SAQWebScrapingProject.R

danielpeslherbe

2020-07-19

```
##SAQ Analysis

##Install and load necessary packages

##tidyverse for data wrangling
##install.packages("tidyverse", repo = 'https://mac.R-project.org')
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
##rvest for HTML/XML parsing
##install.packages('rvest')
library(rvest)
```

```
## Loading required package: xml2
```

```
##
## Attaching package: 'rvest'
```

```
## The following object is masked from 'package:purrr':
##
##     pluck
```

```
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
##stringr for string manipulation (TBD if necessary)
##install.packages('stringr')
library(stringr)
##rebus for verbose regular expressions (TBD if necessary)
##install.packages('rebus')
library(rebus)
```

```
##
## Attaching package: 'rebus'
```

```
## The following object is masked from 'package:stringr':
```

```
##
##     regex
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
##lubridate for ease of time&date manipulation
##install.packages('lubridate')
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
##testing on single product

url <- 'https://www.saq.com/fr/12824197'
webpage <- read_html(url)

name <- webpage %>%
  html_nodes(".page-title") %>%
  html_text()
name <- str_replace_all(name, "[\r\n]", "")
price <- webpage %>%
  html_nodes('.price') %>%
  html_text()
price <- str_replace_all(price[1], "[\r\n]", " $")
info <- webpage %>%
  html_nodes('.type') %>%
  html_text()
type <- str_replace_all(info[1], "[\r\n]", "")
volume <- str_replace_all(info[2], "[\r\n]", "")
origin <- str_replace_all(info[3], "[\r\n]", "")
region <- str_replace_all(info[4], "[\r\n]", "")

productinfo <- c(name, price, type, volume, origin, region)
product <- data.frame()
product[1,1] <- productinfo[1]
product[1,2] <- productinfo[2]
product[1,3] <- productinfo[3]
product[1,4] <- productinfo[4]
product[1,5] <- productinfo[5]
product[1,6] <- productinfo[6]
head(product)
```

```
##                                                                 V1      V2
## 1             19 Crimes Cabernet-Sauvignon Limestone Coast          18,95 $
##                                                 V3
## 1             Vin rouge
##                                                                          V4
## 1             750                               ml
##                                                 V5
## 1             Australie
```

```
##                                                                 V6
## 1             Australie-Méridionale
```

```
##testing on a page

url <- 'https://www.saq.com/fr/produits?p=1&product_list_order=name_asc'
webpage <- read_html(url)

name <- webpage %>%
  html_nodes(".product-item-name") %>%
  html_text()
name <- str_replace_all(name, space(), "")
price <- webpage %>%
  html_nodes('.price-box') %>%
  html_text()
price <- str_replace_all(price, space(), "")
info <- webpage %>%
  html_nodes('.product-item-identity-format') %>%
  html_text()
info <- str_replace_all(info, "[\r\n]", "")
info <- trimws(info)
type <- str_replace_all(substr(info,1, 25), space(), "")
volume <- str_replace_all(substr(info, 150, 250), space(), "")
origin <- str_replace_all(substr(info, 450, 500), space(), "")

products <- cbind(name, price, type, volume, origin)
head(products)
```

```
##      name                                      price     type       volume
## [1,] "\"Y\"d'Yquem2006"                        "241,00$" "Vinblanc" "750ml"
## [2,] "1000StoriesZinfandelCalifornie2017"      "28,60$"  "Vinrouge" "750ml"
## [3,] "11thHourCellarsPinotNoir"                "17,95$"  "Vinrouge" "750ml"
## [4,] "13thStreetWineryGamay2017"               "19,95$"  "Vinrouge" "750ml"
## [5,] "13thStreetWineryRedPalette2016"          "18,95$"  "Vinrouge" "750ml"
## [6,] "14HandsCabernet-SauvignonColumbiaValley" "15,95$"  "Vinrouge" "750ml"
##      origin
## [1,] "France"
## [2,] "États-Unis"
## [3,] "États-Unis"
## [4,] "Canada"
## [5,] "Canada"
## [6,] "États-Unis"
```

```
##Now this has been cleared beforehand
##Let us apply this on all pages for SAQ products
##we will use iteration to get the same page info far all available pages
##Note that we must verify the number of products available which may change
##on a daily basis depending on offerings and availability

productnumber <- 13993
pagenumber <- ceiling(productnumber/24)

namelist <- list()
pricelist <- list()
typelist <- list()
```

```
volumelist <- list()
originlist <- list()

for (j in 2:pagenumber){
  url <- paste0('https://www.saq.com/fr/produits?p=', j, '&product_list_order=name_asc')
  webpage <- read_html(url)

  price <- webpage %>%
    html_nodes('.price-box') %>%
    html_text()
  price <- str_replace_all(price, space(), "")
    name <- webpage %>%
      html_nodes(".product-item-name") %>%
      html_text()
    name <- str_replace_all(name, space(), "")
    info <- webpage %>%
      html_nodes('.product-item-identity-format') %>%
      html_text()
    info <- str_replace_all(info, "[\r\n]", "")
    info <- trimws(info)
    type <- str_replace_all(substr(info,1, 25), space(), "")
    volume <- str_replace_all(substr(info, 150, 250), space(), "")
    origin <- str_replace_all(substr(info, 450, 500), space(), "")
    namelist[[(j-1)]] <- name
    pricelist[[(j-1)]] <- price
    typelist[[(j-1)]] <- type
    volumelist[[(j-1)]] <- volume
    originlist[[(j-1)]] <- origin

  newproducts <- cbind(name, price, type, volume, origin)
  products <- rbind(products, newproducts)

}

head(products)
```

```
##      name                                       price     type       volume
## [1,] "\"Y\"d'Yquem2006"                         "241,00$" "Vinblanc" "750ml"
## [2,] "1000StoriesZinfandelCalifornie2017"      "28,60$"  "Vinrouge" "750ml"
## [3,] "11thHourCellarsPinotNoir"                "17,95$"  "Vinrouge" "750ml"
## [4,] "13thStreetWineryGamay2017"               "19,95$"  "Vinrouge" "750ml"
## [5,] "13thStreetWineryRedPalette2016"          "18,95$"  "Vinrouge" "750ml"
## [6,] "14HandsCabernet-SauvignonColumbiaValley" "15,95$"  "Vinrouge" "750ml"
##      origin
## [1,] "France"
## [2,] "États-Unis"
## [3,] "États-Unis"
## [4,] "Canada"
## [5,] "Canada"
## [6,] "États-Unis"
```

```
##this gives us a large dataset with product names, prices, type,
##volume & country of origin
##let us create a csv with the information
```

```r
write.csv(products, "saqproducts20200719.csv")
```