assighmentE

YunBo Zhang

2025-09-30

```
library(readr)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ purrr     1.1.0
## ✓ forcats   1.0.1     ✓ stringr   1.5.2
## ✓ ggplot2   4.0.0     ✓ tibble    3.3.0
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ── Conflicts ────────────────────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidyr)
data_clean <- read_csv("C:/Users/Cikey/Desktop/data/WDBD.csv", n_max = 2977)#remove the introduction in the bottle
```

```
## Rows: 2977 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (11): Country Name, Country Code, Series Name, Series Code, 2018 [YR2018...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data_long <- data_clean %>%
  pivot_longer(
    cols = `2018 [YR2018]`:`2024 [YR2024]`,
    names_to = "Year",
    values_to = "Value"
  )
data_long <- data_long %>%
  mutate(
    Year = gsub(" \\[YR[0-9]+\\]", "", Year)
  )#Use pivot_longer() to gather year columns into a single Year column and change the value into real year value.
data_wide <- data_long %>%
  pivot_wider(
    names_from = `Series Name`,
    values_from = Value
  )
data_wide#Use pivot_widerrestructure Series Names to see the effect.
```

```
## # A tibble: 20,839 × 53
##    `Country Name` `Country Code` `Series Code`     Year  Access to clean fuels…¹
##    <chr>          <chr>          <chr>             <chr> <chr>
##  1 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2018  14.5
##  2 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2019  15.6
##  3 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2020  16.4
##  4 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2021  17.4
##  5 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2022  18.5
##  6 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2023  ..
##  7 Afghanistan    AFG            EG.CFT.ACCS.RU.ZS 2024  ..
##  8 Afghanistan    AFG            EG.CFT.ACCS.ZS    2018  <NA>
##  9 Afghanistan    AFG            EG.CFT.ACCS.ZS    2019  <NA>
## 10 Afghanistan    AFG            EG.CFT.ACCS.ZS    2020  <NA>
## # ℹ 20,829 more rows
## # ℹ abbreviated name:
## #   ¹`Access to clean fuels and technologies for cooking, rural (% of rural population)`
## # ℹ 48 more variables:
## #   `Access to clean fuels and technologies for cooking (% of population)` <chr>,
## #   `Access to clean fuels and technologies for cooking, urban (% of urban population)` <chr>,
## #   `Access to electricity (% of population)` <chr>, …
```

