R worksheet5 Group 7

2023-12-21

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

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:

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(polite)
library(xml2)
library(magrittr)
library(rvest)
library(httr)
```

```
#Movie Guide
#m1 - Breaking Bad
#m2 - Game of Thrones
#m3 - Arcane
#m4 - Death Note
#m5 - Better Call Saul

polite::use_manners(save_as = "polite_scrape.R")
```

```
## v Setting active project to 'D:/New folder'
```

```r
url <- 'https://www.imdb.com/chart/toptv/?ref=nv_tvv_250'
session <- bow(url, user_agent = "Educational")

# Create empty vectors
title <- character(0)
rank <- 1:50
rating <- character(0)
numVoteCount <- character(0)
numEpisodes <- character(0)
numYear <- character(0)
numSplit <- c()

# Scraping the Title.
title <- scrape(session) %>%
  html_nodes('h3.ipc-title__text') %>%
  html_text
title <- title[2:51]

# Scraping the rating.
rating <- scrape(session) %>%
  html_nodes('span.ratingGroup--imdb-rating') %>%
  html_text
rating <- rating[1:50]

# Scraping the Vote Count.
numVoteCount <- scrape(session) %>%
  html_nodes('span.ipc-rating-star--voteCount') %>%
  html_text
numVoteCount <- numVoteCount[1:50]

# Scraping the span: Year, Number of Episodes, and Year Released.
numSplit <- scrape(session) %>%
  html_nodes('span.sc-43986a27-8') %>%
  html_text

# Get the number of Episodes
retrievedEpisode <- character()
for (i in seq(2, length(numSplit), by = 3)) {
  currentEpisode <- numSplit[i]
  retrievedEpisode <- c(retrievedEpisode, currentEpisode)
}
numEpisodes <- retrievedEpisode[1:50]

# Get the year it was released
retrievedYear <- character()
for (i in seq(1, length(numSplit), by = 3)) {
  currentYear <- numSplit[i]
  retrievedYear <- c(retrievedYear, currentYear)
}
numYear <- retrievedYear[1:50]

# Update Year, Rating, and Vote Count
updateYear <- sub("^(\\d{4}).", "\\1", numYear)
```

```
updateRating <- sub("^(\\d+\\.\\d+).", "\\1", rating)
updateVoteCount <- sub(".?\\s\\((\\S+)\\).*", "\\1", numVoteCount)

# Extract Title
splitTitle <- gsub("\\d+\\.\\s", "", title)


wholeDF <- data.frame(rank, splitTitle, updateRating, updateVoteCount, numEpisodes, updateYear)
colnames(wholeDF) <- c("Rank", "Title", "Rating", "Vote Count", "Number of Episodes", "Year Released")


View(wholeDF)
```

```
url_m1 <- 'https://www.imdb.com/title/tt0903747/reviews?spoiler=hide&sort=curated&dir=desc&ratingFilter=
url_m2 <- 'https://www.imdb.com/title/tt0944947/reviews?spoiler=hide&sort=curated&dir=desc&ratingFilter=
url_m3 <- 'https://www.imdb.com/title/tt11126994/reviews?spoiler=hide&sort=curated&dir=desc&ratingFilte
url_m4 <- 'https://www.imdb.com/title/tt0877057/reviews?spoiler=hide&sort=curated&dir=desc&ratingFilter=
url_m5 <- 'https://www.imdb.com/title/tt3032476/reviews?spoiler=hide&sort=curated&dir=desc&ratingFilter=
tvShowDateURL <- 'https://www.imdb.com/chart/toptv/?ref_=nv_tvv_250'
```

```
session_m1 <- bow(url_m1,
                  user_agent = "Educational")
session_m2 <- bow(url_m2,
                  user_agent = "Educational")
session_m3 <- bow(url_m3,
                  user_agent = "Educational")
session_m4 <- bow(url_m4,
                  user_agent = "Educational")
session_m5 <- bow(url_m5,
                  user_agent = "Educational")
session_m6 <- bow(tvShowDateURL,
                  user_agent = "Educational")

session_m1
```

```
## <polite session> https://www.imdb.com/title/tt0903747/reviews?spoiler=hide&sort=curated&dir=desc&rati
##     User-agent: Educational
##     robots.txt: 34 rules are defined for 2 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session_m2
```

```
## <polite session> https://www.imdb.com/title/tt0944947/reviews?spoiler=hide&sort=curated&dir=desc&rati
##     User-agent: Educational
##     robots.txt: 34 rules are defined for 2 bots
```

