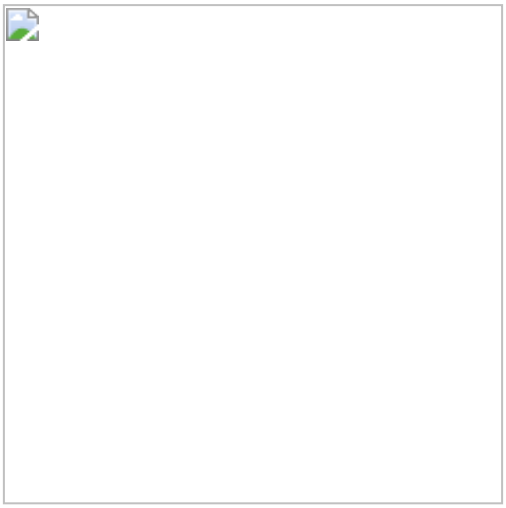
노트북: blog
만든 날짜: 2020-10-07 오후 4:10
URL: https://continuous-development.tistory.com/57?category=793392

R

[R] R을 활용한 크롤링 - 로또 1등 당첨 배출점 크롤링 하기

2020. 8. 7. 17:50 수정 삭제 공개

해당 사이트의 배출점을 크롤링하겠다. 단순 페이지 크롤링에서 스크립트 기능까지 사용하는 크롤링 까지 하겠다.



크롤링을 하기 위해서는 첫번째로 내가 원하는 데이터의 위치를 알아야 한다.
저 배출점의 데이터 얻기 위해서는 개발자 도구(F12)에서 해당 부분을 클릭하면 아래와 같이 나온다. 이제 이 부분을 크롤링하기 위해 준비하자.

```
------------------------ -->
▶<div class="tab_article tab_size5">…</div>
▼<div class="group_content group_data_search">
  ▶<div class="group_title">…</div>
  ▼<table class="tbl_data tbl_data_col">
     <caption>상호명, 배출건수, 소재지, 위치 등 로또6/45 1등 배출점 정
     보 안내</caption>
    ▶<colgroup>…</colgroup>
    ▶<thead>…</thead>
    ▼<tbody>
      ▼<tr>
         <td>1</td>
...      <td>
                                          스파

                                        </td> == $0
         <td>35</td>
         <td>서울 노원구 상계8동  666-3 주공10단지종합상가111</td>
        ▶<td>…</td>
       </tr>
```

#html 소스 가져오기

```r
#크롤링

install.packages("rvest")
library(rvest)

url <- "https://dhlottery.co.kr/store.do?method=topStoreRank&rank=1&pageGubun=L645"
url

link <-read_html(url)
```

rvest는 R의 웹 스크래핑(Web Scraping)을 위한 패키지로 해당 패키지를 인스톨하고 라이브러리로 넣어준다.
url은 해당 사이트의 주소이다.
link 값에 html을 read 하고 확인하면 아래와 같이 해당 html 소스를 가져온 것을 볼 수 있다.

```
> link
{html_document}
<html lang="ko">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">\n<meta charset="EUC-KR">\n<title>1등 배출점 &lt; 당첨 판매점 &lt; 판매점 ¦ 동행복권</title>\
 include  ...
[2] <body>\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n<div>\r\n\r\n\t\r\n\t\t\r\n\t\t\r\n\t\t\t\r\n<script type="text/javascript">\r\n\t$(do
>
```

※ class는.으로 찾아들어가고 id는 #으로 찾아들어간다.

#해당 값 가져오기

여기서부터는 해당 태그 부분에 들어가는 작업이다.
1. 로드워킹

```
#로드워킹(처음부터 순차적으로 찾아 들어가는 방법)
link %>%
  html_nodes('body') %>% #태그 이름으로 찾는것을 의미한다 (중복되지 않는경우에서만 허용된다.)
    html_nodes('.containerWrap') %>%
      html_nodes('.contentSection') %>%
        html_nodes('#article')
```

```
lotto15 ← link %>%
              html_nodes('tbody tr td') %>%
              html_text()
```

2. 디렉트

```
#아이디값으로 바로 찾는 방법
link %>%
  html_nodes('#article')
```

가져온 값을 lotto15라는 값에 넣는다.

