley 大神的这两个包极大的增加了使用的便利性。

常用的 rvest 函数:

- read_html(): 解析网址函数
- html_nodes(doc, css, xpath)：定位函数
- html_text()、html_attrs()、html_table()：提取内容的函数

Figure 5: Hadley Wickham

案例二: 使用 rvest 构建新浪财经 A 股行情查询函数。

实例网页 http://vip.stock.finance.sina.com.cn/corp/go.php/vMS_MarketHistory/stockid/300350.phtml?year=2016&jidu=4

```
library(rvest)
url <- "http://vip.stock.finance.sina.com.cn/corp/go.php/vMS_Mark
tbl <- url%>%read_html()%>%
  html_nodes(xpath = '//*[@id="FundHoldSharesTable"]')%>%
    html_table()%>%`[[` (1)
names(tbl) <- tbl[1,]
tbl <- tbl[-1,]
head(tbl,3)
```

```
##       日期 开盘价 最高价 收盘价 最低价 交易量(股) 交易金额(元)
## 2 2016-11-14 28.470 28.800 28.520 28.330   3954027   113001849
## 3 2016-11-11 28.310 28.600 28.550 28.000   4166197   117641252
## 4 2016-11-10 28.350 28.360 28.310 28.080   3284412    92780917
```

```r
extractTable <- function(url){
  stock <- read_html(url)
  tbl <- stock%>%
    html_nodes(xpath = '//*[@id="FundHoldSharesTable"]')%>%
    html_table()%>% `[[` (1)
  if(length(tbl)>0){
  names(tbl) <- tbl[1,]
  tbl <- tbl[-1,]
  }else{
    tbl <- list() }
  return(tbl)
}
```

Figure 6: extractTable

```r
query_stock <- function(code, years = 2012:2016, jidus = 1:4){
  code <- as.character(code)
  urls = c()
  for(jidu in jidus){
    url = sprintf("http://vip.stock.finance.sina.com.cn/corp/go.php/vMS_MarketHistory/stockid/%s.phtml?year=%s&jidu=%s",
                  code,
                   years,
                   jidu)
    urls = c(urls, url)
  }
  stock <- lapply(urls, extractTable)%>%
    rbindlist()
  names(stock) <- c('Date', 'Open', 'High', 'Close', 'Low', 'Volume', 'Volume_yuan')
  stock$Date <- as.Date(stock$Date)
  stock <- stock[, (names(stock)[-1]):= lapply(.SD, as.numeric), .SDcols = names(stock)[-1]][order(Date)]
  names(stock) <- c("日期", "开盘价","最高价","收盘价","最低价","交易量(股)","交易金额(元)")
  return(stock)
}
```

Figure 7: Query Stock

```r
MaoTai <- query_stock("600519", years = c(2015, 2016), jidus = 1:
head(MaoTai,3)
```

```
##         日期 开盘价 最高价 收盘价 最低价 交易量(股) 交易金额(元)
## 1: 2015-01-05 189.62 204.24 202.52 188.69   9451517  1875063168
## 2: 2015-01-06 200.00 202.56 197.83 196.02   5502001  1094977408
## 3: 2015-01-07 196.04 199.50 192.94 189.99   5479784  1063925632
```

```r
LeSee <- query_stock("300104", years = 2016, jidus = 3:4)
head(LeSee, 3)
```

```
##         日期 开盘价 最高价 收盘价 最低价 交易量(股) 交易金额(元)
## 1: 2016-07-01  52.80  53.96   52.1  51.80  50370206  2648873085
## 2: 2016-07-04  51.80  52.98   52.1  51.53  38379323  2005105466
## 3: 2016-07-05  52.27  52.86   52.3  51.62  30711895  1599846827
```

2.4 抓取 JavaScript 生成的网页

有些网页的数据是通过 JS 异步加载的，这种网页使用上面的方法是不能被解析的。

- R + phantomjs
- R + Selenium(RWebDriver)

