RWorksheet#5_group.Rmd

Calzado_Calvario_Baylon

2024-11-01

```
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
```

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

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

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

```
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
```

```
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
```

```
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. The Office"
```

```
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."
```

```
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] " The Office"
```

```
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
```

```
##  [1] "2008-2013" "2016"      "2006"      "2001"      "2019"      "2002-2008"
##  [7] "2005-2008" "2017"      "1999-2007" "2014"      "1980"      "2019-2023"
## [13] "2011-2019" "2018"      "1973-1974" "2009-2010" "2013"      "2009"
## [19] "2020"      "1959-1964" "2017"      "2010-2017" "2013-2023" "1992-1995"
## [25] "2005-2013"
```

```
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" "9.1" "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
```

```
##  [1] "62 eps"  "6 eps"   "11 eps"  "10 eps"  "5 eps"   "60 eps"  "62 eps"
##  [8] "7 eps"   "86 eps"  "13 eps"  "13 eps"  "12 eps"  "74 eps"  "194 eps"
## [15] "26 eps"  "68 eps"  "78 eps"  "11 eps"  "10 eps"  "156 eps" "10 eps"
## [22] "15 eps"  "98 eps"  "85 eps"  "188 eps"
```

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

```
##  [1] " (2.2M)" " (162K)" " (223K)" " (544K)" " (905K)" " (390K)" " (388K)"
##  [8] " (48K)"  " (496K)" " (131K)" " (45K)"  " (53K)"  " (2.4M)" " (33K)"
## [15] " (31K)"  " (208K)" " (625K)" " (43K)"  " (159K)" " (96K)"  " (29K)"
## [22] " (1M)"   " (558K)" " (122K)" " (744K)"
```

```
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))
popularity <- vector("numeric", length(urls))
```

```
for (i in seq_along(urls)) {

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

  webpage <- scrape(session)

  popularity_text <- webpage %>% html_nodes(".sc-39d285cf-1.dxqvqi") %>% html_text()
  popularity[i] <- as.numeric(gsub(",", "", popularity_text[1]))

  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] 5000  158  111 1000 3500  786  997   53  960  205   80  245 5800  366  126
## [16]  463  908   12  541  213  175 1000 2300  218 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
```

```
popularity
```

```
##  [1]   18 1144 2090  172  176  106  384 4625   37 1483 3922 2955   12  391 2383
## [16]  504  146 3388 1472  315 1878  178   61  498   56
```

```
max_length <- max(length(rank), length(title), length(year), length(rating), length(episodes), length(v
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)))
```

rank	title	year	rating	episodes	vote	user_reviews	critic_reviews	popularity
1.	Breaking Bad	2008–2013	9.5	62 eps	(2.2M)	5000	175	18
2.	Planet Earth II	2016	9.5	6 eps	(162K)	158	6	1144
3.	Planet Earth	2006	9.4	11 eps	(223K)	111	10	2090
4.	Band of Brothers	2001	9.4	10 eps	(544K)	1000	34	172
5.	Chernobyl	2019	9.3	5 eps	(905K)	3500	88	176
6.	The Wire	2002–2008	9.3	60 eps	(390K)	786	77	106
7.	Avatar: The Last Airbender	2005–2008	9.3	62 eps	(388K)	997	57	384
8.	Blue Planet II	2017	9.3	7 eps	(48K)	53	9	4625
9.	The Sopranos	1999–2007	9.2	86 eps	(496K)	960	93	37
10.	Cosmos: A Spacetime Odyssey	2014	9.2	13 eps	(131K)	205	12	1483
11.	Cosmos	1980	9.3	13 eps	(45K)	80	8	3922
12.	Our Planet	2019–2023	9.2	12 eps	(53K)	245	15	2955
13.	Game of Thrones	2011–2019	9.2	74 eps	(2.4M)	5800	368	12
14.	Bluey	2018	9.3	194 eps	(33K)	366	4	391
15.	The World at War	1973–1974	9.2	26 eps	(31K)	126	5	2383
16.	Fullmetal Alchemist: Brotherhood	2009–2010	9.1	68 eps	(208K)	463	16	504
17.	Rick and Morty	2013	9.1	78 eps	(625K)	908	94	146
18.	Life	2009	9.1	11 eps	(43K)	12	9	3388
19.	The Last Dance	2020	9.1	10 eps	(159K)	541	28	1472
20.	The Twilight Zone	1959–1964	9.0	156 eps	(96K)	213	85	315
21.	The Vietnam War	2017	9.1	10 eps	(29K)	175	13	1878
22.	Sherlock	2010–2017	9.1	15 eps	(1M)	1000	121	178
23.	Attack on Titan	2013–2023	9.1	98 eps	(558K)	2300	64	61
24.	Batman: The Animated Series	1992–1995	9.0	85 eps	(122K)	218	25	498
25.	The Office	2005–2013	9.0	188 eps	(744K)	1700	76	56

```
user_reviews <- c(user_reviews, rep(NA, max_length - length(user_reviews)))
critic_reviews <- c(critic_reviews, rep(NA, max_length - length(critic_reviews)))
popularity <- c(popularity, rep(NA, max_length - length(popularity)))
max_length
```