#전처리 작업

필요 없는 값을 지워주기 위해 전처리를 한다.

```
library(stringr)

lotto15 ← str_replace_all(lotto15, "\t¦\n¦\r", "" )
lotto15 ← str_replace_all(lotto15, "[[:space:]]", "" )
```

데이터 전처리 후 출력 값이다.

```
[4] "서울노원구상계8동666-3주공10단지종합상가111" "스파지도보기"                    "2"
 [7] "부일카서비스"                                  "34"                          "부산동구범일제2동830-195번지"
[10] "부일카서비스지도보기"                          "3"                           "일등복권편의점"
[13] "24"                                            "대구달서구본리동2-16번지1층"  "일등복권편의점지도보기"
[16] "4"                                             "세진전자통신"                 "16"
[19] "대구서구평리3동1094-4번지"                     "세진전자통신지도보기"          "5"
[22] "로또휴게실"                                    "15"                          "경기용인시기흥구상갈동378-1"
[25] "로또휴게실지도보기"                            "6"                           "목화휴게소"
[28] "13"                                            "경남사천시용현면주문리4"      "목화휴게소지도보기"
[31] "7"                                             "GS25(양산혜인점)"             "12"
[34] "경남양산시평산동31-5번지"                      "GS25(양산혜인점)지도보기"      "8"
[37] "로또명당인주점"                                "12"                          "충남아산시인주면신성리188-8"
[40] "로또명당인주점지도보기"                        "9"                           "뉴빅마트"
[43] "11"                                            "부산기장군정관읍매학리748-5106호" "뉴빅마트지도보기"
[46] "10"                                            "잠실매점"                     "11"
[49] "서울송파구잠실6동7-18번지잠실역8번출구앞가판" "잠실매점지도보기"              "11"
[52] "버스판매소"                                    "10"                          "서울영등포구영등포동440번지신세계앞"
[55] "버스판매소지도보기"                            "12"                          "인터넷복권판매사이트"
[58] "10"                                            "서초구서초동동행복권(dhlottery.co.kr)" "인터넷복권판매사이트지도보기"
[61] "13"                                            "제이복권방"                   "10"
[64] "서울종로구종로5.6가동58번지평창빌딩1층103호"  "제이복권방지도보기"          "14"
[67] "갈랩분식한식"                                  "9"                           "서울중랑구망우본동490-13번지"
[70] "갈랩분식한식지도보기"                          "15"                          "라이프마트"
[73] "9"                                             "인천중구연안동58-98번지5호"   "라이프마트지도보기"
```

#데이터를 정형화

이제 가지고 데이터를 정형화하는 방법
지금 현재 값이 vector로 순서대로 들어있다. 두 번째 값에는 이름 세 번째 값에는 번호 네 번째 값에는 address 가 들어있다. 그래서 이걸 5로 나눠서 나머지 값이 2,3,4 일 때 저장하는 조건으로 for문을 만들었다. 이걸 다 만든 후 data.frame으로 출력하면

```r
storeName <- NULL
cnt       <- NULL
address   <- NULL

for(idx in 1:length(lotto15)){
  if(idx %%5 == 2){
    storeName <- c(storeName,lotto15[idx])
  }else if(idx %% 5 ==3){
    cnt <- c(cnt,lotto15[idx])
  }else if(idx %% 5 == 4){
    address <- c(address , lotto15[idx])
  }
}

lottoDF <- data.frame(storeName, cnt,address)
lottoDF
```