```
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session_m3
```

```
## <polite session> https://www.imdb.com/title/tt11126994/reviews?spoiler=hide&sort=curated&dir=desc&ra
##     User-agent: Educational
##     robots.txt: 34 rules are defined for 2 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session_m4
```

```
## <polite session> https://www.imdb.com/title/tt0877057/reviews?spoiler=hide&sort=curated&dir=desc&rati
##     User-agent: Educational
##     robots.txt: 34 rules are defined for 2 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session_m5
```

```
## <polite session> https://www.imdb.com/title/tt3032476/reviews?spoiler=hide&sort=curated&dir=desc&rati
##     User-agent: Educational
##     robots.txt: 34 rules are defined for 2 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session_m6
```

```
## <polite session> https://www.imdb.com/chart/toptv/?ref_=nv_tvv_250
##     User-agent: Educational
##     robots.txt: 34 rules are defined for 2 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
reviewerName_m1 <- character(0)
dateReviewed_m1 <- character(0)
userRating_m1 <- character(0)
titleReview_m1 <- character(0)
textReview_m1 <- character(0)

reviewerName_m2 <- character(0)
dateReviewed_m2 <- character(0)
userRating_m2 <- character(0)
titleReview_m2 <- character(0)
textReview_m2 <- character(0)

reviewerName_m3 <- character(0)
dateReviewed_m3 <- character(0)
userRating_m3 <- character(0)
titleReview_m3 <- character(0)
```

```
textReview_m3 <- character(0)

reviewerName_m4 <- character(0)
dateReviewed_m4 <- character(0)
userRating_m4 <- character(0)
titleReview_m4 <- character(0)
textReview_m4 <- character(0)

reviewerName_m5 <- character(0)
dateReviewed_m5 <- character(0)
userRating_m5 <- character(0)
titleReview_m5 <- character(0)
textReview_m5 <- character(0)

tvShowTitle <- character(0)
tvShowDates <- character(0)

#Breaking Bad
tv_m1 <- scrape(session_m1) %>%
  html_elements('div.lister-item')

reviewerName_m1 <- tv_m1 %>%
  html_nodes('span.display-name-link') %>%
  html_text()

dateReviewed_m1 <- tv_m1 %>%
  html_nodes('span.review-date') %>%
  html_text()

userRating_m1 <- tv_m1 %>%
  html_node(".rating-other-user-rating") %>%
  html_text()

titleReview_m1 <- tv_m1 %>%
  html_nodes('a.title') %>%
  html_text()

textReview_m1 <- tv_m1 %>%
  html_nodes('div.text.show-more__control') %>%
  html_text()

DF_m1 <- data.frame(userRating_m1, dateReviewed_m1, reviewerName_m1, titleReview_m1, textReview_m1)
colnames(DF_m1) <- c("User Rating", "Date Reviewed", "Reviewer Name", "Title Review", "Text Review")

View(DF_m1)
```

```
#Game of Thrones

tv_m2 <- scrape(session_m2) %>%
  html_elements('div.lister-item')

reviewerName_m2 <- tv_m2 %>%
  html_nodes('span.display-name-link') %>%
```

```
  html_text()

dateReviewed_m2 <- tv_m2 %>%
  html_nodes('span.review-date') %>%
  html_text()

userRating_m2 <- tv_m2 %>%
  html_node(".rating-other-user-rating") %>%
  html_text()

titleReview_m2 <- tv_m2 %>%
  html_nodes('a.title') %>%
  html_text()

textReview_m2 <- tv_m2 %>%
  html_nodes('div.text.show-more__control') %>%
  html_text()


DF_m2 <- data.frame(userRating_m2, dateReviewed_m2, reviewerName_m2, titleReview_m2, textReview_m2)
colnames(DF_m2) <- c("User Rating", "Date Reviewed", "Reviewer Name", "Title Review", "Text Review")
View(DF_m2)
```

```
#Arcane
tv_m3 <- scrape(session_m3) %>%
  html_elements('div.lister-item')

reviewerName_m3 <- tv_m3 %>%
  html_nodes('span.display-name-link') %>%
  html_text()

dateReviewed_m3 <- tv_m3 %>%
  html_nodes('span.review-date') %>%
  html_text()

userRating_m3 <- tv_m3 %>%
  html_node(".rating-other-user-rating") %>%
  html_text()

titleReview_m3 <- tv_m3 %>%
  html_nodes('a.title') %>%
  html_text()

textReview_m3 <- tv_m3 %>%
  html_nodes('div.text.show-more__control') %>%
  html_text()

