PR10 - Web Scraping

조성우

2020 5 23

다음 영화리뷰 크롤링

```
#install.packages(c('httr','rvest'))

library(httr)

## Warning: package 'httr' was built under R version 3.6.3

library(rvest)

## Warning: package 'rvest' was built under R version 3.6.3

## Loading required package: xml2

review <-NULL #반복문으로 크롤링한 모든 리뷰를 모두 한곳에 할당하기 위한 변수 review를 선언
star <- NULL #반복문으로 크롤링한 모든 별점을 모두 한곳에 할당하기 위한 변수 star를 선언
date <- NULL ##반복문으로 크롤링한 모든 날짜를 모두 한곳에 할당하기 위한 변수 date를 선언

for (i in 1:10){
 url <-
"https://movie.daum.net/moviedb/grade?movieId=87215&type=netizen&page="
#반복문으로 크롤링 해올 페이지들을 하나하나 입력하지 않고 해당하는 주소처럼 query 부분(page= 이후) 을 비워두고 url 에 할당하여 반복문의 i 횟수에 맞게 i 를 할당하여 크롤링하여 페이지를 넘기는식으로 크롤링하고자 합니다
 urls <- paste(url,i,sep="") # url과 i를 합쳐 반복문을 활용한 연속되는 다른페이지를 크롤링해오고자 paste를 활용합니다.
 html_source = read_html(urls)    #read_html을 사용하여 반복문 내에서합성된 urls를 읽어들입니다

 # review
 review_nodes <-html_nodes(html_source, 'p.desc_review') #개발자환경에서 html code를 분석해본 결과 크롤링하고자하는 review는 p 항목 아래에 있는 desc_review 항목으로 존재합니다.
 review_i <- html_text(review_nodes) #텍스트 추출 : 위에서 p.desc_reivew를 저장한 것중 text만 따로 저장합니다
```

```
 review <- append(review, review_i) #누적 : text만 따로 저장한것을 반복문 위에서 미리 선언해둔 review변수와 합쳐주는 방식으로 반복문마다 하나씩 추가시킵니다.

 #rating
 star_nodes <- html_nodes(html_source, 'em.emph_grade') #개발자환경에서 html code를 분석해본 결과 크롤링하고자하는 별점는 em항목 아래에 있는 emph_grade 항목으로 존재합니다.
 star_i <- html_text(star_nodes) #텍스트 추출 : 위에서 em.emph_grade 를 저장한 것중 text만 따로 저장합니다
 star <- append(star,star_i) #누적 : text만 따로 저장한것을 반복문 위에서 미리 선언해둔 star 변수와 합쳐주는 방식으로 반복문마다 하나씩 추가시킵니다.

 # date
 date_nodes = html_nodes(html_source, 'span.info_append') ##개발자환경에서 html code를 분석해본 결과 크롤링하고자하는 날짜는 span항목 아래에 있는 info_append 항목으로 존재합니다.
 date_i <- html_text(date_nodes) #텍스트 추출 : 위에서 span.info_append 를 저장한 것중 text만 따로 저장합니다
 date <- append(date,date_i) #누적 : text만 따로 저장한것을 반복문 위에서 미리 선언해둔 date 변수와 합쳐주는 방식으로 반복문마다 하나씩 추가시킵니다.
}

#merge
daum_m <- data.frame(date,star,review) # 위의 반복문에서 각각 스크래핑된 date,star,review를 merge함수를 통해 데이터프레임으로 합쳐줍니다.

#date - cleaning
daum_m[,1] <- gsub("\n","",daum_m[,1])    #gsub함수를 사용하여 dataframe내의 1열에 해당하는 date 데이터에서 불필요한 \n을 제거합니다.
daum_m[,1] <- gsub("\t","",daum_m[,1])  #gsub함수를 사용하여 dataframe내의 1열에 해당하는 date 데이터에서 불필요한 \t을 제거합니다.

#review - cleaning
daum_m[,3] <- gsub("\r","",daum_m[,3]) #gsub함수를 사용하여 dataframe내의 3열에 해당하는 reivew 데이터에서 불필요한 \r을 제거합니다.
daum_m[,3] <- gsub("\n","",daum_m[,3]) #gsub함수를 사용하여 dataframe내의 3열에 해당하는 reivew 데이터에서 불필요한 \n을 제거합니다.
daum_m[,3] <- gsub("\t","",daum_m[,3]) #gsub함수를 사용하여 dataframe내의 3열에 해당하는 reivew 데이터에서 불필요한 \t을 제거합니다.
```

```
daum_m[,3] <- trimws(daum_m[,3]) #문자열의 앞뒤 공백 제거

write.csv(daum_m,file = "movie_review.csv") #전처리한 daum_m 데이터프레임을 movie_review.csv라는 이름으로 지정된 디렉토리에 저장합니다.
```

PR10 연습문제

문제1 * 위의 코드는 다음영화페이지 크롤링 코드입니다. 모든 코드의 주석을 상세히 달아주세요. 본인이 아는한 최대한 자세히 적어주세요.