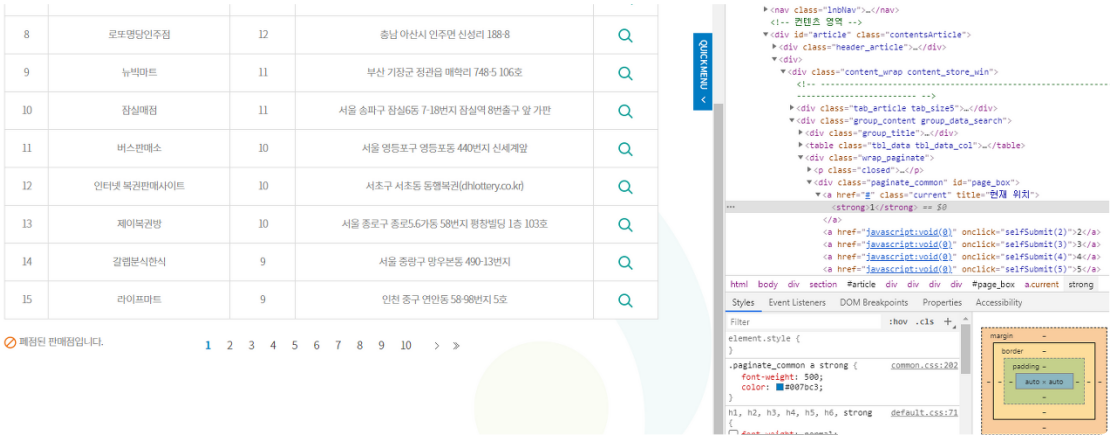
아래와 같은 값이 나온다.

	storeName	cnt	address
1	스파	35	서울노원구상계8동666-3주공10단지종합상가111
2	부일카서비스	34	부산동구범일제2동830-195번지
3	일등복권편의점	24	대구달서구본리동2-16번지1층
4	세진전자통신	16	대구서구평리3동1094-4번지
5	로또휴게실	15	경기용인시기흥구상갈동378-1
6	목화휴게소	13	경남사천시용현면주문리4
7	GS25(양산혜인점)	12	경남양산시평산동31-5번지
8	로또명당인주점	12	충남아산시인주면신성리188-8
9	뉴빅마트	11	부산기장군정관읍매학리748-5106호
10	잠실매점	11	서울송파구잠실6동7-18번지잠실역8번출구앞가판
11	버스판매소	10	서울영등포구영등포동440번지신세계앞
12	인터넷복권판매사이트	10	서초구서초동동행복권(dhlottery.co.kr)
13	제이복권방	10	서울종로구종로5.6가동58번지평창빌딩1층103호
14	갈렙분식한식	9	서울중랑구망우본동490-13번지
15	라이프마트	9	인천중구연안동58-98번지5호

정리돼서 나오는 것을 볼 수 있다.

#셀레니움을 이용한 페이지 자동화 크롤링

첫 번째 페이지는 잘 나왔지만 2페이지 3페이지~마지막 페이지까지의 값을 갖고 싶다.

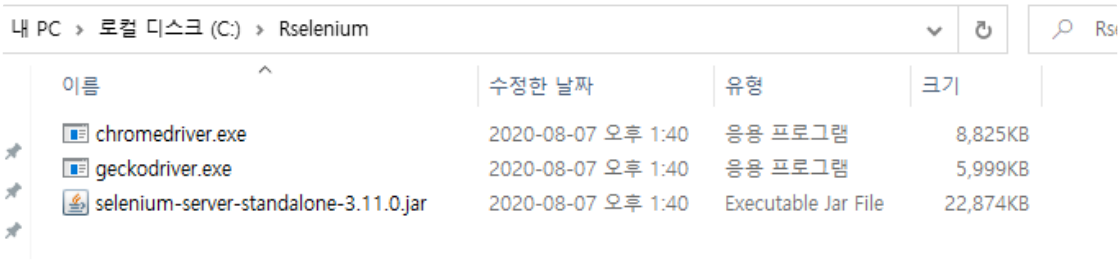


여기서 다음 순번의 로또 배출점을 구하려 한다.

