Project_DataScraping

Shanshan Bradford

```
rm(list = ls())
library(xml2)
library(rvest)

setwd("/Users/syu/Library/CloudStorage/OneDrive-St.JudeChildren'sResearchHospital/UDrive/Documents_syu_l

moviefranch.url <- "http://www.the-numbers.com/movies/franchises/"

# read_html is used to read/extract the html content of a webpage, the output is a list
moviefranch.page <- read_html(moviefranch.url, options = c("NOBLANKS", "NSCLEAN", "DTDLOAD"))

# write scraped data into a file
write(as.character(moviefranch.page), "moviefranch_bgt.txt", sep = "\t")
print(moviefranch.page)
```

```
## {html_document}
## <html xmlns:og="http://ogp.me/ns#" xmlns:fb="http://www.facebook.com/2008/fbml">
## [1] <head>\n<link rel="icon" href="/images/logo_2021/favicon.ico">\n<meta nam ...
## [2] <body>\n\r\n\r\n<div id="wrap">\r\n\r\n<div id="desktopnav">\r\n\r\n<div  ...
```

```
# html_nodes function is used to identify the nodes in html document, the output is lists
table.nodes <- html_nodes(moviefranch.page, xpath = "//table")
# directly use node name for node selection
html_table(read_html(moviefranch.url, "table"), trim = T, fill = T)
```

```
## [[1]]
## # A tibble: 1,419 x 8
##    Franchise  `No. of Movies` `Domestic Box ~` `Infl. Adj. Do~` `Worldwide Box~`
##    <chr>                <int> <chr>            <chr>            <chr>
##  1 Marvel Ci~              39 $10,171,249,939  $10,779,515,564  $26,231,912,490
##  2 Star Wars               15 $5,080,586,579   $8,090,988,804   $10,318,326,428
##  3 James Bond              27 $2,297,557,630   $6,131,090,501   $7,880,393,492
##  4 Batman                  26 $3,152,959,867   $4,340,566,318   $6,801,084,908
##  5 Spider-Man              12 $3,302,222,088   $3,901,570,630   $8,257,025,229
##  6 Harry Pot~              13 $2,875,405,480   $3,751,231,931   $9,561,694,574
##  7 X-Men                   14 $2,458,465,265   $2,921,416,378   $6,075,264,152
##  8 Avengers                 4 $2,619,552,260   $2,760,009,130   $7,756,577,508
##  9 Jurassic ~               6 $1,882,802,527   $2,723,602,562   $5,008,426,006
## 10 Bambi                    2 $102,797,000     $2,704,336,016   $302,000,000
## # ... with 1,409 more rows, and 3 more variables: `First Year` <int>,
## #   `Last Year` <int>, `No. of Years` <int>
```

```
franch.table <- html_table(table.nodes, trim = T, fill = T)
write.csv(franch.table, file = "MovieFranchise_FinanceInfo.csv")


# set another url for scrapping historic inflation data
cpi.url <- "https://inflationdata.com/Inflation/Consumer_Price_Index/HistoricalCPI.aspx?reloaded=true"

# extract the html content of a webpage, the output is a list
cpi.page <- read_html(cpi.url, options = c("NOBLANKS", "NSCLEAN", "DTDLOAD"))

# identify the nodes in html document, the output is lists
inflation.table.node <- html_nodes(cpi.page, xpath = ".//table")

# directly load all nodes and use html_table to automatically parse table
html_table(read_html(cpi.url), trim = T, fill = T)
```

```
## [[1]]
## # A tibble: 110 x 14
##     Year   Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
##    <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  2022  281.  284.  288.  289.    NA    NA    NA    NA    NA    NA    NA    NA
##  2  2021  262.  263.  265.  267.  269.  272.  273.  274.  274.  277.  278.  279.
##  3  2020  258.  259.  258.  256.  256.  258.  259.  260.  260.  260.  260.  260.
##  4  2019  252.  253.  254.  256.  256.  256.  257.  257.  257.  257.  257.  257.
##  5  2018  248.  249.  250.  251.  252.  252.  252.  252.  252.  253.  252.  251.
##  6  2017  243.  244.  244.  245.  245.  245.  245.  246.  247.  247.  247.  247.
##  7  2016  237.  237.  238.  239.  240.  241.  241.  241.  241.  242.  241.  241.
##  8  2015  234.  235.  236.  237.  238.  239.  239.  238.  238.  238.  237.  237.
##  9  2014  234.  235.  236.  237.  238.  238.  238.  238.  238.  237.  236.  235.
## 10  2013  230.  232.  233.  233.  233.  234.  234.  234.  234.  234.  233.  233.
## # ... with 100 more rows, and 1 more variable: Ave. <dbl>
```

```
# or load table nodes into html_table
inflation.table <- html_table(inflation.table.node, trim = T, fill = T)
write.csv(inflation.table, file = "CPIHistoricInflationData.csv")


knitr::opts_chunk$set(echo = TRUE)
```