DF_m3 <- data.frame(userRating_m3, dateReviewed_m3, reviewerName_m3, titleReview_m3, textReview_m3)
colnames(DF_m3) <- c("User Rating", "Date Reviewed", "Reviewer Name", "Title Review", "Text Review")

View(DF_m3)
```

```r
#Death Note

tv_m4 <- scrape(session_m4) %>%
  html_elements('div.lister-item')

reviewerName_m4 <- tv_m4 %>%
  html_nodes('span.display-name-link') %>%
  html_text()

dateReviewed_m4 <- tv_m4 %>%
  html_nodes('span.review-date') %>%
  html_text()

userRating_m4 <- tv_m4 %>%
  html_node(".rating-other-user-rating") %>%
  html_text()

titleReview_m4 <- tv_m4 %>%
  html_nodes('a.title') %>%
  html_text()

textReview_m4 <- tv_m4 %>%
  html_nodes('div.text.show-more__control') %>%
  html_text()

DF_m4 <- data.frame(userRating_m4, dateReviewed_m4, reviewerName_m4, titleReview_m4, textReview_m4)
colnames(DF_m4) <- c("User Rating", "Date Reviewed", "Reviewer Name", "Title Review", "Text Review")

View(DF_m4)
```

```r
#Better Call Saul

tv_m5 <- scrape(session_m5) %>%
  html_elements('div.lister-item')

reviewerName_m5 <- tv_m5 %>%
  html_nodes('span.display-name-link') %>%
  html_text()

dateReviewed_m5 <- tv_m5 %>%
  html_nodes('span.review-date') %>%
  html_text()

userRating_m5 <- tv_m5 %>%
  html_node(".rating-other-user-rating") %>%
  html_text()

titleReview_m5 <- tv_m5 %>%
  html_nodes('a.title') %>%
  html_text()

textReview_m5 <- tv_m5 %>%
  html_nodes('div.text.show-more__control') %>%
```

```
  html_text()

DF_m5 <- data.frame(userRating_m5, dateReviewed_m5, reviewerName_m5, titleReview_m5, textReview_m5)
colnames(DF_m5) <- c("User Rating", "Date Reviewed", "Reviewer Name", "Title Review", "Text Review")

View(DF_m5)
```

```
tvShows <- scrape(session_m6) %>%
  html_elements('ul.ipc-metadata-list')


tvShowTitle <- tvShows %>%
  html_nodes('h3.ipc-title__text') %>%
  html_text()

tvShowDates <- tvShows %>%
  html_nodes('div.sc-43986a27-7') %>%
  html_text()

