wevscraping

promitosh

7/30/2020

For the guacamole recipe page, we already have done this and determined that we need the following selectors: h <- read\_html(“<http://www.foodnetwork.com/recipes/alton-brown/guacamole-recipe-1940609>”) recipe <- h %>% html\_node(".o-AssetTitle\_\_a-HeadlineText“) %>% html\_text() prep\_time <- h %>% html\_node(”.m-RecipeInfo\_\_a-Description–Total“) %>% html\_text() ingredients <- h %>% html\_nodes(”.o-Ingredients\_\_a-Ingredient") %>% html\_text() create a list guacamole <- list(recipe, prep\_time, ingredients)

Since recipe pages from this website follow this general layout, we can use this code to create a function that extracts this information get\_recipe <- function(url){ h <- read\_html(url) recipe <- h %>% html\_node(".o-AssetTitle\_\_a-HeadlineText“) %>% html\_text() prep\_time <- h %>% html\_node(”.m-RecipeInfo\_\_a-Description–Total“) %>% html\_text() ingredients <- h %>% html\_nodes(”.o-Ingredients\_\_a-Ingredient") %>% html\_text() return(list(recipe = recipe, prep\_time = prep\_time, ingredients = ingredients)) }

library(tidyverse) library(rvest) h <- read\_html(url) library(rvest) url <- “<https://en.wikipedia.org/wiki/Murder_in_the_United_States_by_state>” h <- read\_html(url) h<- read\_html(“<http://www.foodnetwork.com/recipes/alton-brown/guacamole-recipe-1940609>”) h <- read\_html(url) url <- “<https://en.wikipedia.org/wiki/Murder_in_the_United_States_by_state>” h <- read\_html(url) get\_recipe(“<http://www.foodnetwork.com/recipes/food-network-kitchen/pancakes-recipe-1913844>”) the functions html\_form(), set\_values(), and submit\_form() permit to query a webpage from R. We learned that tables in html are associated with the table node. list of objects class xml\_node The html\_nodes() function returns a list of objects of class xml\_node. We can see the content of each one using, for example, the html\_text() function. You can see the content for an arbitrarily picked component like this If the content of this object is an html table, we can use the html\_table() function to convert it to a data frame: h<-read\_html(url) library(tidyverse) library(rvest) h <- read\_html(url) library(rvest) url <- “<https://en.wikipedia.org/wiki/Murder_in_the_United_States_by_state>” h <- read\_html(url) class(h) h<- read\_html(“<http://www.foodnetwork.com/recipes/alton-brown/guacamole-recipe-1940609>”) library(rvest) url <- “<https://en.wikipedia.org/wiki/Murder_in_the_United_States_by_state>” h<- read\_html(“<http://www.foodnetwork.com/recipes/alton-brown/guacamole-recipe-1940609>”) ingredients <- h %>% html\_nodes(".o-Ingredients\_\_a-Ingredient“)%>% html\_text() recipe <- h %>% html\_node(”.o-AssetTitle\_\_a-HeadlineText")%>% html\_text()

guacamole <- list(recipe, prep\_time, ingredients) prep\_time <- h %>% html\_node(".m-RecipeInfo\_\_a-Description–Total") %>% html\_text()

h<- read\_html(“<http://www.foodnetwork.com/recipes/alton-brown/guacamole-recipe-1940609>”)

ingredients <- h %>% html\_nodes(".o-Ingredients\_\_a-Ingredient“)%>% html\_text() recipe <- h %>% html\_node(”.o-AssetTitle\_\_a-HeadlineText“)%>% html\_text() prep\_time <- h %>% html\_node(”.m-RecipeInfo\_\_a-Description–Total") %>% html\_text()

guacamole <- list(recipe, prep\_time, ingredients)

get\_recipe <- function(url) { h <- read\_html(url) recipe <- h %>% html\_node(".o-AssetTitle\_\_a-HeadlineText“) %>% html\_text() prep\_time <- h %>% html\_node(”.m-RecipeInfo\_\_a-Description–Total“) %>% html\_text() ingredients <- h %>% html\_nodes(”.o-Ingredients\_\_a-Ingredient") %>% html\_text() return(list(recipe = recipe, prep\_time = prep\_time, ingredients = ingredients)) }

get\_recipe(“<http://www.foodnetwork.com/recipes/food-network-kitchen/pancakes-recipe-1913844>”) url<-“<https://web.archive.org/web/20181024132313/http://www.stevetheump.com/Payrolls.htm>” h <- read\_html(url) nodes <- html\_nodes(h, “table”) html\_table(nodes[[8]]) data(“nodes”) nodes<-html\_nodes(h,“table”) nodes<-html\_table(nodes[[2]]) nodes<-html\_table(nodes[[1]]) tab3<-nodes<-html\_table(nodes[[3]]) tab3<-html\_table(nodes[[3]]) tab1<-nodes<-html\_table(nodes[[1]]) tab4<-nodes<-html\_table(nodes[[4]]) nodes<-html\_nodes(h,“table”) tab4<-nodes<-html\_table(nodes[[4]]) tab\_1<-nodes<-html\_table(nodes[[10]]) tab\_1<-html\_table(nodes[10])

url<- “<https://web.archive.org/web/20181024132313/http://www.stevetheump.com/Payrolls.htm>” h<-read\_html(url) nodes<-html\_nodes(h,“table”) tab\_1<-html\_table(nodes[10]) tab\_1 <- as.data.frame(tab\_1) tab\_1 <- tab\_1[-1, -1] names(tab\_1) <- c(“Team”, “Payroll”, “Average”)

tab\_2<-html\_table(nodes[19]) tab\_2 <- as.data.frame(tab\_2) head(tab\_2) tab\_2<-html\_table(nodes[19]) tab\_2 <- as.data.frame(tab\_2) head(tab\_2) tab\_2<-tab\_2[-1,]

names(tab\_2) <- c(“Team”, “Payroll”, “Average”) str(tab\_1) str(tab\_2)

full\_join(tab\_1,tab\_2, by = “Team”) %>% nrow()

url<- “<https://en.wikipedia.org/w/index.php?title=Opinion_polling_for_the_United_Kingdom_European_Union_membership_referendum&oldid=896735054>” h<-read\_html(url) tab<-html\_nodes(h,“table”) length(tab) length(tab)

tab[[5]] %>% html\_table(fill = TRUE) %>% names()

## R Markdown

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.