2.5 使用 API 获取数据 (以雅虎财经为例)

API：数据提供商为数据使用者提供的使用的接口。

1. 实时行情数据:

   http://download.finance.yahoo.com/d/quotes.csv?s={SYMBOLS}&f=

```
dat <-fread("http://finance.yahoo.com/d/quotes.csv?s=GOOGL&f=ohgp
dat%>%setnames(c("Open", "High", "Low", "Previous_Close"))
dat
```

```
##       Open   High   Low Previous_Close
## 1: 771.76 771.78 745.1         771.75
```

f=ohgp 表示返回 open，high，low，pre_close, 还有很多行情指标详见:Getting Started With the Yahoo Finance API

❷ 雅虎日 (周月) 行情数据 API:

```
http://chart.finance.yahoo.com/table.csv?s={Symbol}&a={StartMonth
&b={StartDayofMonth}&c={StartYear}&d={ToMonth}&e={ToDayof Month}&f
```

```
query_US_stock <- function(Code, startDate = Sys.Date()-365,
                toDate = Sys.Date(), freq="d"){
  require(lubridate)
  startDate <- as.Date(startDate);startMonth <- month(startDate)
  startDay <- day(startDate);startYear <- year(startDate)
  toDate <- as.Date(toDate);toMonth <- month(toDate) -1
toDay <- day(toDate);toYear <- year(toDate)
url <- sprintf("http://chart.finance.yahoo.com/table.csv?s=%s&
      a=%s&b=%s&c=%s&d=%s&e=%s&f=%s&g=%s&ignore=.csv",
Code,startMonth, startDay, startYear, toMonth, toDay, toYear,freq
downloader::download(url, "stock.csv", quiet = TRUE,mode = "wb")
stock <- data.table::fread("stock.csv")
file.remove("stock.csv")
```

```
query_US_stock("BABA",freq = "w")%>%head
```

```
##          Date   Open   High    Low  Close   Volume Adj Close
## 1: 2016-11-07 100.07 100.62  91.10  92.99 18178200     92.99
## 2: 2016-10-31 102.65 104.10  96.46  97.57 19381700     97.57
## 3: 2016-10-24 104.98 105.30 101.55 101.93  8904000    101.93
## 4: 2016-10-17 101.50 104.99 101.27 103.94  9112100    103.94
## 5: 2016-10-10 106.79 109.00  99.00 101.85 16009500    101.85
## 6: 2016-10-03 105.45 107.55 104.98 106.00 10786900    106.00
```

3. Yahoo YQL (Yahoo Query Language) API：以类似 SQL 的方式可以实现强大的查询功能，返回 json 等格式。

```
https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%
20yahoo.finance.quotes%20where%20symbol%20in%20(%22AAPL%22)
&format=json&env=store%3A%2F%2Fdatatables.org%
2Falltableswithkeys&callback=
```

q 后面的参数使用 SQL 的格式呈现如下

```
select * from yahoo.finance.quotes
where symbol in ("AAPL")
```

```
url = "https://query.yahooapis.com/v1/public/yql?q=select%20*%20f

AAPL.json <- read_html(url)%>%
  html_text()
AAPL.ls <- rjson::fromJSON(AAPL.json)
names(AAPL.ls$query$results$quote)%>%head(10) # More Than 80
```

```
##  [1] "symbol"           "Ask"              "AverageDailyVolume"
##  [4] "Bid"                 "AskRealtime"         "BidRealtime"
##  [7] "BookValue"              "Change_PercentChange" "Change"
## [10] "Commission"
```

国内的 API

- 新浪财经
- 腾讯财经
- 通联数据

2.6 获取金融数据的 R 扩展包

- quantmod: getSymbols 系列
- tseries : get.hist.quote
- Quandl
- WindR

3. 其他话题

1. 正则表达式
2. 文件管理
3. 数据存储
4. 编码问题
5. 反爬虫绕行

Thank You!