

爬虫程序对比与前后处理二合一程序

靳军

10153700108

华东师范大学

统计系

2016 年 12 月 12 日

目录

1	Matlab 金融相关爬虫程序详解	3
1.1	关键命令 1: 建立空信息集	3
1.2	关键命令 2: 读取网页	3
1.3	关键命令 3: 网页编码展开	3
1.4	关键命令 4: 指示所需数据	5
1.5	关键命令 5: 数据写入空白矩阵	6
2	R 语言爬虫程序基础编程展示与对比	6
3	R 语言金融专属爬虫程序	7
3.1	quantmod	7
3.2	quantmod+ggplot	9
4	前后处理二合一程序编写	10
4.1	获取数据并求波动率	10
4.2	蒙特卡洛模拟	10
4.3	绘图	11
4.4	定价	11
4.5	结果展示	12
4.6	完整代码展示	13
5	声明	14

1 Matlab 金融相关爬虫程序详解

声明：源文件为作业样例“readsinastu.m”，以下为关键命令详解：

1.1 关键命令 1：建立空信息集

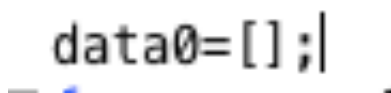
A screenshot of a MATLAB code editor showing the command 'data0=[];' in a blue monospaced font. The code is on a single line, and the background is white with a faint grid.

图 1: 建立初始空信息集

作用：用于储存之后爬下来的数据。

1.2 关键命令 2：读取网页

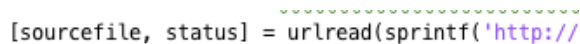
A screenshot of a MATLAB code editor showing the command '[sourcefile, status] = urlread(sprintf('http://'))'. The code is in a blue monospaced font. The background is white with a faint grid.

图 2: 建读取网页数据

示例程序中并不是完整语法,完整语法为:“S = urlread('URL','method',PARAMS)”,其中 URL 指的是网页链接, method 指的是的读取还是写入,通常与默认情况用“get”(读取),第三个则是网页的性质,即编码特点,这里不写选择默认的话则不支持中文录入,因为中文网页编码为 GBK 编码。

1.3 关键命令 3：网页编码展开

A screenshot of a MATLAB code editor showing the command 'disp sourcefile'. The code is in a blue monospaced font. The background is white with a faint grid.

图 3: 网页转换为编码

作用：用以下两幅图可以说明。此命令执行了一次提取网页编码的命令，相当于“检视元素”命令（如下图）。



图 4: 原始网站 (存在检视元素功能)

