HW04 - DS6306-402

Dan Crouthamel

June 03, 2019

Problem 1 - Harry Potter Cast

In the IMDB, there are listings of full cast members for movies. Navigate to http://www.imdb.com/title/tt1201607/fullcredits?ref_=tt_ql_1. Feel free to View Source to get a good idea of what the page looks like in code.

Scrape the page with any R package that makes things easy for you. Of particular interest is the table of the Cast in order of crediting. Please scrape this table (you might have to fish it out of several of the tables from the page) and make it a data.frame() of the Cast in your R environment

```
# Read info
library('rvest')
```

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

```
library('tidyr')
url <- 'https://www.imdb.com/title/tt1201607/fullcredits?ref_=tt_ql_1'
site <- read_html(url)

# If you use the browser dev tools, you'll see there is a table with class of cast_list
# I foudn the following to helpful with selectors -> http://flukeout.github.io/
node <- html_node(site, "table.cast_list")
table <- html_table(node, header = FALSE)

# Use View(table) to see results. We see the first row is empty, so nuke it
table <- table[-1, ]

# The first and third columns column don't look useful, so get rid of them
table$X1 <- NULL
table$X3 <- NULL

# Give more meaningful names to the other columns
names(table) <- c("Actor", "Character")

# There appear to be rows that say "Rest of cast" ... get rid of them
table <- table[-(table$Actor=="Rest of cast listed alphabetically:"),]

# Fix Warwick as per instructions
table[table$Actor=="Warwick Davis",][2]="Griphook / Professor Filius Flitwick"

# Split first column into First and Last. Middle name should be included in First.
# https://regex101.com, very useful for testing, learning
# This is basically finding the last space separator in name, and telling the
# separate function to split/separate on that.
table <- table %>% separate(Actor, c("FirstName", "Surname"),
                            "[ ](?=[^ ]+$)", extra="merge")

# Show first 10 rows
```

```r
head(table, 10)
```

```
##    FirstName   Surname                              Character
## 3    Michael    Gambon           Professor Albus Dumbledore
## 4       Alan   Rickman              Professor Severus Snape
## 5     Daniel Radcliffe                          Harry Potter
## 6     Rupert     Grint                           Ron Weasley
## 7       Emma    Watson                      Hermione Granger
## 8     Evanna     Lynch                         Luna Lovegood
## 9   Domhnall   Gleeson                           Bill Weasley
## 10  Clémence     Poésy                        Fleur Delacour
## 11   Warwick     Davis Griphook / Professor Filius Flitwick
## 12      John      Hurt                              Ollivander
```

Problem 2 - ESPN

On the ESPN website, there are statistics of each NBA player. Navigate to the San Antonio Spurs current statistics (likely http://www.espn.com/nba/team/stats/_/name/sa/san-antonio-spurs). You are interested in the Shooting Statistics table.

Scrape the page with any R package that makes things easy for you. There are a few tables on the page, so make sure you are targeting specifically the Shooting Statistics table.

```
# Read info
library('rvest')
library('tidyr')
url <- 'http://www.espn.com/nba/team/stats/_/name/sa/san-antonio-spurs'
site <- read_html(url)

# There were a couple of different ways I could have done this.
# I decided on grabbing all tables, looking at them and then
# figuring out which ones to use.
# Names = Table 6, Stats = Table 8
tables <- html_nodes(site,"table")
playerTable <- tables[6]
playerStats <- tables[8]

# Create data frames, include headers, use View(mainTable), make sure it looks good
playerTable <- as.data.frame(html_table(playerTable, header = TRUE))
playerStats <- as.data.frame(html_table(playerStats, header = TRUE))
#mainTable <- as.data.frame(c(playerTable,playerStats), header = TRUE)
mainTable <- cbind(playerTable,playerStats)

# Delete Totals row, which is the last row
mainTable <- mainTable[1:(nrow(mainTable)-1),]

# Split name column into Name and Position
mainTable <- mainTable %>% separate(Name, c("Name", "Position"),
                                    "[ ](?=[^ ]+$)", extra="merge")

# Assignment asks that appropriate columns are numeric, but they already are!
str(mainTable)
```

```
## 'data.frame':    13 obs. of  16 variables:
##  $ Name    : chr  "DeMar DeRozan" "LaMarcus Aldridge" "Derrick White" "Rudy Gay" ...
##  $ Position: chr  "SG" "C" "PG" "SF" ...
##  $ FGM     : num  8.3 7.9 5.9 4 3.9 3.3 2 2 1 1.2 ...
##  $ FGA     : num  17 17.3 10.7 10 8 5.1 5.4 6.1 3 2 ...
##  $ FG.     : num  48.7 45.5 54.7 40 48.2 63.9 36.8 32.6 33.3 60 ...
##  $ X3PM    : num  0 0.4 0.7 1.1 2.1 0 1.1 0.4 0.6 0 ...
##  $ X3PA    : num  0.1 1.6 2.4 2.7 4.4 0.1 3 3.1 2.2 0.2 ...
##  $ X3P.    : num  0 27.3 29.4 42.1 48.4 0 38.1 13.6 27.3 0 ...
##  $ FTM     : num  5.4 3.9 2.7 2 0.9 0.7 0.7 0.9 0.6 0.2 ...
##  $ FTA     : num  6.3 4.7 3.7 2.4 1.3 1.3 0.9 1.4 1 0.4 ...
##  $ FT.     : num  86.4 81.8 73.1 82.4 66.7 55.6 83.3 60 60 50 ...
##  $ X2PM    : num  8.3 7.4 5.1 2.9 1.7 3.3 0.9 1.6 0.4 1.2 ...
##  $ X2PA    : num  16.9 15.7 8.3 7.3 3.6 5 2.4 3 0.8 1.8 ...
##  $ X2P.    : num  49.2 47.3 62.1 39.2 48 65.7 35.3 52.4 50 66.7 ...
##  $ SC.EFF  : num  1.29 1.16 1.41 1.11 1.34 ...
```

```
##  $ SH.EFF  : num  0.49 0.47 0.58 0.46 0.62 0.64 0.47 0.36 0.43 0.6 ...
```

```r
# Color BarChart
library(ggplot2)
fgppgPlot <-ggplot(data=mainTable, aes_string("Name", "`FG.`", fill = "Position"))
fgppgPlot +
  geom_bar(stat = "identity") +
  coord_flip() +
  ylab("Field Goal Percentage") +
  xlab("Player") +
  ggtitle("San Antonio Shooting Percentage per Game 2018-19") +
  theme(plot.title = element_text(hjust = 0.5))
```