Class Activity 17

Your name here

February 12 2024

Group Activity 1

1. Go to the the numbers webpage and extract the table on the front page.

```
session1 <- bow(url = "https://www.the-numbers.com/movie/budgets/all") %>% scrape() %>%
  html_nodes(css = "table") %>%
  html_table()

table_base <- session1 %>% .[[1]]
```

2. Find out the number of pages that contain the movie table, while looking for the changes in the url in the address bar. How does the url changes when you go to the next page?

Answer: The starting count of the movie gets concatenated to the url in increments of 100.

3. Write a for loop to store all the data in multiple pages to a single data frame.

```
library(tidyverse)
library(rvest)

new_urls <- "https://www.the-numbers.com/movie/budgets/all/"

# Create an empty data frame
df1 <- list()

# Generate a vector of indices
index <- seq(1, 6301, 100)
```

```
# Loop through indices, scrape data, and bind the resulting data frames
start_time <- proc.time() # Capture start time
for (i in 1:length(index)) {
  url <- str_glue("{new_urls}{index[i]}")
  webpage <- read_html(url)
  table_new <- html_table(webpage)[[1]] %>%
    janitor::clean_names() %>%
    mutate(across(everything(), as.character))
  df1[[i]] <- table_new
}
```

```
end_time <- proc.time() # Capture end time
end_time - start_time # Calculate duration
   user  system elapsed 
  3.952   0.108  74.550 

df1_final <- do.call(rbind, df1)
df1_final1 <- reduce(df1, dplyr::bind_rows)
```

```
# alternate using map_df()
start_time <- proc.time() # Capture start time

urls <- map(index, function(i) str_glue({new_urls}, {index[i]}))
urls <- map(index, ~str_glue({new_urls}, {.x}))

library(tidyverse)
library(rvest)
library(glue)
library(janitor)

# Assuming 'urls' is already defined
movies_data <- map_df(urls, ~read_html(.x) %>%
                        html_table() %>%
                        .[[1]] %>%
                        janitor::clean_names() %>%
                        mutate(across(everything(), as.character)))
end_time <- proc.time() # Capture end time
end_time - start_time # Calculate duration
   user  system elapsed 
  3.914   0.065  49.112 
```

Group Activity 2

1. Go to the scrapethis and extract the table on the front page.

```
session1 <- bow(url = "https://www.scrapethissite.com/pages/forms/") %>% scrape() %>%
  html_nodes(css = "table") %>%
  html_table()

table_base <- session1 %>% .[[1]]
```

2. Find out the number of pages that contain the movie table, while looking for the changes in the url in the address bar. How does the url changes when you go to the next page?

Answer: The url field has `?page_num=` added with the number of pages running from 1 to 24.

3. Write a for loop to store all the data in multiple pages to a single data frame.

```
library(tidyverse)
library(rvest)

new_urls <- "http://scrapethissite.com/pages/forms/?page_num="

# Create an empty data frame
df2 <- list()

# Generate a vector of indices
```

```r
index <- seq(1, 24)
```

```r
library(tidyverse)
library(rvest)

new_urls <- "http://scrapethissite.com/pages/forms/?page_num="

# Generate a vector of indices
index <- seq(1, 24)
```

```r
df2 <- list()
start_time <- proc.time() # Capture start time

for (i in index) {
  url <- str_glue("{new_urls}{i}")
  webpage <- read_html(url)
  table_new <- html_table(webpage)[[1]] %>%
    janitor::clean_names() %>%
    #set_names(~ifelse(is.na(.) | . == "", paste("V", seq_along(.), sep=""), .)) %>%
    mutate(across(everything(), as.character))
  df2[[i]] <- table_new
}
end_time <- proc.time() # Capture end time
end_time - start_time # Calculate duration
   user  system elapsed
  1.457   0.060   9.050

df2_final <- bind_rows(df2)
df2_final
# A tibble: 582 x 9
   team_name             year  wins  losses ot_losses win_percent goals_for_gf
   <chr>                 <chr> <chr> <chr>  <chr>     <chr>       <chr>
 1 Boston Bruins         1990  44    24     <NA>      0.55        299
 2 Buffalo Sabres        1990  31    30     <NA>      0.388       292
 3 Calgary Flames        1990  46    26     <NA>      0.575       344
 4 Chicago Blackhawks    1990  49    23     <NA>      0.613       284
 5 Detroit Red Wings     1990  34    38     <NA>      0.425       273
 6 Edmonton Oilers       1990  37    37     <NA>      0.463       272
 7 Hartford Whalers      1990  31    38     <NA>      0.388       238
 8 Los Angeles Kings     1990  46    24     <NA>      0.575       340
 9 Minnesota North Stars 1990  27    39     <NA>      0.338       256
10 Montreal Canadiens    1990  39    30     <NA>      0.487       273
# i 572 more rows
# i 2 more variables: goals_against_ga <chr>, x <chr>
```

```r
# alternate using map
urls <- map(index, function(i) str_glue({new_urls}, {i}))
urls <- map(index, ~str_glue("{new_urls}{.x}"))

start_time <- proc.time() # Capture start time
sports_data <- map_df(urls, ~read_html(.x) %>%
                  html_table() %>%
                  .[[1]] %>%
                  janitor::clean_names() %>%
                   mutate(across(everything(), as.character)))

end_time <- proc.time() # Capture end time
end_time - start_time # Calculate duration
   user  system elapsed
  1.463   0.056   8.241

sports_data
# A tibble: 582 x 9
   team_name              year  wins  losses ot_losses win_percent goals_for_gf
   <chr>                  <chr> <chr> <chr>  <chr>     <chr>       <chr>
 1 Boston Bruins          1990  44    24     <NA>      0.55        299
 2 Buffalo Sabres         1990  31    30     <NA>      0.388       292
 3 Calgary Flames         1990  46    26     <NA>      0.575       344
 4 Chicago Blackhawks     1990  49    23     <NA>      0.613       284
 5 Detroit Red Wings      1990  34    38     <NA>      0.425       273
 6 Edmonton Oilers        1990  37    37     <NA>      0.463       272
 7 Hartford Whalers       1990  31    38     <NA>      0.388       238
 8 Los Angeles Kings      1990  46    24     <NA>      0.575       340
 9 Minnesota North Stars 1990  27    39     <NA>      0.338       256
10 Montreal Canadiens     1990  39    30     <NA>      0.487       273
# i 572 more rows
# i 2 more variables: goals_against_ga <chr>, x <chr>
```