RWorksheet_5a

Jalando-on, Nandin, Palabrica

2024-11-27

IMDB

```r
library(rvest)
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
```

```r
library(stringr)
library(polite)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(knitr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()          masks stats::filter()
## x kableExtra::group_rows() masks dplyr::group_rows()
## x readr::guess_encoding()  masks rvest::guess_encoding()
## x dplyr::lag()             masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
link = "https://www.imdb.com/chart/toptv/"
page = read_html(link)
session <- bow(link, user_agent = "Educational")
        session
```

```
## <polite session> https://www.imdb.com/chart/toptv/
##     User-agent: Educational
##     robots.txt: 35 rules are defined for 3 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```r
nam <- page %>% html_nodes(".ipc-title__text") %>% html_text()
name <- nam[!grepl("Top 250 TV Shows|IMDb Charts|Recently viewed|More to explore", nam, ignore.case = T
name
```

```
##  [1] "1. Breaking Bad"
##  [2] "2. Planet Earth II"
##  [3] "3. Planet Earth"
##  [4] "4. Band of Brothers"
##  [5] "5. Chernobyl"
##  [6] "6. The Wire"
##  [7] "7. Avatar: The Last Airbender"
##  [8] "8. Blue Planet II"
##  [9] "9. The Sopranos"
## [10] "10. Cosmos: A Spacetime Odyssey"
## [11] "11. Cosmos"
## [12] "12. Our Planet"
## [13] "13. Game of Thrones"
## [14] "14. Bluey"
## [15] "15. The World at War"
## [16] "16. Fullmetal Alchemist: Brotherhood"
## [17] "17. Rick and Morty"
## [18] "18. Life"
## [19] "19. The Last Dance"
## [20] "20. The Twilight Zone"
## [21] "21. The Vietnam War"
## [22] "22. Sherlock"
## [23] "23. Attack on Titan"
## [24] "24. Batman: The Animated Series"
## [25] "25. Arcane"
```

```r
rank <- str_extract(name, "^\\d+\\.")
rank
```

```
##  [1] "1."  "2."  "3."  "4."  "5."  "6."  "7."  "8."  "9."  "10." "11." "12."
## [13] "13." "14." "15." "16." "17." "18." "19." "20." "21." "22." "23." "24."
## [25] "25."
```

```r
title <- str_replace(name, "^\\d+\\.", "")
title
```

```
##  [1] " Breaking Bad"                     " Planet Earth II"
##  [3] " Planet Earth"                     " Band of Brothers"
##  [5] " Chernobyl"                        " The Wire"
##  [7] " Avatar: The Last Airbender"       " Blue Planet II"
##  [9] " The Sopranos"                     " Cosmos: A Spacetime Odyssey"
## [11] " Cosmos"                           " Our Planet"
## [13] " Game of Thrones"                  " Bluey"
## [15] " The World at War"                 " Fullmetal Alchemist: Brotherhood"
## [17] " Rick and Morty"                   " Life"
## [19] " The Last Dance"                   " The Twilight Zone"
## [21] " The Vietnam War"                  " Sherlock"
## [23] " Attack on Titan"                  " Batman: The Animated Series"
## [25] " Arcane"
```

```
yea = page %>% html_nodes(".sc-5bc66c50-6.OOdsw.cli-title-metadata-item") %>% html_text()
year <- str_extract_all(yea, "\\b\\d{4}(?:-\\d{4})?\\b") %>% unlist()
year
```

```
## NULL
```

```
rating = page %>% html_nodes(".ipc-rating-star--rating") %>% html_text()
rating
```

```
##  [1] "9.5" "9.5" "9.4" "9.4" "9.3" "9.3" "9.3" "9.3" "9.2" "9.2" "9.3" "9.2"
## [13] "9.2" "9.3" "9.2" "9.1" "9.1" "9.1" "9.0" "9.0" "9.1" "9.1" "9.1" "9.0"
## [25] "9.0"
```

```
episode <- page %>% html_nodes(".sc-5bc66c50-6.OOdsw.cli-title-metadata-item") %>%
html_text()
episodes <- str_extract_all(episode, "\\b\\d+ eps\\b") %>% unlist()
episodes
```

```
## NULL
```

```
vote = page %>% html_nodes(".ipc-rating-star--voteCount") %>% html_text()
vote
```

```