페이지를 넘기는 형태를 보니 selfSubmit이라는 onclick을 통해 페이지가 넘어가고 있었다.

```
</a>
<a href="javascript:void(0)" onclick="selfSubmit(2)">2</a> ==
<a href="javascript:void(0)" onclick="selfSubmit(3)">3</a>
<a href="javascript:void(0)" onclick="selfSubmit(4)">4</a>
<a href="javascript:void(0)" onclick="selfSubmit(5)">5</a>
<a href="javascript:void(0)" onclick="selfSubmit(6)">6</a>
<a href="javascript:void(0)" onclick="selfSubmit(7)">7</a>
<a href="javascript:void(0)" onclick="selfSubmit(8)">8</a>
<a href="javascript:void(0)" onclick="selfSubmit(9)">9</a>
<a href="javascript:void(0)" onclick="selfSubmit(10)">10</a>
<a class="go next" href="javascript:void(0)" onclick=
"selfSubmit(11)">다음 페이지</a>
<a class="go end" href="javascript:void(0)" onclick=
"selfSubmit(198)">끝 페이지</a>
```

사전에 이거 먼저 깔아주셔야 됩니다.



cmd 창을 통해 셀레니움을 깐 폴더로 이동해서 아래의 명령어를 입력해 준다.
java -Dwebdriver.gecko.driver="geckodriver.exe" -jar selenium-server-standalone-3.11.0.jar -port 4445

이제 R에서 셀레니움을 install을 해준다.
셀레니움은 동적 웹페이지 크롤링을 가능케 하는 함수이다. 동적이라는 건 어떠한 동작을 하면서 크롤링하는 작업을 말한다.

```
228
229 install.packages("RSelenium")
230 library(RSelenium)
231
```

아래 작업은 페이지의 마지막 값을 알기 위해 사용하였다.

```
last ← link %>%
  html_nodes('.paginate_common') %>%
  html_nodes('a') %>%
  html_attr('onclick') %>% tail(1)
```

```
> last
[1] "selfSubmit(198)"
```

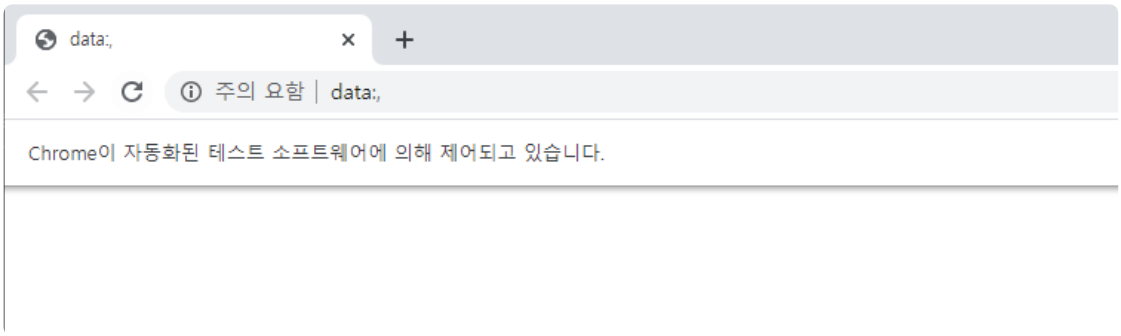
```
> end ← regmatches(last,gregexpr('[0-9]',last))
> end ← as.numeric(end[[1]])
> end ← as.numeric(paste(end[1],end[2],end[3], sep=""))
> end
[1] 198
```

이 가져온 값을 전처리를 통해 숫자의 값만 가져온다.

remoteDriver를 세팅해준다.
remoteDriver 클래스는 JsonWireProtocol을 사용하여 Selenium 서버와 통신하게 끔하다.

```
remDr ← remoteDriver(remoteServerAddr ="localhost", #내 아이피에서
                     port=4445L,                    # 포트는 4445를 사용하고
                     browserName = 'chrome')       #브라우저는 크롬을 사용한다.
remDr$open()                                       #자동화를 가능케하는 사이트를 연다.
```

open을 할경우 아래와 같이 창이 뜬다. 저렇게 자동화된 테스트 소프트웨어~ 라고 써있다.