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

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

```
##   rank             title      year rating episodes    vote user_reviews
## 1   1.      Breaking Bad 2008-2013    9.5   62 eps  (2.2M)          5000
## 2   2.   Planet Earth II      2016    9.5    6 eps  (162K)           158
## 3   3.      Planet Earth      2006    9.4   11 eps  (223K)           111
## 4   4.  Band of Brothers      2001    9.4   10 eps  (544K)          1000
## 5   5.         Chernobyl      2019    9.3    5 eps  (905K)          3500
## 6   6.          The Wire 2002-2008    9.3   60 eps  (390K)           786
##   critic_reviews popularity
## 1            175         18
## 2              6       1144
## 3             10       2090
## 4             34        172
## 5             88        176
## 6             77        106
```

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

```
rl <- c("https://www.imdb.com/title/tt0903747/reviews/?ref_=tt_ov_ql_2",
        "https://www.imdb.com/title/tt5491994/reviews/?ref_=tt_ov_ql_2",
        "https://www.imdb.com/title/tt0795176/reviews/?ref_=tt_ov_ql_2",
        "https://www.imdb.com/title/tt0185906/reviews/?ref_=tt_ov_ql_2",
        "https://www.imdb.com/title/tt7366338/reviews/?ref_=tt_ov_ql_2")

name <- vector("list", length(rl))
date_of_review <- vector("list", length(rl))
user_ratings <- vector("list", length(rl))
title_of_review <- vector("list", length(rl))
is_helpful <- vector("list", length(rl))
is_not_helpful <- vector("list", length(rl))
text_review <- vector("list", length(rl))
```

```
for (i in seq_along(rl)) {
  session <- bow(rl[i], user_agent = "Educational")
  webpage <- scrape(session)

  namez <- webpage %>% html_nodes(".ipc-link.ipc-link--base") %>% html_text(trim = TRUE) %>% head(40)
  name[[i]] <- namez[!grepl("Permalink", namez, ignore.case = TRUE)]

  date_of_review[[i]] <- webpage %>% html_nodes(".ipc-inline-list__item.review-date") %>% html_text(trim

  user_ratings[[i]] <- webpage %>% html_nodes(".ipc-rating-star--rating") %>% html_text(trim = TRUE) %>%

  title_of <- webpage %>% html_nodes(".ipc-title__text") %>% html_text(trim = TRUE) %>% head(21)
  title_of_review[[i]] <- title_of[!grepl("User reviews|More from this title|More to explore|Recently vi

  text_review[[i]] <- webpage %>% html_nodes(".ipc-html-content-inner-div") %>% html_text(trim = TRUE) %
}
```

```
reviews_data <- data.frame(
  Name = unlist(name),
  Date = unlist(date_of_review),
  Rating = unlist(user_ratings),
  Title = unlist(title_of_review),
  Review_Text = unlist(text_review),
  stringsAsFactors = FALSE
)
```

```
write.csv(reviews_data, "user_reviews.csv")
print(head(reviews_data))
```

```
##                Name         Date Rating                            Title
## 1           FiRE010  Jul 3, 2021     10                     Really Great
## 2        bruhperson  Mar 6, 2019     10                  It's ok I guess
## 3      KinoKoopaKid Jul 29, 2021     10                       99.1% pure
## 4       jehuschultz Feb 18, 2020     10                         The Best
## 5    Supermanfan-13  Nov 8, 2021     10                Damn near perfect!
## 6 manishsingh-03299 May 30, 2019     10 Those days ain't gonna come back..
##
## 1 I have never watched a show that is as consistently genuine and engaging as Breaking Bad. This is u
## 2
## 3
## 4
```

```
## 5
## 6
```

