Data Scraping

Jonathan Olds

9/27/2020

Loading packages necessary for scraping

```
library(rvest)
```

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

```
library(XML)
```

```
##
## Attaching package: 'XML'

## The following object is masked from 'package:rvest':
##
##     xml
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     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 readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## x purrr::pluck()          masks rvest::pluck()
## x XML::xml()              masks rvest::xml()
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following object is masked from 'package:purrr':
##
##     compact
```

Vector of Years for Data Scraping

```
years = 2008:2019
```

Standard Batting Tables From 2008-2019

```
standard_batting_tables_scrape_function = function(years){
  df = list()
  ## Obtaining Standard Batting Data By Year
  for(i in 1:length(years)){
      url = read_html(paste("https://www.baseball-reference.com/leagues/MLB/",years[i],"-standard-battin
      data = url %>% html_nodes(xpath = '//comment()') %>%    # select comment nodes
          html_text() %>%    # extract comment text
          paste(collapse = '') %>%    # collapse to a single string
          read_html() %>%
          html_node('table') %>%    # select the desired table
          html_table()
      ## Removing Header Rows
      index = seq(0, nrow(data), by=26)
      data = data[-index,]
      df[[i]] = data
  }
 return(df)
}

sb_stats = standard_batting_tables_scrape_function(years)
```

Advanced Batting Tables 2008-2019

```
advanced_batting_tables_scrape_function = function(years){
  df = list()
  ## Obtaining Advanced Batting Data By Year
  for(i in 1:length(years)){
    url = read_html(paste("https://www.baseball-reference.com/leagues/MLB/",years[i],"-advanced-batting
    data = url %>% html_nodes(xpath = '//comment()') %>%    # select comment nodes
      html_text() %>%    # extract comment text
      paste(collapse = '') %>%    # collapse to a single string
      read_html() %>%
      html_node('table') %>%    # select the desired table
      html_table()
    ## Removing Header Rows
    index = seq(0, nrow(data), by=26)
    data = data[-index,]
    df[[i]] = data
  }
  return(df)
}
```

```
ab_stats = advanced_batting_tables_scrape_function(years)
```

Value Batting Tables 2008-2019

```
value_batting_tables_scrape_function = function(years){
  df = list()
  ## Obtaining Value Batting Data By Year
  for(i in 1:length(years)){
    url = read_html(paste("https://www.baseball-reference.com/leagues/MLB/",years[i],"-value-batting.sh
    data = url %>% html_nodes(xpath = '//comment()') %>%    # select comment nodes
      html_text() %>%    # extract comment text
      paste(collapse = '') %>%    # collapse to a single string
      read_html() %>%
      html_node('table') %>%    # select the desired table
      html_table()
    ## Removing Header Rows
    index = seq(0, nrow(data), by=26)
    data = data[-index,]
    df[[i]] = data
  }
  return(df)
}
vb_stats = value_batting_tables_scrape_function(years)
```

Standard Pitching Tables 2008-2019

```
standard_pitching_tables_scrape_function = function(years){
  df = list()
  ## Obtaining Standard Pitching Data By Year
  for(i in 1:length(years)){
    url = read_html(paste("https://www.baseball-reference.com/leagues/MLB/",years[i],"-standard-pitching
    data = url %>% html_nodes(xpath = '//comment()') %>%    # select comment nodes
      html_text() %>%    # extract comment text
      paste(collapse = '') %>%    # collapse to a single string
      read_html() %>%
      html_node('table') %>%    # select the desired table
      html_table()
    ## Removing Header Rows
    index = seq(0, nrow(data), by=26)
    data = data[-index,]
    df[[i]] = data
  }
  return(df)
}
sp_stats = standard_pitching_tables_scrape_function(years)
```

Value Pitching Tables 2008-2019

```
value_pitching_tables_scrape_function = function(years){
  df = list()
  ## Obtaining Advanced Pitching Data By Year
  for(i in 1:length(years)){
    url = read_html(paste("https://www.baseball-reference.com/leagues/MLB/",years[i],"-value-pitching.s
    data = url %>% html_nodes(xpath = '//comment()') %>%    # select comment nodes
      html_text() %>%    # extract comment text
      paste(collapse = '') %>%    # collapse to a single string
      read_html() %>%
      html_node('table') %>%    # select the desired table
      html_table()
    ## Removing Header Rows
    index = seq(0, nrow(data), by=26)
    data = data[-index,]
    df[[i]] = data
  }
  return(df)
}
vp_stats = value_pitching_tables_scrape_function(years)
```