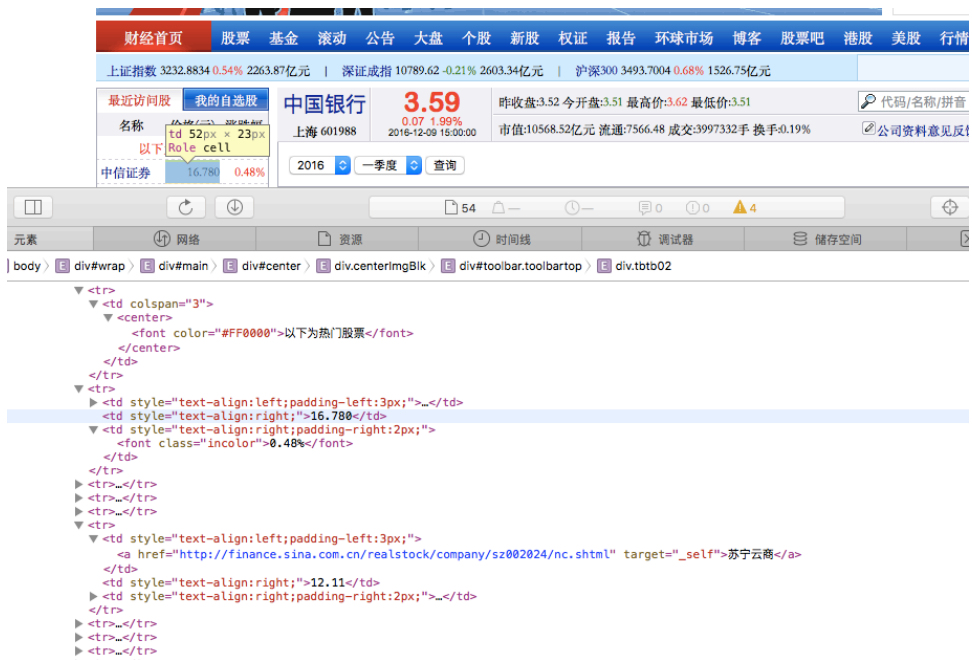


图 5: 导入编码文字 (即 body、div、th 结构)

1.4 关键命令 4：指示所需数据

```
expr2 = '<div align="center">(\d*\.\d*)</div>'; %从源文件中获取目标数据
[datafile, data_tokens] = regexp(sourcefile, expr2, 'match', 'tokens');
```

图 6: 精确指示所需数据位置

作用：指示程序发现指定位置的数据。至于怎么发现的，请看下图

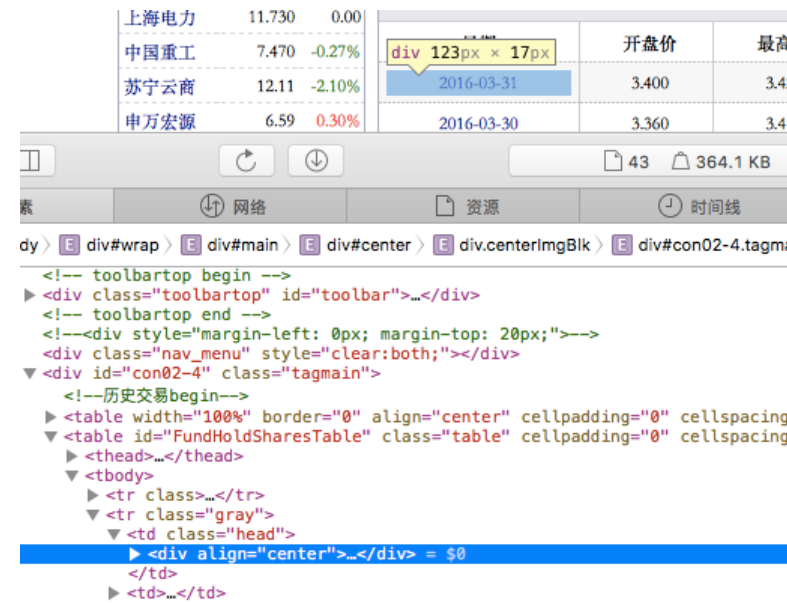


图 7: 精确指示所需数据位置

即先检视所需摘录网页，在网页元素中回溯找寻所需数据位置、前后编码。

先定义表达式形式，这里利用了 `expr` 定义了网页中的制定字符串，然后使用 `regexp` 正则表达式搜索，来确定数据位置。

1.5 关键命令 5：数据写入空白矩阵

```
for idx = 1:length(data_tokens)
    data[idx] = str2double(data_tokens{idx}{1});
```

