

RWorksheet#5_group(Leysa,Calambro)

Camilo Leysa

2024-11-11

Loading needed libraries:

```
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 4.4.2
```

```
library(polite)
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(httr)
```

```
## Warning: package 'httr' was built under R version 4.4.2
```

```
library(stringr)
library(ggplot2)
```

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

```
## v Setting active project to "C:/Worksheet_#5/Worksheet_#5".
```

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

```
## <polite session> https://www.imdb.com/chart/toptv/?ref_=nv_tv_250
## User-agent: Educational
## robots.txt: 35 rules are defined for 3 bots
## Crawl delay: 5 sec
## The path is scrapable for this user-agent
```

1.

Getting the TV Show title.

```
#Title
title_lists <- scrape(session) %>% html_nodes("h3.ipc-title__text") %>% html_text(trim = TRUE)
#filter unwanted
title_lists <- title_lists[!grepl("Recently viewed", title_lists)]
title_lists
```

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

List of top 50 TV Shows

```
class(title_lists)
```

```
## [1] "character"
```

```
title_List <- as.data.frame(title_lists[2:51])
title_List
```

```
## title_lists[2:51]
## 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
## 26          <NA>
## 27          <NA>
## 28          <NA>
## 29          <NA>
## 30          <NA>
## 31          <NA>
## 32          <NA>
## 33          <NA>
## 34          <NA>
## 35          <NA>
## 36          <NA>
## 37          <NA>
## 38          <NA>
## 39          <NA>
## 40          <NA>
## 41          <NA>
## 42          <NA>
## 43          <NA>
## 44          <NA>
## 45          <NA>
## 46          <NA>
## 47          <NA>
## 48          <NA>
## 49          <NA>
## 50          <NA>
```

Seperating the rank number and the TV Show title.

```
colnames(title_List) <- "ranks"
split_df <- strsplit(as.character(title_List$ranks), ".", fixed = TRUE)
split_df <- data.frame(do.call(rbind, split_df))
split_df <- split_df[-c(3:4)]
colnames(split_df) <- c("Ranks", "Title")
str(split_df)
```

```
## 'data.frame':   50 obs. of  2 variables:
## $ Ranks: chr  "1" "2" "3" "4" ...
## $ Title: chr  " Breaking Bad" " Planet Earth II" " Planet Earth" " Band of Brothers" ...
```

The Rank and the Title of the TV Shows

```
class(split_df)
```

```
## [1] "data.frame"
```

```
split_df
```

##	Ranks	Title
## 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
## 26	<NA>	<NA>
## 27	<NA>	<NA>
## 28	<NA>	<NA>
## 29	<NA>	<NA>
## 30	<NA>	<NA>
## 31	<NA>	<NA>
## 32	<NA>	<NA>
## 33	<NA>	<NA>
## 34	<NA>	<NA>
## 35	<NA>	<NA>
## 36	<NA>	<NA>
## 37	<NA>	<NA>
## 38	<NA>	<NA>
## 39	<NA>	<NA>
## 40	<NA>	<NA>
## 41	<NA>	<NA>
## 42	<NA>	<NA>
## 43	<NA>	<NA>
## 44	<NA>	<NA>
## 45	<NA>	<NA>

```
## 46 <NA> <NA>
## 47 <NA> <NA>
## 48 <NA> <NA>
## 49 <NA> <NA>
## 50 <NA> <NA>
```

Top 50 TV Show Rating

```
rating <- scrape(session) %>% html_nodes("span.ipc-rating-star--rating") %>% html_text
tv_rating <- as.data.frame(rating [1:50])
tv_rating
```

```
##      rating[1:50]
## 1             9.5
## 2             9.5
## 3             9.4
## 4             9.4
## 5             9.3
## 6             9.3
## 7             9.3
## 8             9.3
## 9             9.2
## 10            9.2
## 11            9.3
## 12            9.2
## 13            9.2
## 14            9.3
## 15            9.2
## 16            9.1
## 17            9.1
## 18            9.1
## 19            9.0
## 20            9.0
## 21            9.1
## 22            9.1
## 23            9.1
## 24            9.0
## 25            9.0
## 26            <NA>
## 27            <NA>
## 28            <NA>
## 29            <NA>
## 30            <NA>
## 31            <NA>
## 32            <NA>
## 33            <NA>
## 34            <NA>
## 35            <NA>
## 36            <NA>
## 37            <NA>
## 38            <NA>
## 39            <NA>
## 40            <NA>
```

```
## 41      <NA>
## 42      <NA>
## 43      <NA>
## 44      <NA>
## 45      <NA>
## 46      <NA>
## 47      <NA>
## 48      <NA>
## 49      <NA>
## 50      <NA>
```