##  [1] " (2.2M)" " (163K)" " (224K)" " (547K)" " (910K)" " (392K)" " (391K)"
##  [8] " (49K)"  " (500K)" " (132K)" " (46K)"  " (54K)"  " (2.4M)" " (34K)"
## [15] " (31K)"  " (210K)" " (629K)" " (44K)"  " (160K)" " (97K)"  " (30K)"
## [22] " (1M)"   " (565K)" " (123K)" " (332K)"
```

```
urls <- c("https://www.imdb.com/title/tt0903747/?ref_=chttvtp_i_1",
          "https://www.imdb.com/title/tt5491994/?ref_=chttvtp_i_2",
          "https://www.imdb.com/title/tt0795176/?ref_=chttvtp_i_3",
          "https://www.imdb.com/title/tt0185906/?ref_=chttvtp_i_4",
          "https://www.imdb.com/title/tt7366338/?ref_=chttvtp_i_5",
          "https://www.imdb.com/title/tt0306414/?ref_=chttvtp_i_6",
          "https://www.imdb.com/title/tt0417299/?ref_=chttvtp_i_7",
          "https://www.imdb.com/title/tt6769208/?ref_=chttvtp_i_8",
          "https://www.imdb.com/title/tt0141842/?ref_=chttvtp_i_9",
          "https://www.imdb.com/title/tt2395695/?ref_=chttvtp_i_10",
          "https://www.imdb.com/title/tt0081846/?ref_=chttvtp_i_11",
          "https://www.imdb.com/title/tt9253866/?ref_=chttvtp_i_12",
          "https://www.imdb.com/title/tt0944947/?ref_=chttvtp_i_13",
          "https://www.imdb.com/title/tt7678620/?ref_=chttvtp_i_14",
          "https://www.imdb.com/title/tt0071075/?ref_=chttvtp_i_15",
          "https://www.imdb.com/title/tt1355642/?ref_=chttvtp_i_16",
          "https://www.imdb.com/title/tt2861424/?ref_=chttvtp_i_17",
          "https://www.imdb.com/title/tt1533395/?ref_=chttvtp_i_18",
          "https://www.imdb.com/title/tt8420184/?ref_=chttvtp_i_19",
          "https://www.imdb.com/title/tt0052520/?ref_=chttvtp_i_20",
          "https://www.imdb.com/title/tt1877514/?ref_=chttvtp_i_21",
          "https://www.imdb.com/title/tt1475582/?ref_=chttvtp_i_22",
          "https://www.imdb.com/title/tt2560140/?ref_=chttvtp_i_23",
          "https://www.imdb.com/title/tt0103359/?ref_=chttvtp_i_24",
          "https://www.imdb.com/title/tt0386676/?ref_=chttvtp_i_25")
```

```
user_reviews <- vector("numeric", length(urls))
critic_reviews <- vector("numeric", length(urls))
for (i in seq_along(urls)) {

  session <- bow(urls[i], user_agent = "Educational")

  webpage <- scrape(session)

  reviewz <- webpage %>% html_nodes(".score") %>% html_text()

  if (length(reviewz) >= 2) {

    user_reviews[i] <- ifelse(grepl("K", reviewz[1]),
                              as.numeric(gsub("K", "", reviewz[1])) * 1000,
                              as.numeric(reviewz[1]))
    critic_reviews[i] <- as.numeric(reviewz[2])
  } else {
    user_reviews[i] <- NA
    critic_reviews[i] <- NA
  }
}
```

```
user_reviews
```

```
##  [1] 5100  158  111 1000 3500  787 1000   53  966  205   80  245 5900  369  126
## [16]  468  910   12  542  214  175 1000 2300  219 1700
```

```
critic_reviews
```

```
##  [1] 175   6  10  34  88  77  57   9  93  12   8  15 368   4   5  16  94   9  28
## [20]  85  13 121  64  25  76
```

```
max_length <- max(length(rank), length(title), length(year), length(rating), length(episodes), length(vo
rank <- c(rank, rep(NA, max_length - length(rank)))
title <- c(title, rep(NA, max_length - length(title)))
year <- c(year, rep(NA, max_length - length(year)))
rating <- c(rating, rep(NA, max_length - length(rating)))
episodes <- c(episodes, rep(NA, max_length - length(episodes)))
vote <- c(vote, rep(NA, max_length - length(vote)))
user_reviews <- c(user_reviews, rep(NA, max_length - length(user_reviews)))
critic_reviews <- c(critic_reviews, rep(NA, max_length - length(critic_reviews)))
max_length
```

```
## [1] 25
```

```
movies = data.frame(rank, title, year, rating, episodes, vote, user_reviews, critic_reviews, stringsAsF