图 8: 数据写入矩阵

用处：将原有字符串（str）形式的数据变成双精度浮点（double）形式数据，并且写入 data 矩阵。至于 for 循环，只是为了遍历所有数据，不做解释。

2 R 语言爬虫程序基础编程展示与对比

Rcurl 与 XML 为 R 语言基本爬虫包，可以进行几乎全部的爬虫任务，其原理相当本质。下面给出一个例子。

```
library(Rcurl)
library(XML)
res = data.frame()
for (i in 1:10) {
    url = paste("http://sh.lianjia.com/ershoufang/d", i, "s7", sep = "")
    webi = getURL(url, .encoding = "utf-8")
    nodi = getNodeSet(html, path = "//div[@class='list-wrap']/div[@class='info-panel']/a[@name='selectDetail']")
    biaoti = sapply(nodi, function(X) xmlGetAttr(X, "title"))
    Encoding(biaoti) = "UTF-8"
    nodi = getNodeSet(html, path = "//div[@class='list-wrap']/div[@class='where']/span")
    xhmi = sapply(nodi, xmlValue)
    xiaoqu = xhmi[(1:length(biaoti))*3 - 2]
    huxing = gsub("\\s+", "", xhmi[(1:length(biaoti))*3 - 1])
    mianji = as.numeric(gsub("[^0-9]*$", "", xhmi[(1:length(biaoti))*3 - 0]))
    nodi = getNodeSet(html, path = "//div[@class='list-wrap']/div[@class='price']/span[@class='num']")
    zongjia = as.numeric(gsub("[^0-9]", "", sapply(nodi, xmlValue)))
    nodi = getNodeSet(html, path = "//div[@class='list-wrap']/div[@class='price-pre']")
    danjia = as.numeric(gsub("[^0-9]", "", sapply(nodi, xmlValue)))
    res = rbind(res, data.frame(biaoti, lianjie, xiaoqu, huxing, mianji, quyu, diduan, zongjia, danjia, stringsAsFactors = FALSE))
    Sys.sleep(1)
}
```

图 9: Rcurl 与 XML 程序示例

不难发现，虽然一些细节性的语法与 Matlab 不同，但是，大体思路是相似的。下面给出一些同义核心语句：

1. R: paste("网页")+getURL(url) \Leftrightarrow Matlab: [sourcefile, status] = url-read()
2. R: getNodeSet(html, path = "") \Leftrightarrow Matlab: expr = "" + regexp(sourcefile, expr)

最后给出 R 示例运行成果展示：

	biaoti	xiaoqu	huxing	mianji	zongjia	danjia
1	香樟苑（普陀），看房方便，房型正气，简约二室	香樟苑（普陀）	2室1厅	67.83	330	48651
2	中虹华苑，业主信赖，看房有钥匙，成熟社区	中虹华苑	1室1厅	49.97	270	54032
3	成熟社区，新出房源，品质装修，简约二室	白金湾广场（公寓）	2室2厅	127.68	1350	105733
4	人气房源，采光棒，满五税费低，新上房源	丰庄十二街坊	2室1厅	47.47	230	48451
5	满五税费低，配套完善，高清实拍，真实在售	石泉一村	1室0厅	34.79	210	60362
6	链家好房，正规成熟小区，抢鲜笋盘，简约二室	仁德路67弄10支弄	2室1厅	67.41	410	60821
7	梅园六街坊，空气清新，人气房源，真实在售	梅园六街坊	2室1厅	51.59	550	106609
8	人气房源，成熟社区，业主信赖，3房出售	君怡公寓	3室2厅	140.29	720	51322
9	古美八村，钥匙在店，新出房源，一链倾城	古美八村	1室1厅	43.91	280	63766
10	上门实拍，新出好房，如您所见，实地看房	虹二小区	1室0厅	36.72	285	77614
11	开鲁四村，上门实勘，满五年少税，如您所见	开鲁四村	2室1厅	57.84	325	56189
12	受欢迎好房，改善住房，好楼层，业主信赖	仁恒河滨城	3室2厅	151.44	1365	90134

图 10: 运行结果

3 R 语言金融专属爬虫程序

3.1 quantmod

所幸的是，R 语言是开源语言，有前辈已经发明了金融项目的专属 R 包，将繁琐的爬虫语句封锁在一行或几行简易的语句里，大大方便了使用者。下面给出简易的代码：

```
library(quantmod)
getSymbols("601988.ss")
```