Number of People who Voted

```
tv_votes <- scrape(session) %>% html_nodes("span.ipc-rating-star--voteCount") %>% html_text
total_tv_votes <- as.data.frame(tv_votes[1:50])
total_tv_votes
```

```
##      tv_votes[1:50]
## 1      (2.2M)
## 2      (162K)
## 3      (224K)
## 4      (546K)
## 5      (908K)
## 6      (391K)
## 7      (390K)
## 8      (49K)
## 9      (499K)
## 10     (131K)
## 11     (46K)
## 12     (54K)
## 13     (2.4M)
## 14     (34K)
## 15     (31K)
## 16     (209K)
## 17     (628K)
## 18     (44K)
## 19     (160K)
## 20     (97K)
## 21     (29K)
## 22     (1M)
## 23     (563K)
## 24     (122K)
## 25     (317K)
## 26      <NA>
## 27      <NA>
## 28      <NA>
## 29      <NA>
## 30      <NA>
## 31      <NA>
## 32      <NA>
## 33      <NA>
## 34      <NA>
## 35      <NA>
```

```
## 36      <NA>
## 37      <NA>
## 38      <NA>
## 39      <NA>
## 40      <NA>
## 41      <NA>
## 42      <NA>
## 43      <NA>
## 44      <NA>
## 45      <NA>
## 46      <NA>
## 47      <NA>
## 48      <NA>
## 49      <NA>
## 50      <NA>
```

Number of Episodes of each TV Shows

```
episodes <- scrape(session) %>% html_nodes("span.sc-6ade9358-7.exckou.cli-title-metadata-item") %>% html_text()
cl_episodes <- gsub("\\D", "", episodes)
cleaned_ep <- str_extract(episodes, "\\d+(?=\\s*eps)")
cleaned_ep <- as.numeric(cleaned_ep)
cleaned_ep <- cleaned_ep[!is.na(cleaned_ep)]
cleaned_episodes <- as.data.frame(cleaned_ep[1:25])
cleaned_episodes
```

```
##      cleaned_ep[1:25]
## 1                62
## 2                 6
## 3                11
## 4                10
## 5                 5
## 6                60
## 7                62
## 8                 7
## 9                86
## 10               13
## 11               13
## 12               12
## 13               74
## 14              194
## 15               26
## 16               68
## 17               78
## 18               11
## 19               10
## 20              156
## 21               10
## 22               15
## 23               98
## 24               85
## 25               18
```

Year of TV Shows released

```

tv_years <- scrape(session) %>% html_nodes("span.sc-6ade9358-7.exckou.cli-title-metadata-item") %>% html
clyear <- gsub(".*?(\\d{4})(-\\d{4})?.*", "\\1", tv_years)
yeartv <- str_extract(tv_years, "\\b\\d{4}(-\\d{4})?\\b")
yeartv <- as.numeric(yeartv)
yeartv <- yeartv[!is.na(yeartv)]
tv_year_of_air <- as.data.frame(yeartv[1:50])
tv_year_of_air

```

```

##      yeartv[1:50]
## 1          2008
## 2          2016
## 3          2006
## 4          2001
## 5          2019
## 6          2002
## 7          2005
## 8          2017
## 9          1999
## 10         2014
## 11         1980
## 12         2019
## 13         2011
## 14         2018
## 15         1973
## 16         2009
## 17         2013
## 18         2009
## 19         2020
## 20         1959
## 21         2017
## 22         2010
## 23         2013
## 24         1992
## 25         2021
## 26          NA
## 27          NA
## 28          NA
## 29          NA
## 30          NA
## 31          NA
## 32          NA
## 33          NA
## 34          NA
## 35          NA
## 36          NA
## 37          NA
## 38          NA
## 39          NA
## 40          NA
## 41          NA
## 42          NA
## 43          NA
## 44          NA