tvShowDates <- substr(tvShowDates, 1, 4)
tvShowDates
```

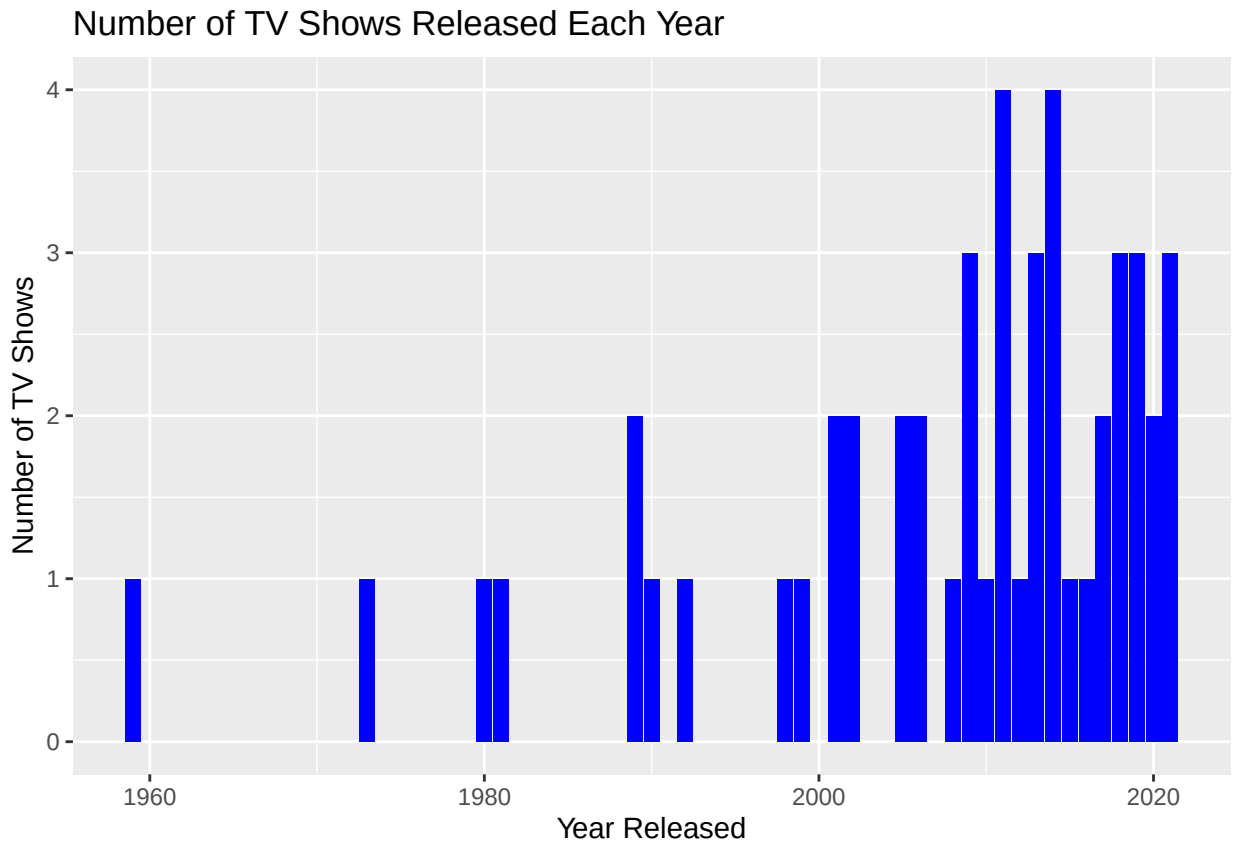
```
##   [1] "2008" "2016" "2006" "2001" "2019" "2002" "2005" "2017" "1999" "2014"
##  [11] "1980" "2019" "2011" "1973" "2018" "2013" "2009" "2020" "2009" "1959"
##  [21] "2010" "2017" "1992" "2013" "2020" "2005" "2021" "2001" "2015" "2011"
##  [31] "2002" "2011" "2021" "2006" "1981" "2011" "1990" "2014" "1989" "2021"
##  [41] "1989" "2018" "2014" "1998" "2012" "2013" "2014" "2019" "2018" "2009"
##  [51] "1994" "2013" "2015" "2015" "2005" "2021" "1969" "1999" "2000" "1985"
##  [61] "1999" "1975" "2014" "2011" "1995" "1999" "1989" "1990" "1989" "2015"
##  [71] "2003" "1989" "2011" "1976" "2023" "2023" "2014" "1997" "2020" "2019"
##  [81] "2001" "2013" "2019" "2005" "1997" "2019" "2011" "2020" "2017" "2019"
##  [91] "2016" "2022" "2004" "2002" "2010" "1989" "2003" "2004" "2021" "2015"
## [101] "2022" "1997" "2007" "2003" "2016" "1987" "2017" "1984" "2004" "1988"
## [111] "2019" "2016" "2004" "2006" "2021" "2002" "2010" "2012" "2005" "2021"
## [121] "2013" "2007" "2016" "2020" "2004" "2016" "1995" "1987" "2016" "2009"
## [131] "2004" "1993" "2006" "2006" "2015" "2019" "2021" "2015" "2015" "2011"
## [141] "1986" "2010" "1989" "2018" "2017" "1986" "2022" "1988" "2014" "2019"
## [151] "2014" "1989" "2009" "1987" "1990" "2010" "2017" "2018" "2021" "2019"
## [161] "1980" "2018" "2003" "2012" "2010" "2010" "1960" "2022" "2018" "2018"
## [171] "2015" "1972" "1993" "2019" "2001" "2003" "2015" "2020" "2007" "1986"
## [181] "1995" "2003" "2017" "2019" "2000" "2012" "2011" "1986" "2022" "2004"
## [191] "1999" "2012" "2015" "2006" "2009" "2004" "1997" "2022" "2019" "2020"
## [201] "2016" "2010" "2001" "1998" "2001" "2014" "2010" "2010" "2008" "2014"
## [211] "1995" "2011" "2016" "1997" "2017" "2005" "2001" "2015" "2004" "2011"
## [221] "2014" "2022" "2010" "2022" "2003" "1999" "2015" "1998" "1999" "2016"
## [231] "2013" "2015" "2014" "2018" "2018" "2013" "2021" "2009" "2015" "2005"
## [241] "2004" "2014" "2002" "1955" "2008" "2018" "2022" "2009" "2010" "2016"
```