그 다음 url을 넣어서 navigate를 넣어주면 해당 url로 이동한다.

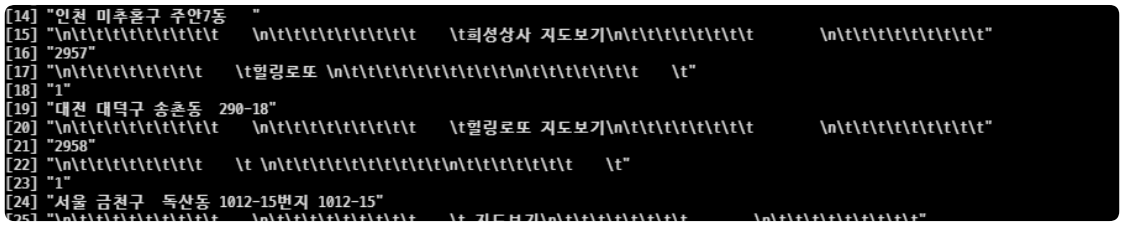
```
248 remDr$navigate("https://dhlottery.co.kr/store.do?method=topStoreRank&rank=1&pageGubun=L645") #해당 url로 이동하게 한다.
```

크롤링하는 소스이다.

```
251
252 lottoStore = c() #초기화 해준다.
253
254 #1페이지부터 끝 페이지 까지의 정보를 가져오기 위한 for문
255 #현재 자동화 되고있는 사이트를 아래의 for문을 통해 submit을 날리라는 명령어다.
256
257 for(idx in 1:end){
258   front <-"selfSubmit("
259   back <-")"
260   script <- paste(front, idx , back,sep="") # 이 3문장으로 script에 넣고 스크립트를 실행 시킬 단어를 만든다. 여기서는 script는 selfSubmit(숫자) 값이다.
261   pagemove <- remDr$executeScript (script, args=1:2) # 이 script에  들어있는 텍스트를 받아 javascript를 실행 시킨다.
262
263   #소스 받아오기
264   source <- remDr$getPageSource()[[1]] # 현재 자동화로 켜진 사이트의 소스를 가져온다.
265   js_html <- read_html(source) # 소스를 받아온다.
266
267   js_link <- html_nodes(js_html, 'tbody') #tbody라는 태그를 가져온다.
268   js_link
269
270   stores <- js_link %>%      #tbody에 안에있는 내용에서 tr을 타고 td 까지 탄다음에 td안에 있는 값을 text로 받는다.
271     html_nodes('tr') %>%
272     html_nodes('td') %>%
273     html_text()
274
275   lottoStore = c(lottoStore, stores)   # lottoStore 에 벡터 형태로 추가해서 넣어주는 작업을 한다. 이렇게 for문에서 나온 데이터를  lottoStore에 계속 넣는다.
276
277 }
278
```

이 selfSubmit에 따라 자바스크립트가 실행되면 페이지가 1->198까지 순차적으로 이동한다.

위에 for문을 돌리게 되면 페이지가 자동으로 넘어가고 넘어가는 페이지를 lottoStore에 저장한다.



값을 출력해 보면 위같은 형식으로 나온다.

내가 가져온 값은

이 부분에 해당하는 값을 전부 가져왔다.