```



```
## 45      NA
## 46      NA
## 47      NA
## 48      NA
## 49      NA
## 50      NA
```

Number of User Reviews

```
review <- scrape(session) %>% html_nodes("ul.ipc-inline-list.sc-b782214c-0.b1lRjU.baseAlt") %>% html_text()
reviews <- as.data.frame(review[1:50])
reviews
```

```
##      review[1:50]
## 1      <NA>
## 2      <NA>
## 3      <NA>
## 4      <NA>
## 5      <NA>
## 6      <NA>
## 7      <NA>
## 8      <NA>
## 9      <NA>
## 10     <NA>
## 11     <NA>
## 12     <NA>
## 13     <NA>
## 14     <NA>
## 15     <NA>
## 16     <NA>
## 17     <NA>
## 18     <NA>
## 19     <NA>
## 20     <NA>
## 21     <NA>
## 22     <NA>
## 23     <NA>
## 24     <NA>
## 25     <NA>
## 26     <NA>
## 27     <NA>
## 28     <NA>
## 29     <NA>
## 30     <NA>
## 31     <NA>
## 32     <NA>
## 33     <NA>
## 34     <NA>
## 35     <NA>
## 36     <NA>
## 37     <NA>
## 38     <NA>
## 39     <NA>
## 40     <NA>
```

```
## 41      <NA>
## 42      <NA>
## 43      <NA>
## 44      <NA>
## 45      <NA>
## 46      <NA>
## 47      <NA>
## 48      <NA>
## 49      <NA>
## 50      <NA>
```

```
#tried several links but still getting NA's.
```

Getting the number of Critic Reviews

```
critic <- scrape(session) %>% html_nodes("ul.ipc-inline-list.sc-b782214c-0.b1lRjU.baseAlt") %>% html_text()
critic_rev <- as.data.frame(critic [1:50])
critic_rev
```

```
##      critic[1:50]
## 1      <NA>
## 2      <NA>
## 3      <NA>
## 4      <NA>
## 5      <NA>
## 6      <NA>
## 7      <NA>
## 8      <NA>
## 9      <NA>
## 10     <NA>
## 11     <NA>
## 12     <NA>
## 13     <NA>
## 14     <NA>
## 15     <NA>
## 16     <NA>
## 17     <NA>
## 18     <NA>
## 19     <NA>
## 20     <NA>
## 21     <NA>
## 22     <NA>
## 23     <NA>
## 24     <NA>
## 25     <NA>
## 26     <NA>
## 27     <NA>
## 28     <NA>
## 29     <NA>
## 30     <NA>
## 31     <NA>
## 32     <NA>
## 33     <NA>
```

```
## 34      <NA>
## 35      <NA>
## 36      <NA>
## 37      <NA>
## 38      <NA>
## 39      <NA>
## 40      <NA>
## 41      <NA>
## 42      <NA>
## 43      <NA>
## 44      <NA>
## 45      <NA>
## 46      <NA>
## 47      <NA>
## 48      <NA>
## 49      <NA>
## 50      <NA>
```

```
#tried several links but still getting NA's.
```

Getting the popularity Review

```
popularity_review <- scrape(session) %>% html_nodes("div.hero-rating-bar__popularity__down") %>% html_text()
pop <- as.data.frame(popularity_review [1:50])
pop
```

```
##      popularity_review[1:50]
## 1      <NA>
## 2      <NA>
## 3      <NA>
## 4      <NA>
## 5      <NA>
## 6      <NA>
## 7      <NA>
## 8      <NA>
## 9      <NA>
## 10     <NA>
## 11     <NA>
## 12     <NA>
## 13     <NA>
## 14     <NA>
## 15     <NA>
## 16     <NA>
## 17     <NA>
## 18     <NA>
## 19     <NA>
## 20     <NA>
## 21     <NA>
## 22     <NA>
## 23     <NA>
## 24     <NA>
## 25     <NA>
## 26     <NA>
```

```
## 27          <NA>
## 28          <NA>
## 29          <NA>
## 30          <NA>
## 31          <NA>
## 32          <NA>
## 33          <NA>
## 34          <NA>
## 35          <NA>
## 36          <NA>
## 37          <NA>
## 38          <NA>
## 39          <NA>
## 40          <NA>
## 41          <NA>
## 42          <NA>
## 43          <NA>
## 44          <NA>
## 45          <NA>
## 46          <NA>
## 47          <NA>
## 48          <NA>
## 49          <NA>
## 50          <NA>
```

```
#tried several links but still getting NA's.
```

Data frame of TV Shows

```
final_data <- cbind(split_df,tv_rating,total_tv_votes,cleaned_episodes,tv_year_of_air,reviews,critic_re
colnames(final_data) <- c("Ranks", "TV Show Title", "Rating", "Number of People Voted", "Number of Epis
final_data
```

