Упражнение 1

Дроздецкая Анна

21 02 2021

C помощью пакета rvest или парсинга XML с помощью xpath запростов соберите данные с сайта согласно своему варианту. В итоговой таблице должно быть не менее 50 записей и не менее 5 признаков, из которых как минимум два количественныхю Снабдите файл справочником в формате Markdawn. Результаты: .csv-файл с данными, .md-файл со справочником, .Rmd-файл с кодом загрузки данных разместить в репозитории github, файл .Rmd должен содержать постановку задачи и комментарии по ходу сбора данных.

## Загружаем библиотеку rvest и указываем ссылку на сайт

library('rvest')

## Loading required package: xml2

# Ссылка на топ 20 фильмов 2020 года  
url <- 'https://www.kinopoisk.ru/top/y/2020/'  
  
webpage <- read\_html(url)

# Место в топе

# Парсим номер места, которое фильм занимает в топе  
rank <- webpage %>% html\_nodes(".js-rum-hero td a") %>% html\_attr("name")  
# Избавляемся от NA в массиве  
rank[!is.na(rank)]

## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15"  
## [16] "16" "17" "18" "19" "20"

# Названия на русском языке

# Парсим названия фильмов на русском языке  
names\_ru <- webpage %>% html\_nodes(".all") %>% html\_text  
# Избавляемся от лишнего  
names\_ru <- names\_ru[3:22]  
names\_ru

## [1] "Душа (2020)"   
## [2] "Легенда о волках (2020)"   
## [3] "ВИЧ в России (2020)"   
## [4] "Доктор Лиза (2020)"   
## [5] "Довод (2020)"   
## [6] "Суд над чикагской семеркой (2020)"   
## [7] "Огонь (2020)"   
## [8] "Еще по одной (2020)"   
## [9] "Серебряные коньки (2020)"   
## [10] "Манк (2020)"   
## [11] "Социальная дилемма (2020)"   
## [12] "Зависнуть в Палм-Спрингс (2020)"   
## [13] "Легенды «Смертельной битвы»: Месть Скорпиона (2020)"  
## [14] "Зов предков (2020)"   
## [15] "Тёмная Лига справедливости: Война Апоколипса (2020)"  
## [16] "Дьявол всегда здесь (2020)"   
## [17] "Дорогие товарищи! (2020)"   
## [18] "Человек из Подольска (2020)"   
## [19] "Неадекватные люди 2 (2020)"   
## [20] "Банкир (2020)"

# Названия на английском языке

# Функция перебора тегов внутри тегов более высокого уровня  
get\_tags <- function(node){  
 raw\_data <- html\_nodes(node, selector) %>% html\_text  
 data\_NAs <- ifelse(length(raw\_data) == 0, NA, raw\_data)  
}  
  
selector <- '.text-grey'  
  
new\_names\_en = array()  
  
# Парсим названия фильмов на английском языке  
for(i in 1:length(names\_ru)){  
 tag <- paste0('#top250\_place\_', toString(i))  
 doc <- html\_nodes(webpage, tag)  
 names\_en <- sapply(doc, get\_tags)  
 new\_names\_en <- append(new\_names\_en, names\_en)  
}  
  
new\_names\_en = new\_names\_en[2:length(new\_names\_en)]  
new\_names\_en

## [1] "Soul"   
## [2] "WolfWalkers"   
## [3] NA   
## [4] NA   
## [5] "Tenet"   
## [6] "The Trial of the Chicago 7"   
## [7] NA   
## [8] "Druk"   
## [9] NA   
## [10] "Mank"   
## [11] "The Social Dilemma"   
## [12] "Palm Springs"   
## [13] "Mortal Kombat Legends: Scorpion's Revenge"  
## [14] "The Call of the Wild"   
## [15] "Justice League Dark: Apokolips War"   
## [16] "The Devil All the Time"   
## [17] NA   
## [18] NA   
## [19] NA   
## [20] "The Banker"

# Рейтинг фильма

# парсим рейтинг фильмов  
rating <- webpage %>% html\_nodes(".continue") %>% html\_text  
# Преобразуем строку в число  
rating <- as.numeric(rating)  
rating

## [1] 8.172 7.825 7.642 7.567 7.466 7.444 7.360 7.314 7.289 7.127 7.124 7.066  
## [13] 7.066 7.057 7.054 7.034 7.030 7.028 7.016 7.007

# Количество голосов

# Парсим количество голосов  
vote <- webpage %>% html\_nodes(".js-rum-hero div span") %>% html\_text  
# Избавляемся от скобок и пробелов  
vote <- gsub("[[:punct:]]", "", vote)  
vote <- gsub(pattern = "\\s", replacement = "", x=vote)  
# Преобразуем строку в число  
vote <- as.numeric(vote)  
vote

## [1] 76987 11543 2772 40018 177464 13731 23550 124897 14130 7719  
## [11] 4630 109290 8978 56282 3696 30327 6992 5664 53679 11748

# Создание директории и сохранение DataFrame в .csv

data.dir <- './data'  
  
# Создаем директорию для данных  
if (!file.exists(data.dir)) {  
 dir.create(data.dir)  
}  
  
# Создаём файл с логом загрузок  
log.filename <- './data/download.log'  
if (!file.exists(log.filename)) file.create(log.filename)

## [1] TRUE

DF <- data.frame(Rank = rank[!is.na(rank)], russian\_name = names\_ru,  
 english\_name = new\_names\_en, Rating = rating,  
 Vote = vote)  
  
# Загружаем данные в .csv файл  
write.csv(DF, file = './data/kinopoisk\_top\_2020.csv', row.names = FALSE)  
write(paste('File "kinopoisk\_top\_2020.csv" downloaded, date: ', Sys.time()), file = log.filename, append = TRUE)