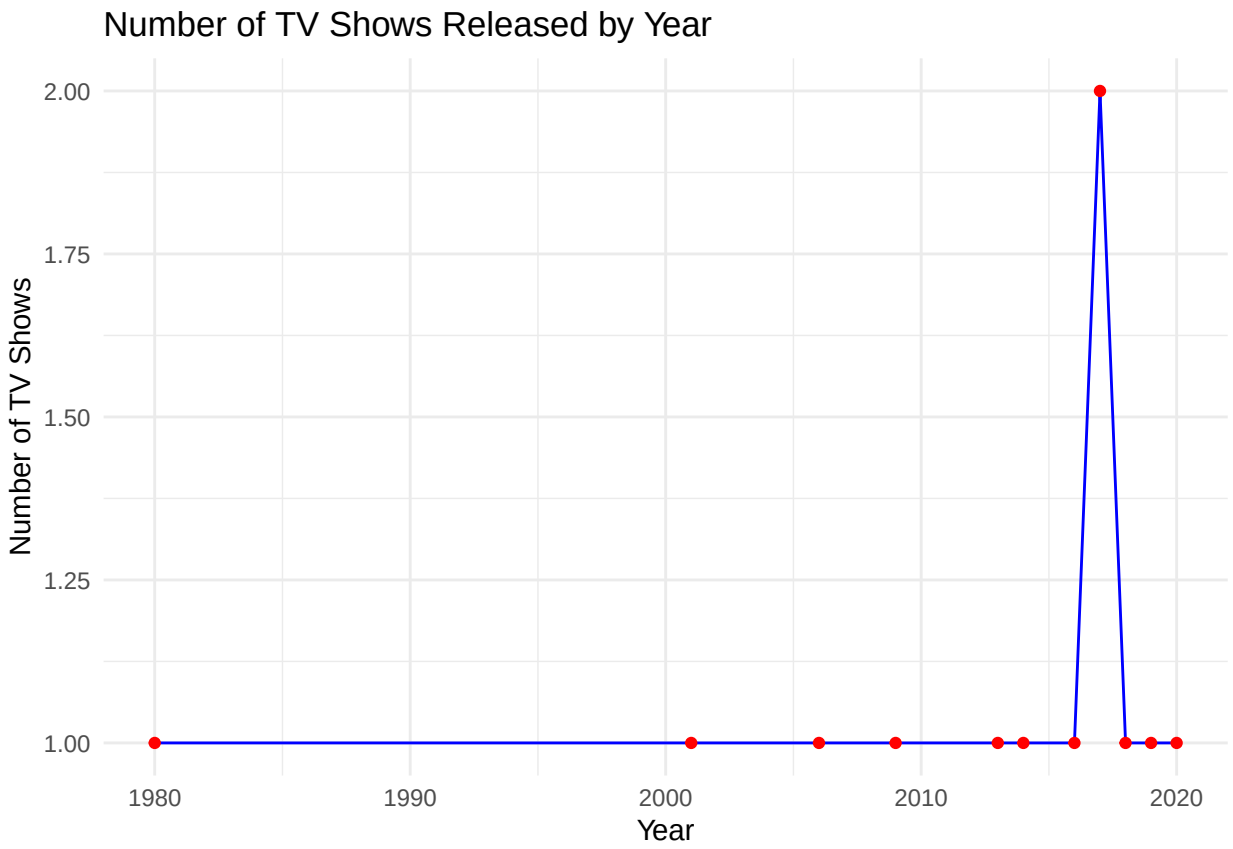
```
library(ggplot2)

movies$year <- as.numeric(movies$year)
```

```
## Warning: NAs introduced by coercion
```

```
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)
```

```
##   year n
## 8 2017 2
```