##	Ranks	TV Show Title	Rating	Number of People Voted
## 1	1	Breaking Bad	9.5	(2.2M)
## 2	2	Planet Earth II	9.5	(162K)
## 3	3	Planet Earth	9.4	(224K)
## 4	4	Band of Brothers	9.4	(546K)
## 5	5	Chernobyl	9.3	(908K)
## 6	6	The Wire	9.3	(391K)
## 7	7	Avatar: The Last Airbender	9.3	(390K)
## 8	8	Blue Planet II	9.3	(49K)
## 9	9	The Sopranos	9.2	(499K)
## 10	10	Cosmos: A Spacetime Odyssey	9.2	(131K)
## 11	11	Cosmos	9.3	(46K)
## 12	12	Our Planet	9.2	(54K)
## 13	13	Game of Thrones	9.2	(2.4M)
## 14	14	Bluey	9.3	(34K)
## 15	15	The World at War	9.2	(31K)
## 16	16	Fullmetal Alchemist Brotherhood	9.1	(209K)
## 17	17	Rick and Morty	9.1	(628K)
## 18	18	Life	9.1	(44K)
## 19	19	The Last Dance	9.0	(160K)

##	20	20	The Twilight Zone	9.0	(97K)
##	21	21	The Vietnam War	9.1	(29K)
##	22	22	Sherlock	9.1	(1M)
##	23	23	Attack on Titan	9.1	(563K)
##	24	24	Batman: The Animated Series	9.0	(122K)
##	25	25	Arcane	9.0	(317K)
##	26	<NA>	<NA>	<NA>	<NA>
##	27	<NA>	<NA>	<NA>	<NA>
##	28	<NA>	<NA>	<NA>	<NA>
##	29	<NA>	<NA>	<NA>	<NA>
##	30	<NA>	<NA>	<NA>	<NA>
##	31	<NA>	<NA>	<NA>	<NA>
##	32	<NA>	<NA>	<NA>	<NA>
##	33	<NA>	<NA>	<NA>	<NA>
##	34	<NA>	<NA>	<NA>	<NA>
##	35	<NA>	<NA>	<NA>	<NA>
##	36	<NA>	<NA>	<NA>	<NA>
##	37	<NA>	<NA>	<NA>	<NA>
##	38	<NA>	<NA>	<NA>	<NA>
##	39	<NA>	<NA>	<NA>	<NA>
##	40	<NA>	<NA>	<NA>	<NA>
##	41	<NA>	<NA>	<NA>	<NA>
##	42	<NA>	<NA>	<NA>	<NA>
##	43	<NA>	<NA>	<NA>	<NA>
##	44	<NA>	<NA>	<NA>	<NA>
##	45	<NA>	<NA>	<NA>	<NA>
##	46	<NA>	<NA>	<NA>	<NA>
##	47	<NA>	<NA>	<NA>	<NA>
##	48	<NA>	<NA>	<NA>	<NA>
##	49	<NA>	<NA>	<NA>	<NA>
##	50	<NA>	<NA>	<NA>	<NA>

##	Number of Episodes	Year Released	User Reviews	Critic Reviews
##	1	62	2008	<NA>
##	2	6	2016	<NA>
##	3	11	2006	<NA>
##	4	10	2001	<NA>
##	5	5	2019	<NA>
##	6	60	2002	<NA>
##	7	62	2005	<NA>
##	8	7	2017	<NA>
##	9	86	1999	<NA>
##	10	13	2014	<NA>
##	11	13	1980	<NA>
##	12	12	2019	<NA>
##	13	74	2011	<NA>
##	14	194	2018	<NA>
##	15	26	1973	<NA>
##	16	68	2009	<NA>
##	17	78	2013	<NA>
##	18	11	2009	<NA>
##	19	10	2020	<NA>
##	20	156	1959	<NA>
##	21	10	2017	<NA>
##	22	15	2010	<NA>