图 11: 十分简易的语句

运行结果：

	601988.SS.Open	601988.SS.High	601988.SS.Low	601988.SS.Close	601988.SS.Volume	601988.SS.Adjusted
2007-01-01	4.85310	5.25995	4.83373	5.25995	0	3.61391
2007-01-02	4.85310	5.25995	4.83373	5.25995	0	3.61391
2007-01-03	4.85310	5.25995	4.83373	5.25995	0	3.61391
2007-01-04	5.51180	5.78303	5.20182	5.45368	751673900	3.74701
2007-01-05	5.13402	5.17276	4.91122	4.91122	812165700	3.37431
2007-01-08	4.71748	4.97903	4.67874	4.92091	574569700	3.38097
2007-01-09	4.90153	5.02746	4.79498	5.01778	447459100	3.44752
2007-01-10	5.08558	5.12433	4.89185	4.94028	384147600	3.39427
2007-01-11	4.91122	4.91122	4.74654	4.77560	337987500	3.28113
2007-01-12	4.72717	4.81435	4.55281	4.58187	240448100	3.14802
2007-01-15	4.56249	4.84341	4.50437	4.83373	256774000	3.32107
2007-01-16	4.89185	4.94028	4.69811	4.78529	267985100	3.28779
2007-01-17	4.78529	4.88216	4.66905	4.72717	244920700	3.24785
2007-01-18	4.67874	4.73686	4.50437	4.61093	210253200	3.16799
2007-01-19	4.63030	4.60811	4.66740	4.60811	187487400	3.15780

图 12: 运行结果

可见运行结果和 Matlab 程序运行结果一样，而且数据更加齐全。日开盘价、最高价、最低价、收盘价、成交量，以及 adjusted price，十分详细。除此以外还可以添加一些其他指标并作图，例如 Volume、MACD。只需使用 chartSeries () 函数，下给出运行示例（Matlab 程序做的是中国银行的，这个也做中国银行的好了）：



图 13: 运行结果

3.2 quantmod+ggplot

ggplot 作为可视化的代表，近年来风头正盛，其针对时间序列的展示确实效果极佳。以上证指数为例给出代码：

```
1 library(quantmod)
2 library(ggplot2)
3 getSymbols('^SSEC',src='yahoo',from = '1997-01-01')
4 close=Cl(SSEC)
5 time=index(close)
6 value= as.vector(close)
7 data=data.frame(time,value)
8 ggplot(data,aes(time,value))+ geom_line()
9
```

图 14: ggplot 时间序列代码

其中数据传递一定要是 data 类型的数据。

结果呈现：

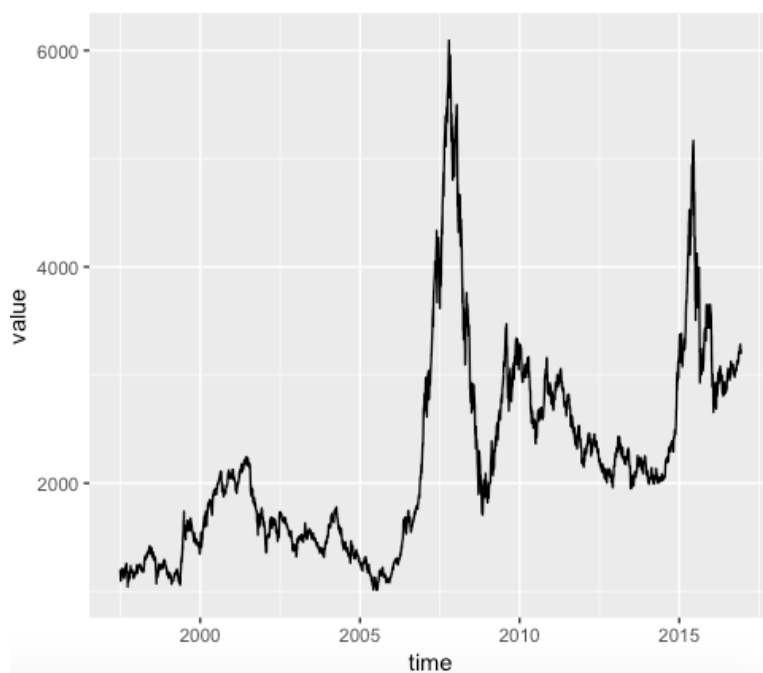


图 15: ggplot 时间序列呈现

其实功能还有很多，在此不一一呈现。

4 前后处理二合一程序编写

既然已经把数据获取的程序缩短，并且已经在 environment 内直接生成了矩阵，那么把前后处理并作一个程序便成为了可能。现在要做的是：

1. 根据所的矩阵在 R 环境内求得波动率
2. 在 R 环境内使用所得波动率进行蒙特卡洛模拟

下面，仿照作业样本“作业样本”，我们一股票代码为 601288 的农业银行使用 R 语言进行一站式定价：

4.1 获取数据并求波动率

```
4 getSymbols("601288.ss")
5 ZGYH=as.matrix(`601288.SS`) #将中国银行表格数据转化为矩阵
6 Daily=as.matrix(dailyReturn(`601288.SS`)) #计算日收益率
7 sigma=sd(Daily)*(252)^0.5#求波动率
```

