PR8

PR8\_201821479\_황혜린

2018년 11월 9일

# 문제 1 : 다음 영화 리뷰 크롤링

#install.packages(c('httr', 'rvest'))  
library(httr) #post방식으로 데이터를 요청해서 가져오는 경우에 추가로 사용   
library(rvest) #get 방식의http 주소를 사용하는경우

## Loading required package: xml2

review = NULL  
star = NULL  
date = NULL #각 변수들에 아직 값이 할당하기 이전에 NULL을 지정합니다  
  
for (i in 1:10) {  
 url <- c("https://movie.daum.net/moviedb/grade?movieId=87215&type=netizen&page=")  
 urls <- paste(url,i,sep="") #1~10페이지를 불러옵니다. url에 할당한 페이지주소에 paste를 통해 번호를 조합하여 실행합니다  
 html\_sourse = read\_html(urls) #read\_html함수를 사용하여 html페이지를 html\_sourse라는 변수에 저장  
   
 #review  
 review0 = html\_nodes(html\_sourse ,'p.desc\_review')  
 #html\_nodes 함수를 사용하여 리뷰글(p.desc\_review)에 해당하는 부분을 review0에 할당   
 review1 = html\_text(review0) #html\_text함수를 사용하여 text부분을 review1변수에 저장  
 review = append(review, review1) #review에 review1값을 지정합니다  
   
  
 #rating  
 star0 = html\_nodes(html\_sourse, 'em.emph\_grade')  
 #별점('em.emph\_grade')을 star0에 할당   
 star1 = html\_text(star0) #텍스트부분을 star1에 저장   
 star = append(star, star1) #star에 star1값을 지정   
   
 #date  
 date0 = html\_nodes(html\_sourse, 'span.info\_append')  
 #날짜('span.info\_append')를 date0에 할당   
 date1 = html\_text(date0) #텍스트부분을 date1에 저장   
 date = append(date, date1) #date에 date1값을 지정   
}  
  
#merge  
daum\_m = data.frame(date, star, review)  
#date, star, review라는 컬럼을 가진 daum\_m데이터프레임을 생성합니다  
  
#date - cleaning  
daum\_m[,1] = gsub("\n", "", daum\_m[,1])  
daum\_m[,1] = gsub("\t", "", daum\_m[,1])  
#daum\_m의 1열에서 리뷰 데이터를 불러올 때 불필요하게 생성된 \n, \t를 삭제합니다  
  
#review - cleaning  
daum\_m[,3] = gsub("\r", "", daum\_m[,3])  
daum\_m[,3] = gsub("\n", "", daum\_m[,3])  
daum\_m[,3] = gsub("\t", "", daum\_m[,3])  
#daum\_m의 3열에서 데이터를 불러올 때 불필요하게 생성된 \n, \t, \r을 삭제합니다  
  
write.csv(daum\_m, file = "movie\_review.csv") #만든 데이터프레임을 movie\_review라는 이름의 csv파일로 저장합니다

# 문제2 나혼자산다 클립영상

library(httr) #post방식으로 데이터를 요청해서 가져오는 경우에 추가로 사용   
library(rvest) #get 방식의http 주소를 사용하는경우  
  
contents = NULL  
Press = NULL  
 #각 변수들에 아직 값이 할당하기 이전에 NULL을 지정합니다  
  
for (i in 1:5) {  
 url <- c("https://search.daum.net/search?w=vclip&q=%EB%82%98%20%ED%98%BC%EC%9E%90%20%EC%82%B0%EB%8B%A4&spacing=0&DA=PGD&page=")  
 urls <- paste(url,i,sep="") #1~5페이지를 불러옵니다. url에 할당한 페이지주소에 paste를 통해 번호를 조합하여 실행합니다  
 html\_sourse = read\_html(urls) #read\_html함수를 사용하여 html페이지를 html\_sourse라는 변수에 저장  
   
  
 contents0 = html\_nodes(html\_sourse ,'a.f\_link\_b')  
 #클립영상 제목에 해당하는 부분을 contents0에 할당   
 contents1 = html\_text(contents0) #text부분을 contents1변수에 저장  
 contents = append(contents, contents1) #contents에 contents1값을 지정합니다  
   