## 23	98	2013	<NA>	<NA>
## 24	85	1992	<NA>	<NA>
## 25	18	2021	<NA>	<NA>
## 26	62	NA	<NA>	<NA>
## 27	6	NA	<NA>	<NA>
## 28	11	NA	<NA>	<NA>
## 29	10	NA	<NA>	<NA>
## 30	5	NA	<NA>	<NA>
## 31	60	NA	<NA>	<NA>
## 32	62	NA	<NA>	<NA>
## 33	7	NA	<NA>	<NA>
## 34	86	NA	<NA>	<NA>
## 35	13	NA	<NA>	<NA>
## 36	13	NA	<NA>	<NA>
## 37	12	NA	<NA>	<NA>
## 38	74	NA	<NA>	<NA>
## 39	194	NA	<NA>	<NA>
## 40	26	NA	<NA>	<NA>
## 41	68	NA	<NA>	<NA>
## 42	78	NA	<NA>	<NA>
## 43	11	NA	<NA>	<NA>
## 44	10	NA	<NA>	<NA>
## 45	156	NA	<NA>	<NA>
## 46	10	NA	<NA>	<NA>
## 47	15	NA	<NA>	<NA>
## 48	98	NA	<NA>	<NA>
## 49	85	NA	<NA>	<NA>
## 50	18	NA	<NA>	<NA>
##	Popularity Review			
## 1	<NA>			
## 2	<NA>			
## 3	<NA>			
## 4	<NA>			
## 5	<NA>			
## 6	<NA>			
## 7	<NA>			
## 8	<NA>			
## 9	<NA>			
## 10	<NA>			
## 11	<NA>			
## 12	<NA>			
## 13	<NA>			
## 14	<NA>			
## 15	<NA>			
## 16	<NA>			
## 17	<NA>			
## 18	<NA>			
## 19	<NA>			
## 20	<NA>			
## 21	<NA>			
## 22	<NA>			
## 23	<NA>			
## 24	<NA>			
## 25	<NA>			

```
## 26          <NA>
## 27          <NA>
## 28          <NA>
## 29          <NA>
## 30          <NA>
## 31          <NA>
## 32          <NA>
## 33          <NA>
## 34          <NA>
## 35          <NA>
## 36          <NA>
## 37          <NA>
## 38          <NA>
## 39          <NA>
## 40          <NA>
## 41          <NA>
## 42          <NA>
## 43          <NA>
## 44          <NA>
## 45          <NA>
## 46          <NA>
## 47          <NA>
## 48          <NA>
## 49          <NA>
## 50          <NA>
```

```
write.csv(final_data, "Top_50_tvshows.csv", row.names = FALSE)
```

2. 5 TV Shows chosen for user reviews.

```
top_5 <- scrape(session) %>% html_nodes("h3.ipc-title__text") %>% html_text(trim = TRUE)
tv_top5 <- as.data.frame(list(Rank_Title = top_5[2:6]), stringsAsFactors = FALSE)
split_list <- strsplit(as.character(tv_top5$Rank_Title), "\\.", fixed = FALSE)
split_Df <- data.frame(do.call(rbind, lapply(split_list, function(x) {
  if(length(x) == 2) {
    return(x)
  } else {
    return(c(x[1], NA))
  }
})), stringsAsFactors = FALSE)[-c(3:4)]
colnames(split_Df) <- c("Ranks", "Title")
str(split_Df)
```

```
## 'data.frame':    5 obs. of  2 variables:
## $ Ranks: chr  "1" "2" "3" "4" ...
## $ Title: chr  " Breaking Bad" " Planet Earth II" " Planet Earth" " Band of Brothers" ...
```

```
split_Df
```

```
##   Ranks      Title
## 1     1  Breaking Bad
## 2     2 Planet Earth II
```

```
## 3      3      Planet Earth
## 4      4      Band of Brothers
## 5      5      Chernobyl
```

Getting the reviewer's name

```
urev <- scrape(session) %>% html_nodes("ul.ipc-inline-list.ipc-inline-list--show-dividers.ipc-inline-li")
userev <- as.data.frame(urev)
userev
```

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

Date of Review of the Person

```
revdate <- scrape(session) %>% html_nodes("li.ipc-inline-list_item review-date") %>% html_text
datereview <- as.data.frame(revdate)
datereview
```

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

User rating

```
user_rate <- scrape(session) %>% html_nodes("span.ipc-rating-star--rating") %>% html_text
tv_rating <- as.data.frame(rating [1:20])
tv_rating
```

```
##      rating[1:20]
## 1              9.5
## 2              9.5
## 3              9.4
## 4              9.4
## 5              9.3
## 6              9.3
## 7              9.3
## 8              9.3
## 9              9.2
## 10             9.2
## 11             9.3
## 12             9.2
## 13             9.2
## 14             9.3
## 15             9.2
## 16             9.1
## 17             9.1
## 18             9.1
## 19             9.0
## 20             9.0
```

3. Graph of each TV show releases each year


```
str(final_data$`Year Released`)
```

```
## num [1:50] 2008 2016 2006 2001 2019 ...
```

```
final_data$`Year Released` <- as.numeric(as.character(final_data$`Year Released`))
fdata <- final_data %>% filter(!is.na(`Year Released`))
tv_shows_by_year <- final_data %>%
  group_by(`Year Released`) %>%
  summarise(num_shows = n()) %>%
  arrange(`Year Released`)

# Create a time series plot
ggplot(tv_shows_by_year, aes(x = `Year Released`, y = num_shows)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "Number of TV Shows Released by Year",
       x = "Year",
       y = "Number of Shows") +
  theme_minimal()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## ('geom_line()').
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## ('geom_point()').
```