图 16: 获取数据求波动率

4.2 蒙特卡洛模拟

```
SPaths = matrix(NA,NReps,NSteps+1);
SPaths[,1] = S0;
dt = T/NSteps
nudt = (r-0.5*sigma^2)*dt
sidt = sigma*sqrt(dt)
for (i in 1:NReps){
  for (j in 1:NSteps){
    SPaths[i,j+1] = SPaths[i,j]*exp(nudt + sidt*rnorm(1))
  }
}
}#到这步数值模拟已经结束
```

图 17: 获取数据求波动率

核心语句：rnorm (1)。。。。

4.3 绘图

```

step=c(1:(NSteps+1))
data=data.frame(step)#创建data数据
for (k in 1:NReps){
  data=data.frame(data,SPaths[k,])
}
dfidfm=melt(data,id.vars="step")#熔开data
ggplot(dfidfm,aes(x=step,y=valu))+geom_line(aes(color=variable))#绘图

```

图 18: 形成 data 结构使用 ggplot

注意：一定要 data 结构，不可省略。

4.4 定价

```

payoff=matrix(0,NReps,1)
for (m in 1:NReps){
  ax=SPaths[m,];
  if (min(ax)<sb)
    payoff[m]=0
  else
    payoff[m]=max(0,K-ax[NSteps])
}
P=mean(exp(-r*T)*payoff)
Vector[1,n]=P

```

图 19: 毫无新意的定价

4.5 结果展示



图 20: 定价线路模拟

Average	0.397163749094592
---------	-------------------

图 21: 最终确定价格

4.6 完整代码展示

```

4  getSymbols("601288.ss")
5  ZGYH=as.matrix(`601288.SS`) #将中国银行表格数据转化为矩阵
6  Daily=as.matrix(dailyReturn(`601288.SS`)) #计算日收益率
7  sigma=sd(Daily)*(252)^0.5#求波动率
8  S0=2.6;#初始价格
9  r=0.0284;#无风险收益率
10 T=5/12;#设定时间
11 NSteps=700; #每支模拟步数
12 NReps=20;#平行支数
13 Recycles=15;#重复模拟次数
14 sb=1;#障碍水平
15 K=4;#敲定价格
16 Vector=matrix(0,1,Recycles)
17 for (n in 1:Recycles){
18   SPaths = matrix(NA,NReps,NSteps+1);
19   SPaths[,1] = S0;
20   dt = T/NSteps
21   nudt = (r-0.5*sigma^2)*dt
22   sidt = sigma*sqrt(dt)
23   for (i in 1:NReps){
24     for (j in 1:NSteps){
25       SPaths[i,j+1] = SPaths[i,j]*exp(nudt + sidt*rnorm(1))
26     }
27   }#到这步数值模拟已经结束
28   step=c(1:(NSteps+1))
29   data=data.frame(step)#创建data数据
30   for (k in 1:NReps){
31     data=data.frame(data,SPaths[k,])
32   }
33   dfidfm=melt(data,id.vars="step")#熔开data
34   ggplot(dfidfm,aes(x=step,y=value))+geom_line(aes(color=variable))#绘图
35   payoff=matrix(0,NReps,1)
36   for (m in 1:NReps){
37     ax=SPaths[i,];
38     if (min(ax)<sb)
39       payoff[i]=0
40     else
41       payoff[i]=max(0,K-ax[NSteps])
42   }
43   P=mean(exp(-r*T)*payoff)
44   Vector[1,n]=P
45 }
46 Average=mean(Vector)
47 #Average即是定价。

```

图 22: 获取数据——运算分析——后处理展示一站式编程

5 声明

此作品为本人原创，未经允许，请勿转载。