문제2 * 위의 코드는 날짜, 평점, 리뷰내용을 크롤링하고있습니다. 다음영화페이지에서 그외의 데이터중 크롤링이 가능한것이 있으면 시도해보시기 바랍니다. 그리고 그 결과를 출력해주세요. 다음영화페이지가 아닌 다른 웹페이지의 내용을 크롤링하시면 더욱 좋습니다.

```
#수상정보를 크롤링하겠습니다.

library(httr)
library(rvest)
library(stringr)

## Warning: package 'stringr' was built under R version 3.6.3

award <- NULL

for (i in 1:5){
  url <- "https://movie.daum.net/moviedb/award?movieId=87215&page="
  urls <- paste(url,i,sep="")
  html_source = read_html(urls)

  # review
  award_nodes <-html_nodes(html_source, '.tit_movie') #award 위치
  award_i <- html_text(award_nodes) #텍스트 추출
  award <- append(award, award_i) #누적
}


award <- str_trim(award)
award

##  [1] "17회 디렉터스 컷 어워즈, 2017"      "1회 한중국제영화제, 2017"
##  [3] "21회 부천국제판타스틱영화제, 2017" "22회 춘사영화제, 2017"
```

```
##  [5] "53회 백상예술대상, 2017"                "31회 워싱턴DC국제영화제, 2017"
##  [7] "31회 프리부르국제영화제, 2017"          "15회 피렌체 한국영화제, 2017"
##  [9] "11회 아시안필름어워즈, 2017"            "46회 로테르담국제영화제, 2017"
## [11] "8회 올해의 영화상, 2017"                "3회 한국영화제작가협회상, 2016"
## [13] "37회 청룡영화상, 2016"                  "47회 인도국제영화제, 2016"
## [15] "36회 한국영화평론가협회상, 2016"        "17회 샌디에이고아시안영화제, 2016"
## [17] "11회 파리한국영화제, 2016"              "25회 부일영화상, 2016"
## [19] "49회 시체스국제영화제, 2016"            "20회 판타지아국제영화제, 2016"
## [21] "69회 칸영화제, 2016"
```

문제3 * 크롤링이 기업경영이나, 새로운 서비스 창출에 중요하게 사용된 사례를 찾고 소개해주세요.

패션상품 검색엔진을 주 영업으로하는 크로키닷컴의 지그재그앱에 활용된 크롤링 기술을 소개하겠습니다. 2015 년 베타서비스를 시작한 지그재그는 그당시 각 웹페이지에 흩어져있던 수많은 온라인 쇼핑몰을 지그재그에 모두 모아 유저가 원하는 옷을 종합적으로 빠르게 찾도록 도와주고자 런칭한 어플리케이션입니다.

위의 설명에서 알수있다시피, 지그재그의 핵심기술은 웹크롤링 기술입니다.

앱개발 초기당시, 여성패션쇼핑몰의 고객유입 경로를 조사하였는데 네이버를 통한 유입이 가장 많을것이란 대표의 예상과 달리 네이버의 웹페이지 즐겨찾기 서비스인 북마크를 통한 유입이 가장 많았다는 뜻밖이 결과에서 쇼핑몰전용 즐겨찾기 서비스를 제공하는 어플을 만들고자 하는 아이디어를 얻었고 베타버전은 해당 아이디어만을 구현하여 매출상위 300 위까지의 쇼핑몰만을 대상으로 각 제품을 일일이 스크래핑해온 후 서비스하였습니다.