```
tvShowDF <- data.frame(tvShowTitle,tvShowDates)
colnames(tvShowDF) <- c("TV Show Title","Year_Released")


tvShowDF <- head(tvShowDF,50)
```

```
View(tvShowDF)
tvShowDF$Year_Released <- substr(tvShowDF$Year_Released, 1, 4)
```

```
tvShowDF$Year_Released <- as.numeric(tvShowDF$Year_Released)
ggplot(tvShowDF, aes(x = Year_Released)) +
  geom_bar(stat = "count", fill = "blue") +
  labs(title = "Number of TV Shows Released Each Year",
       x = "Year Released",
       y = "Number of TV Shows")
```




```
cat("The year(s) that have the most number of TV shows released are year 2011 and year 2014")
```

```
## The year(s) that have the most number of TV shows released are year 2011 and year 2014
```

```
url1 <- 'https://www.amazon.com/Memory-Stick-Portable-Thumb-Keychain/dp/B0CDKS2TGW/ref=sr_1_1_sspa?qid=
url2 <- 'https://www.amazon.com/GTIOT-USB-Computer-Portable-High-Speed/dp/B0CPXYBK1V/ref=sr_1_2_sspa?qi
url3 <- 'https://www.amazon.com/SAMSUNG-Internal-Expansion-MZ-V9P2T0B-AM/dp/B0BHJJ9Y77/ref=sr_1_4?qid=1
session1 <- bow(url1,
                user_agent = "Educational")
session2 <- bow(url2,
                 user_agent = "Educational")
session3 <- bow(url3,
                 user_agent = "Educational")
session1
```

```
## <polite session> https://www.amazon.com/Memory-Stick-Portable-Thumb-Keychain/dp/B0CDKS2TGW/ref=sr_1_
##     User-agent: Educational
##     robots.txt: 154 rules are defined for 4 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session2
```

```
## <polite session> https://www.amazon.com/GTIOT-USB-Computer-Portable-High-Speed/dp/B0CPXYBK1V/ref=sr_
##     User-agent: Educational
##     robots.txt: 154 rules are defined for 4 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session3
```

```
## <polite session> https://www.amazon.com/SAMSUNG-Internal-Expansion-MZ-V9P2T0B-AM/dp/B0BHJJ9Y77/ref=s
##     User-agent: Educational
##     robots.txt: 154 rules are defined for 4 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
amazon1 <- scrape(session1) %>%
  html_elements('div.cm-cr-dp-review-list')

amazonPrice1 <- scrape(session1) %>%
  html_node('span.a-offscreen') %>%
  html_text()

amazonDescription1 <- scrape(session1) %>%
  html_node('div.a-spacing-medium') %>%
  html_text()


amazonRatings1 <- scrape(session1) %>%
  html_node('span.a-icon-alt') %>%
  html_text()


amazonReviews1 <- scrape(session1) %>%
  html_node('span.review-text') %>%
  html_text()


##Number 2


amazonPrice2 <- scrape(session2) %>%
  html_node('span.a-offscreen') %>%
  html_text()

amazonDescription2 <- scrape(session2) %>%
  html_node('div.a-spacing-medium') %>%
```

```
  html_text()

amazonRatings2 <- scrape(session2) %>%
  html_node('span.a-icon-alt') %>%
  html_text()


amazonReviews2 <- scrape(session2) %>%
  html_node('span.review-text') %>%
  html_text()


#Number 3


amazonPrice3 <- scrape(session3) %>%
  html_node('span.a-offscreen') %>%
  html_text()

amazonDescription3 <- scrape(session3) %>%
  html_node('div.a-spacing-medium') %>%
  html_text()

amazonRatings3 <- scrape(session3) %>%
  html_node('span.a-icon-alt') %>%
  html_text()


amazonReviews3 <- scrape(session3) %>%
  html_node('span.review-text') %>%
  html_text()


Price <- c(amazonPrice1, amazonPrice2, amazonPrice3)
Description <- c(amazonDescription1, amazonDescription2, amazonDescription3)
Ratings <- c(amazonRatings1, amazonRatings2, amazonRatings3)
Reviews <- c(amazonReviews1, amazonReviews2, amazonReviews3)


amazonDF <- data.frame(Price, Description, Ratings, Reviews)
View(amazonDF)
```