#At the bottom of the dataset, there were two metadata rows: one for indicator names and one for units of measurement.We extracted these rows, reshaped them into long format, and merged them back into the main dataset.e did not standardize units, but we kept them in a separate Unit column for clarity.

```
library(tidyverse)
movies <- read.csv("C:/Users/Cikey/Desktop/data/movies.csv", stringsAsFactors = FALSE)
head(movies)
```

```
##   movieId                                      title             genres
## 1  182337                          Cinétracts (1968) (no genres listed)
## 2  195495                             Familia (2005)              Drama
## 3    3078                       Liberty Heights (1999)              Drama
## 4  134704 Comedy Central Roast of Charlie Sheen (2011)             Comedy
## 5  219976                      47 Hours to Live (2019)    Horror|Thriller
## 6  205715                                  Reis (2017) (no genres listed)
```

```
movies_long <- movies %>% #separate the mixed information in genre and years in title
  separate_rows(genres, sep = "\\|")
head(movies_long)
```

```
## # A tibble: 6 × 3
##   movieId title                                      genres
##     <int> <chr>                                      <chr>
## 1  182337 Cinétracts (1968)                          (no genres listed)
## 2  195495 Familia (2005)                             Drama
## 3    3078 Liberty Heights (1999)                     Drama
## 4  134704 Comedy Central Roast of Charlie Sheen (2011) Comedy
## 5  219976 47 Hours to Live (2019)                    Horror
## 6  219976 47 Hours to Live (2019)                    Thriller
```

```
movies_clean <- movies_long %>%
  separate(title, into = c("title_name", "year"), sep = " \\(", remove = TRUE) %>%
  mutate(year = str_replace(year, "\\)", ""))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 411 rows [16, 24, 25, 26, 41,
## 82, 83, 84, 85, 161, 162, 212, 214, 261, 264, 269, 270, 277, 380, 385, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 33 rows [349, 736, 737,
## 754, 1042, 1237, 1387, 1388, 1389, 1905, 1906, 2487, 2488, 2489, 2504, 2505,
## 2860, 3135, 3403, 3525, ...].
```

```
head(movies_clean)
```

```
## # A tibble: 6 × 4
##   movieId title_name                          year  genres
##     <int> <chr>                               <chr> <chr>
## 1  182337 Cinétracts                          1968  (no genres listed)
## 2  195495 Familia                             2005  Drama
## 3    3078 Liberty Heights                     1999  Drama
## 4  134704 Comedy Central Roast of Charlie Sheen 2011  Comedy
## 5  219976 47 Hours to Live                    2019  Horror
## 6  219976 47 Hours to Live                    2019  Thriller
```

```
movies_clean <- movies_clean %>%# some movie lose the information of genre,create a type call"unknown"for them
mutate(genres = ifelse(genres == "(no genres listed)", "Unknown", genres))
head(movies_clean)
```

```
## # A tibble: 6 × 4
##   movieId title_name                          year  genres
##     <int> <chr>                               <chr> <chr>
## 1  182337 Cinétracts                          1968  Unknown
## 2  195495 Familia                             2005  Drama
## 3    3078 Liberty Heights                     1999  Drama
## 4  134704 Comedy Central Roast of Charlie Sheen 2011  Comedy
## 5  219976 47 Hours to Live                    2019  Horror
## 6  219976 47 Hours to Live                    2019  Thriller
```

```
library(tidyverse)
links <- read.csv("C:/Users/Cikey/Desktop/data/links.csv", stringsAsFactors = FALSE)
ratings <- read.csv("C:/Users/Cikey/Desktop/data/ratings.csv", stringsAsFactors = FALSE)
tags <- read.csv("C:/Users/Cikey/Desktop/data/tags.csv", stringsAsFactors = FALSE)
str(links)
```

```
## 'data.frame':    87585 obs. of  3 variables:
##  $ movieId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ imdbId : int  114709 113497 113228 114885 113041 113277 114319 112302 114576 113189 ...
##  $ tmdbId : int  862 8844 15602 31357 11862 949 11860 45325 9091 710 ...
```

```
head(links)
```

```
##   movieId imdbId tmdbId
## 1       1 114709    862
## 2       2 113497   8844
## 3       3 113228  15602
## 4       4 114885  31357
## 5       5 113041  11862
## 6       6 113277    949
```

```
str(ratings)
```

```
## 'data.frame':    32000204 obs. of  4 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  17 25 29 30 32 34 36 80 110 111 ...
##  $ rating   : num  4 1 2 5 5 2 1 5 3 5 ...
##  $ timestamp: int  944249077 944250228 943230976 944249077 943228858 943228491 944249008 944248943 943231119 944249008 ...
```

```
head(ratings)
```

```
##   userId movieId rating timestamp
## 1      1      17      4 944249077
## 2      1      25      1 944250228
## 3      1      29      2 943230976
## 4      1      30      5 944249077
## 5      1      32      5 943228858
## 6      1      34      2 943228491
```

```
str(tags)
```

```
## 'data.frame':    2000072 obs. of  4 variables:
##  $ userId   : int  22 22 22 34 34 34 55 58 58 58 ...
##  $ movieId  : int  26479 79592 247150 2174 2174 8623 5766 7451 7451 7451 ...
##  $ tag      : chr  "Kevin Kline" "misogyny" "acrophobia" "music" ...
##  $ timestamp: int  1583038886 1581476297 1622483469 1249808064 1249808102 1249808497 1319322078 1672551536 1672551510 167255
1502 ...
```