그다음 가져온 값을 전처리하는 작업을 거친다.

```r
#전처리 작업
lottoStore ← str_replace_all(lottoStore, "\t¦\n¦\r", "" )    # 필요없는 \t\n\r text 지우기
lottoStore ← str_replace_all(lottoStore, "[[:space:]]", "" )

storeName ← NULL
cnt       ← NULL
address   ← NULL

for(idx in 1:length(lottoStore)){    #두번째 값이 storeName 이고 세번째 값이 cnt(번호) 이고 4번째 값이 address 이다.
  if(idx %%5 == 2){                  #나머지 값은 필요가 없어서 이 순서대로 값을 가져와서 저장한다.
    storeName ← c(storeName,lottoStore[idx])
  }else if(idx %% 5 ==3){
    cnt ← c(cnt,lottoStore[idx])
  }else if(idx %% 5 == 4){
    address ← c(address , lottoStore[idx])
  }
}

lottoDF ← data.frame(storeName, cnt,address) #만든걸 데이터프레임으로 만든다.
lottoDF
```

전처리 후 결과이다.

```
[905] "우미슈퍼지도보기"                    "182"
[907] "우일"                                "4"
[909] "서울송파구마천2동25-2번지"           "우일지도보기"
[911] "183"                                 "우정식품"
[913] "4"                                   "부산동래구온천제1동185-93번지"
[915] "우정식품지도보기"                    "184"
[917] "운수대통"                            "4"
[919] "경기수원시권선구호매실동87-2"        "운수대통지도보기"
[921] "185"                                 "원당역복권방"
[923] "4"                                   "경기고양시덕양구성사동410-7"
[925] "원당역복권방지도보기"                "186"
```

마지막 만든 데이터 프레임을 csv파일로 생성한다.

```r
lottoDF ← data.frame(storeName, cnt,address) #만든걸 데이터프레임으로 만든다.
lottoDF

write.csv(lottoDF , "lotto_store.csv", row.names = F) # csv 파일로 읽는다.
```

csv 파일을 읽으면 아래와 같이 나온다.

	A	B	C	D	E	F	G	H
1	storeName	cnt	address					
2	스파	35	서울노원구상계8동666-3주공10단지종합상가111					
3	부일카서비	34	부산동구범일제2동830-195번지					
4	일등복권판	24	대구달서구본리동2-16번지1층					
5	세진전자통	16	대구서구평리3동1094-4번지					
6	로또휴게실	15	경기용인시기흥구상갈동378-1					
7	목화휴게소	13	경남사천시용현면주문리4					
8	GS25(양산	12	경남양산시평산동31-5번지					
9	로또명당인	12	충남아산시인주면신성리188-8					
10	뉴빅마트	11	부산기장군정관읍매학리748-5106호					
11	잠실매점	11	서울송파구잠실6동7-18번지잠실역8번출구앞가판					
12	버스판매소	10	서울영등포구영등포동440번지신세계앞					
13	인터넷복권	10	서초구서초동동행복권(dhlottery.co.kr)					
14	제이복권방	10	서울종로구종로5.6가동58번지평창빌딩1층103호					
15	갈렙분식한	9	서울중랑구망우본동490-13번지					
16	라이프마트	9	인천중구연안동58-98번지5호					
17	북마산복권	9	경남창원시마산합포구39-4번지					

'**R**' 카테고리의 다른 글□

[R]R을 활용한 상관분석과 회귀분석 - 2□

[R] R을 활용한 크롤링 - 로또 1등 당첨 배출점 크롤링 하기□

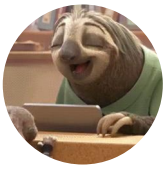
[R] R에서 교차검증을 위한 데이터 셋 분리방법 3가지□

[R] R을 활용한 상관분석과 회귀분석 - 1□

[R] R을 통한 텍스트마이닝에서 워드클라우드 까지□

[R] R로 하는 비정형 데이터 처리 (facebook 데이터를 통한 긍정/부정 나누기)□

R 크롤링 R로 하는 크롤링 R을 통한 크롤링 크롤링



꾸까꾸

혼자 끄적끄적하는 블로그 입니다.