write.csv(movies, "movies.csv")
print(head(movies))
```

```
##   rank             title year rating episodes   vote user_reviews
## 1   1.      Breaking Bad   NA    9.5       NA  (2.2M)         5100
## 2   2.   Planet Earth II   NA    9.5       NA  (163K)          158
## 3   3.      Planet Earth   NA    9.4       NA  (224K)          111
## 4   4.  Band of Brothers   NA    9.4       NA  (547K)         1000
## 5   5.         Chernobyl   NA    9.3       NA  (910K)         3500
## 6   6.          The Wire   NA    9.3       NA  (392K)          787
```

rank	title	year	rating	episodes	vote	user_reviews	critic_reviews
1.	Breaking Bad	NA	9.5	NA	(2.2M)	5100	175
2.	Planet Earth II	NA	9.5	NA	(163K)	158	6
3.	Planet Earth	NA	9.4	NA	(224K)	111	10
4.	Band of Brothers	NA	9.4	NA	(547K)	1000	34
5.	Chernobyl	NA	9.3	NA	(910K)	3500	88
6.	The Wire	NA	9.3	NA	(392K)	787	77
7.	Avatar: The Last Airbender	NA	9.3	NA	(391K)	1000	57
8.	Blue Planet II	NA	9.3	NA	(49K)	53	9
9.	The Sopranos	NA	9.2	NA	(500K)	966	93
10.	Cosmos: A Spacetime Odyssey	NA	9.2	NA	(132K)	205	12
11.	Cosmos	NA	9.3	NA	(46K)	80	8
12.	Our Planet	NA	9.2	NA	(54K)	245	15
13.	Game of Thrones	NA	9.2	NA	(2.4M)	5900	368
14.	Bluey	NA	9.3	NA	(34K)	369	4
15.	The World at War	NA	9.2	NA	(31K)	126	5
16.	Fullmetal Alchemist: Brotherhood	NA	9.1	NA	(210K)	468	16
17.	Rick and Morty	NA	9.1	NA	(629K)	910	94
18.	Life	NA	9.1	NA	(44K)	12	9
19.	The Last Dance	NA	9.0	NA	(160K)	542	28
20.	The Twilight Zone	NA	9.0	NA	(97K)	214	85
21.	The Vietnam War	NA	9.1	NA	(30K)	175	13
22.	Sherlock	NA	9.1	NA	(1M)	1000	121
23.	Attack on Titan	NA	9.1	NA	(565K)	2300	64
24.	Batman: The Animated Series	NA	9.0	NA	(123K)	219	25
25.	Arcane	NA	9.0	NA	(332K)	1700	76

```
##   critic_reviews
## 1            175
## 2              6
## 3             10
## 4             34
## 5             88
## 6             77
```

```
movies %>%
  kable("latex", booktabs = TRUE) %>%
  kable_styling(latex_options = "scale_down")
```

```
link2 = "https://www.imdb.com/title/tt0903747/reviews/?ref_=tt_ov_ql_2"
page2 = read_html(link)
session2 <- bow(link, user_agent = "Educational")
        session2
```

```
## <polite session> https://www.imdb.com/chart/toptv/
##     User-agent: Educational
##     robots.txt: 35 rules are defined for 3 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
reviews <- page2 %>% html_nodes(".ipc-link--base") %>%
  html_text()
reviews
```

```
## [1] "Learn more about how list ranking is determined."
```

```
date <- page2 %>% html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text()
date
```

```
## character(0)
```

```
user_rating <- page2 %>% html_nodes(".sc-a2ac93e5-4.gyib0i") %>%
  html_text()
user_rating
```

```
## character(0)
```

```
link1 = "https://www.imdb.com/chart/toptv/"
page1 = read_html(link)
session1 <- bow(link1, user_agent = "Educational")
        session1
```

```
## <polite session> https://www.imdb.com/chart/toptv/
##     User-agent: Educational
##     robots.txt: 35 rules are defined for 3 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
user_review = page %>% html_nodes(".score") %>% html_text()
user_review
```

```
## character(0)
```

```
library(ggplot2)

movies$year <- as.numeric(movies$year)
year_counts <- movies %>%
  filter(!is.na(year)) %>%
  count(year)

ggplot(year_counts, aes(x = year, y = n)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  labs(title = "Number of TV Shows Released by Year",
       x = "Year",
       y = "Number of TV Shows") +
  theme_minimal()
```

```
most_releases <- year_counts[which.max(year_counts$n), ]
print(most_releases)
```

```
## [1] year n
## <0 rows> (or 0-length row.names)
```