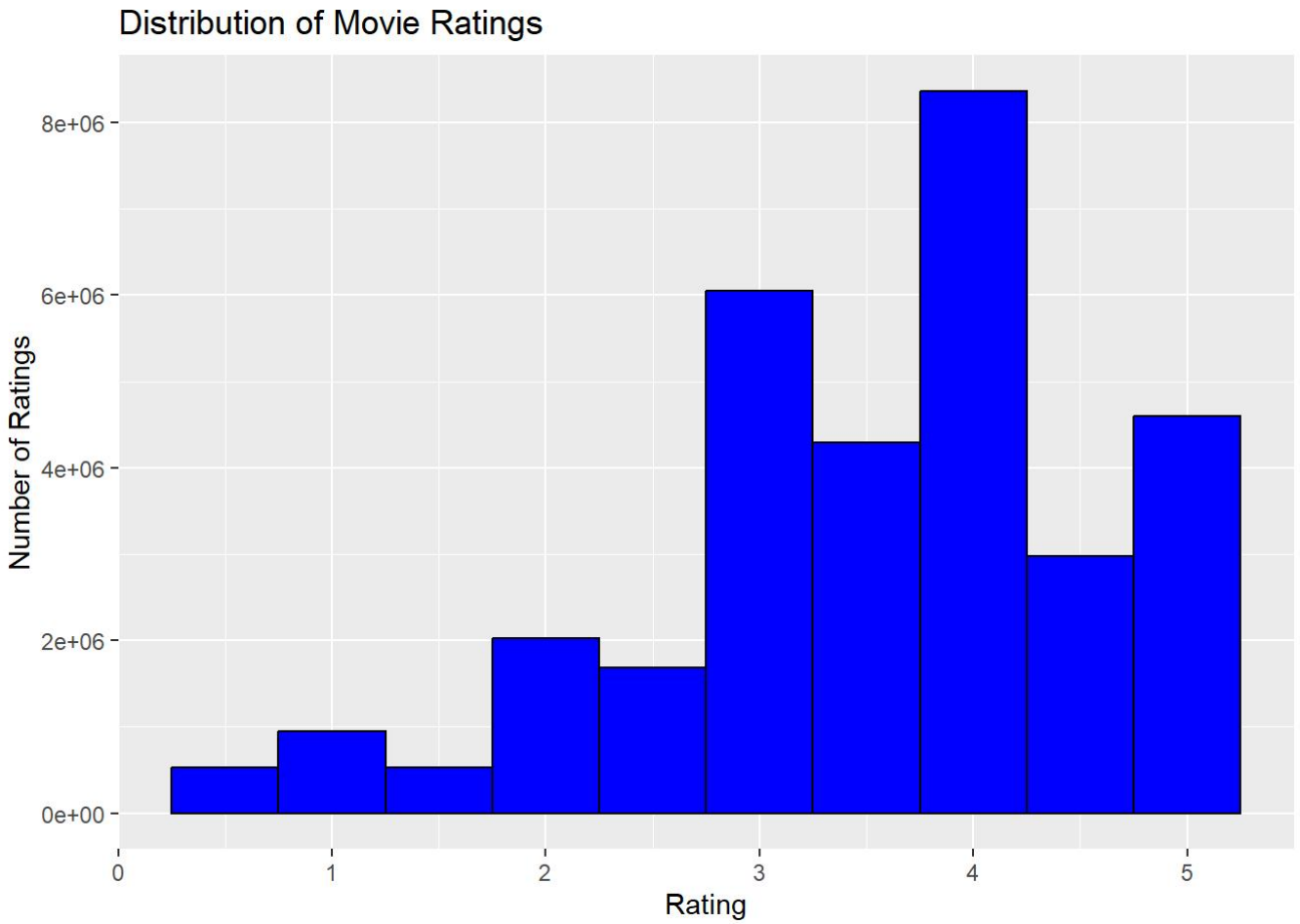
```
head(tags)
```

```
##   userId movieId          tag  timestamp
## 1     22   26479  Kevin Kline 1583038886
## 2     22   79592     misogyny 1581476297
## 3     22  247150   acrophobia 1622483469
## 4     34    2174        music 1249808064
## 5     34    2174        weird 1249808102
## 6     34    8623 Steve Martin 1249808497
```