  
 Press0 = html\_nodes(html\_sourse, 'a.f\_nb')  
 #영상을 올린 매체를 Press0에 할당   
 Press1 = html\_text(Press0) #텍스트부분을 Press1에 저장  
 Press = append(Press, Press1) #Press에 Press1값을 지정   
   
}  
  
  
daum\_p = data.frame(contents, Press)  
#contents, Press라는 컬럼을 가진 daum\_p데이터프레임을 생성합니다  
  
#cleaning  
daum\_p[,1] = gsub("[\r\n\t]", "", daum\_p[,1])  
daum\_p[,2] = gsub("[\r\n\t]", "", daum\_p[,2])  
#데이터를 불러올 때 불필요하게 생성된 \n, \t, \r을 삭제합니다  
  
write.csv(daum\_p, file = "live\_alone.csv") #만든 데이터프레임을 line\_alone이라는 이름의 csv파일로 저장합니다

# 문제3 Ancestry.com 온라인 가계도 서비스

-Ancestry.com은 미국과 캐나다, 유럽, 호주등 이민자들로 형성된 나라에서 고객의 뿌리를 찾아주는 비즈니스 모델을 개발하여 사업을 전개하였으며 현재 200만 명 이상의 회원을 보유한 회사입니다.

-크롤링(Crawling)을 통해 수집된 디지털 데이터, 스캔된 행정 문서를 활용하여 작성된 온라인데이터베이스에 사용자들이 제공한 데이터를 결합하여 조상과 관련된 콘텐츠 서비스를 제공합니다.

-축적된 데이터는 4페타바이트급으로 10페타바이트(10,000,000GB) 데이터 저장소에 보관 중이라고 합니다.

# 도전문제

#install.packages(c('httr', 'rvest'))  
library(httr) #post방식으로 데이터를 요청해서 가져오는 경우에 추가로 사용   
library(rvest) #get 방식의http 주소를 사용하는경우  
  
Review = NULL  
Star = NULL  
Date\_ = NULL #각 변수들에 아직 값이 할당하기 이전에 NULL을 지정합니다  
  
for (i in 1:829) {  
 url <- c("https://movie.daum.net/moviedb/grade?movieId=87215&type=netizen&page=")  
 urls <- paste(url,i,sep="") #모든 리뷰를 크롤링하기 위해 829페이지를 불러옵니다.   
 html\_sourse = read\_html(urls) #read\_html함수를 사용하여 html페이지를 html\_sourse라는 변수에 저장  
   
 #review  
 Review0 = html\_nodes(html\_sourse ,'p.desc\_review')  
 #html\_nodes 함수를 사용하여 리뷰글(p.desc\_review)에 해당하는 부분을 Review0에 할당   
 Review1 = html\_text(Review0) #html\_text함수를 사용하여 text부분을 Review1변수에 저장  
 Review = append(Review, Review1) #Review에 Review1값을 지정합니다  
   
  
 #rating  
 Star0 = html\_nodes(html\_sourse, 'em.emph\_grade')  
 #별점('em.emph\_grade')을 Star0에 할당   
 Star1 = html\_text(Star0) #텍스트부분을 Star1에 저장   
 Star = append(Star, Star1) #Star에 Star1값을 지정   
   
 #date  
 Date\_0 = html\_nodes(html\_sourse, 'span.info\_append')  
 #날짜('span.info\_append')를 Date\_0에 할당   
 Date\_1 = html\_text(Date\_0) #텍스트부분을 Date\_1에 저장   
 Date\_ = append(Date\_, Date\_1) #Date\_에 Date\_1값을 지정   
}  
  
#merge  
Daum\_m = data.frame(Date\_, Star, Review)  
#Date\_, Star, Review라는 컬럼을 가진 Daum\_m데이터프레임을 생성합니다  
  
#cleaning  
Daum\_m[,1] = gsub("[\n\r\t]", "", Daum\_m[,1])  
Daum\_m[,2] = gsub("[\n\r\t]", "", Daum\_m[,2])  
Daum\_m[,3] = gsub("[\n\r\t]", "", Daum\_m[,3])  
#데이터를 불러올 때 불필요하게 생성된 \n, \t, \r을 삭제합니다  
  
write.csv(Daum\_m, file = "all\_movie\_review.csv") #만든 데이터프레임을 all\_movie\_review라는 이름의 csv파일로 저장합니다