이후 대표는 시장의 반응이 긍정적임을 확인하고 어플리케이션의 프로세스 개선을 위한 아이디어로

크롤링 기술을 도입하여 더 많은 쇼핑몰의 더 많은 제품에 대한 스크래핑 자동화를 구현하고자 하였고

이를 자동적으로 웹을 돌아다니며 상품정보와 이미지를 긁어오는 크롤링 알고리즘 봇을 개발하여 도입함으로서 앱을 개선하여 서비스 할 수 있게되었고 해당 서비스를 통해 수많은 쇼핑몰의 개별 상품들의 정보를 앱 안으로 가져옴으로서 당사는 앱안에서 행해지는 유저들의 소비자행동을 매우 구체적으로 파악하고 이를 적극적으로 활용할 수 있었습니다.

PR10 도전문제

- 위의 코드는 for loop 내에서 i가 1부터 10까지만 동작합니다. 하지만 실제 영화리뷰가 10개만 있는것은 아닙니다. 위의 코드와 수업시간에 학습하였던 CSS selector를 활용하여 실제 영화의 리뷰수만큼을 모두 크롤링할 수 있도록 만들어주세요.

```
library(rvest)
library(httr)
library(stringr)

url_base<-
"https://movie.daum.net/moviedb/grade?movieId=87215&type=netizen&page="

url_baseToSpan<-
"https://movie.daum.net/moviedb/grade?movieId=87215&type=netizen&page=1"   #
별점을 준 네티즌 수를 보여주는 html 코드를 읽기 위한 url  하나를 선언합니다

span <- read_html(url_baseToSpan)
span <- html_nodes(span,'span.txt_menu') %>% html_text() #html 에서 별점을 준 네티즌
수를 보여주는것은 span의 txt_menu  class 입니다.  이를 span 변수에 할당합니다


span <- gsub(",","",span)    #할당된 span 에서 불필요한 특수문자를 제거합니다.
span <- gsub("[)]","",span)
span <- gsub("[(]","",span)

span <- span[1]            #span.txt_menu 는 네티즌별점갯수, 전문기별점갯수 두개를 보여주는데
우리가 관심있는것은 네티즌별점갯수이므로 첫번째에 할당된 네티즌별점갯수만 할당합니다.
span <- as.numeric(span)    #이를 활용하기위해 numeric 형으로 변환하여줍니다.

if (span %% 10 == 0){    #한페이지당 10개의 리뷰를 가지므로 10으로 나눠주고,  나머지가 0이 아닐
경우 이에대해 1페이지가 추가되므로 조건문으로 해당 코드를 만들어줍니다.
  span <- span%/%10
}else {
  span <- span%/%10 + 1
}

span

## [1] 811

total_review =NULL
for (i in 1:span){
  url<- paste0(url_base,i)
  htxt <- read_html(url)
```

```
  user <- html_nodes(htxt,'.link_profile') %>% html_text()
  grade<- html_nodes(htxt,'.emph_grade') %>% html_text()
  review <- html_nodes(htxt,'.desc_review') %>% html_text()
  page <-data.frame(user,grade,review)

  total_review <- rbind(total_review,page)
}

total_review[,1] <- gsub("\n","",total_review[,1])    #gsub 함수를 사용하여
dataframe 내의 1 열에 해당하는 date  데이터에서 불필요한 \n을 제거합니다.
total_review[,1] <- gsub("\t","",total_review[,1])  #gsub 함수를 사용하여
dataframe 내의 1 열에 해당하는 date  데이터에서 불필요한 \t을 제거합니다.

#review - cleaning
total_review[,3] <- gsub("\r","",total_review[,3]) #gsub 함수를 사용하여
dataframe 내의 3 열에 해당하는 reivew  데이터에서 불필요한 \r을 제거합니다.
total_review[,3] <- gsub("\n","",total_review[,3]) #gsub 함수를 사용하여
dataframe 내의 3 열에 해당하는 reivew  데이터에서 불필요한 \n을 제거합니다.
total_review[,3] <- gsub("\t","",total_review[,3]) #gsub 함수를 사용하여
dataframe 내의 3 열에 해당하는 reivew  데이터에서 불필요한 \t을 제거합니다.
total_review[,3] <- trimws(total_review[,3]) #문자열의 앞뒤 공백 제거

write.csv(total_review,"pages.csv")
```

##과제 수행시 주의사항* 크롤링의 경우 오류가 없는 코드임에도 마크다운 사용시 오류가 나는 경우가 있습니다. 오류가 발생하는경우 rstudio 를 껐다가 켜본 후 그래도 해결되지 않으면, r 파일로 작성하셔서 제출하시거나, 마크다운을 워드파일이 아닌 html 로 변환하여 압축하고 zip 파일로 제출해 주시기 바랍니다.