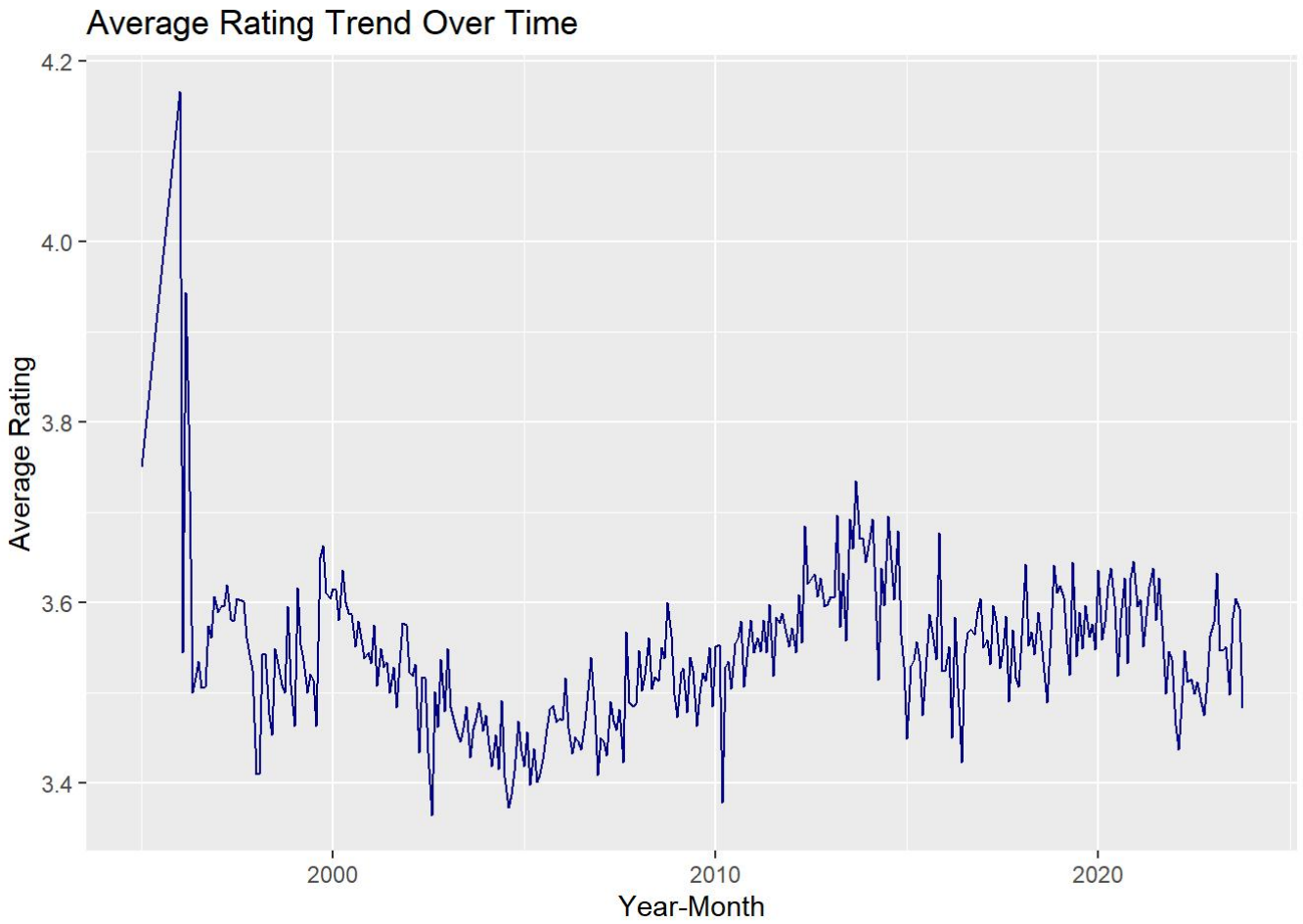
```
movie_summary <- ratings %>% #Compute the average rating and number of ratings per movie.
  group_by(movieId) %>%
  summarise(
    avg_rating = mean(rating, na.rm = TRUE),
    num_ratings = n()
  ) %>%
  arrange(desc(num_ratings))
head(movie_summary)
```

```
## # A tibble: 6 × 3
##   movieId avg_rating num_ratings
##     <int>      <dbl>       <int>
## 1     318       4.40      102929
## 2     356       4.05      100296
## 3     296       4.20       98409
## 4    2571       4.16       93808
## 5     593       4.15       90330
## 6     260       4.10       85010
```

```
ggplot(ratings, aes(x = rating)) +
  geom_histogram(binwidth = 0.5, color = "black", fill = "blue") +
  labs(title = "Distribution of Movie Ratings",
       x = "Rating",
       y = "Number of Ratings")# Explore the distribution of ratings across all users
```


Distribution of Movie Ratings
Number of Ratings
8e+06
6e+06
4e+06
2e+06
0e+00
0
1
2
3
4
5
Rating

```
ratings <- ratings %>%
  mutate(date = as.POSIXct(timestamp, origin = "1970-01-01"))
ratings <- ratings %>%
  mutate(year_month = format(date, "%Y-%m"))
monthly_trend <- ratings %>%
  group_by(year_month) %>%
  summarise(avg_rating = mean(rating))
ggplot(monthly_trend, aes(x = as.Date(paste0(year_month, "-01")), y = avg_rating)) +
  geom_line(color = "darkblue") +
  labs(title = "Average Rating Trend Over Time",
       x = "Year-Month",
       y = "Average Rating")#Convert timestamp into readable dates and analyze trends over time
```


Average Rating Trend Over Time
4.2
4.0
3.8
3.6
3.4
Average Rating
2000
2010
2020
Year-Month