Standard Fielding Tables 2008-2019

```
standard_fielding_tables_scrape_function = function(years){
  df = list()
  ## Obtaining Advanced Pitching Data By Year
  for(i in 1:length(years)){
    url = read_html(paste("https://www.baseball-reference.com/leagues/MLB/",years[i],"-standard-fieldin
    data = url %>% html_nodes(xpath = '//comment()') %>%    # select comment nodes
      html_text() %>%    # extract comment text
      paste(collapse = '') %>%    # collapse to a single string
      read_html() %>%
      html_node('table') %>%    # select the desired table
      html_table()
    ## Removing Header Rows
    index = seq(0, nrow(data), by=26)
    data = data[-index,]
    df[[i]] = data
  }
  return(df)
}
sf_stats = standard_fielding_tables_scrape_function(years)
```

Removing "*,#,+" from Player Names in All Datasets

```
remove_junk_function = function(data){
  for(i in 1:12){
  x = data[[i]]$Name
  for(j in 1:length(x)){
    x[j] = gsub("[*]", "", x[j])
  }
  for(j in 1:length(x)){
    x[j] = gsub("[#]", "", x[j])
  }
  for(j in 1:length(x)){
    x[j] = gsub("[+]", "", x[j])
  }
  for(j in 1:length(x)){
    x[j] = stringi::stri_trans_general(x[j], "Latin-ASCII")
  }
  data[[i]]$Name = x
  }
  data
}

sb_stats = remove_junk_function(sb_stats)
ab_stats = remove_junk_function(ab_stats)
vb_stats = remove_junk_function(vb_stats)
sp_stats = remove_junk_function(sp_stats)
vp_stats = remove_junk_function(vp_stats)
sf_stats = remove_junk_function(sf_stats)
```

Creating 2019 Tables

```
sb_2019 = sb_stats[[12]]
ab_2019 = ab_stats[[12]]
vb_2019 = vb_stats[[12]]
sp_2019 = sp_stats[[12]]
vp_2019 = vp_stats[[12]]
sf_2019 = sf_stats[[12]]
```

Collecting Top 100 Players List 2011-2020

```
top100_2011 = read_csv("Top 100 Player Datasets/2011 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2012 = read_csv("Top 100 Player Datasets/2012 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2013 = read_csv("Top 100 Player Datasets/2013 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2014 = read_csv("Top 100 Player Datasets/2014 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2015 = read_csv("Top 100 Player Datasets/2015 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2016 = read_csv("Top 100 Player Datasets/2016 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2017 = read_csv("Top 100 Player Datasets/2017 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2018 = read_csv("Top 100 Player Datasets/2018 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2019 = read_csv("Top 100 Player Datasets/2019 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

```
top100_2020 = read_csv("Top 100 Player Datasets/2020 Top 100 Players List.csv")
```

```
## Parsed with column specification:
## cols(
##   `Top 100 Rank` = col_double(),
##   Name = col_character()
## )
```

Merging Batting Datasets

```
full_batting_data = list()
merge_batting_datasets_function = function(dataset){
for(i in 1:12){
dataset[[i]] = join(sb_stats[[i]], ab_stats[[i]], by = "Name", match = "first")
dataset[[i]] = join(dataset[[i]], vb_stats[[i]], by = "Name", type = "full", match = "first")
}
  dataset
}

full_batting_data = merge_batting_datasets_function(full_batting_data)
```

Merging Pitching Datasets

```
full_pitching_data = list()
merge_pitching_datasets_function = function(dataset){
  for(i in 1:12){
    dataset[[i]] = join(sp_stats[[i]], vp_stats[[i]], by = "Name", type = "full", match = "first")
  }
  dataset
}

full_pitching_data = merge_pitching_datasets_function(full_pitching_data)
```

Renaming Fielding Data

```
full_fielding_data = sf_stats
```