Netflix Case Study

marty

2025-02-18

Introduction

The following analysis is a case study about Netflix. This analysis aims to answer the question: **What differentiates Netflix's content from other streaming services?**

Ultimately, answering this question can help move the needle with a core problem in the streaming industry: capturing subscribers

Throughout the study, I will be using data on Netflix's content from 2016-2021 alongside a database from TMDB (The Movie Database). The TMDB database contains 1.2 million observations of historical data about movies and TV shows (e.g.genres, keywords, popularity). I'll be manipulating that data to find any trends in Netflix's content over the years, and thus identify what makes them different from their competitors.

These are the datasets I'll be using:

- Netflix Movies & TV Shows (https://www.kaggle.com/datasets/shivamb/netflix-shows): created by Shivam Bansal, under the CC0: Public Domain (https://creativecommons.org/publicdomain/zero/1.0/) license
- Full TMDB Movies Database 2024 (https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies/data): created by user name asaniczka on Kaggle, under the ODC Attribution (https://opendatacommons.org/licenses/by/1-0/index.html) license
- Full TMDB TV Shows Database 2024 (https://www.kaggle.com/datasets/asaniczka/full-tmdb-tv-shows-dataset-2023-150k-shows): created by user name asaniczka on Kaggle, under the ODC Attribution (https://opendatacommons.org/licenses/by/1-0/index.html) license

Setting up the Environment

To start, I'll be installing the tidyverse library of functions for its data analysis capabilities.

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/marti/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\marti\AppData\Local\Temp\RtmpgHxsM8\downloaded_packages
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.0.4
```

```
## ── 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 becom
e errors
```

Importing the Datasets

Then, I'll be importing the datasets under the following aliases on RStudio:

1. *netflix*: a data frame containing all movies and TV shows on the Netflix streaming service from 2016-2021
2. *movies*: a data frame containing 1.1 million observations on movies, including details such as genre, keywords, audience score, budget, etc.
3. *shows*: a data frame containing 168,000 observations on TV shows, including details such as genre, descriptions, release date, etc.

```
# importing the netflix data set as "netflix"
netflix <- read.csv("C:\\Users\\marti\\Desktop\\NETFLIX CASE STUDY B\\Data Sources\\netflix_titl
es.csv")

# importing the movies data set in a "movies" table
movies <- read.csv("C:\\Users\\marti\\Desktop\\NETFLIX CASE STUDY B\\Data Sources\\movie_datase
t.csv")

# importing the shows data as in a "shows" table
shows <- read.csv("C:\\Users\\marti\\Desktop\\NETFLIX CASE STUDY B\\Data Sources\\tv_datset.cs
v")
```

Preparing the Datasets

Not every variable in the **movies** and **shows** tables was going to be useful in the analysis, so I removed the unnecessary ones below:

```
# removing unnecessary or redundant variables from the movies table
movies <- movies %>% select(-spoken_languages,-production_countries,-tagline,-poster_path,-overv
iew,-original_title,-original_language,-imdb_id,-homepage,-backdrop_path,-adult,-runtime,-statu
s,-id)

# removing unnecessary or redundant variables from the shows table
shows <- shows %>% select(-episode_run_time,-production_countries,-spoken_languages,-origin_coun
try,-networks,-overview,-languages,-tagline,-status,-type,-poster_path,-original_name,-in_produc
tion,-homepage,-backdrop_path,-adult,-original_language,-number_of_seasons,-id)
```

Cleaning the Datasets

Next, I needed to join the **netflix** table with the other two so that I could analyze trends throughout Netflix's content over the years. All three tables had unique ID values for its movies/shows, so I decided to *join the tables based on their titles (title) and release data (date_added).* Titles were already set for all the tables, so I just had to standardize the dates.

Fixing the date_added column of the netflix table

```
# Change the format of date added from a chr to a date time object
netflix$date_added <- mdy(netflix$date_added)
```

Fixing the date column of the movies table

```
#change the format of the release date column in movies from chr to a date time object
movies$release_date <- ymd(movies$release_date)

#show only the year for the release date column (in order to match the values in the netflix rel
ease year column)
movies$release_date <- year(movies$release_date)

#rename the release date column in movies to enable joining with the netflix table
movies <- movies %>% rename(release_year=release_date)
```

With dates standardized across the netflix and movies table, I could now join them together:

Creating a ̓netmovies" table that joins the netflix and movies

tables for additional movies details

```
# Join the movies and netflix table according to movie title and release date
netmovies <- merge(x=netflix,y=movies,by=c("title","release_year"),all.x=TRUE)

# Remove duplicate entries by only keeping the unique movies (filter by id and content type of t
he netflix table)
netmovies <- netmovies %>% distinct(show_id,.keep_all=TRUE) %>% filter(type=="Movie")

# Adding a year released column
netmovies <- netmovies %>% mutate(year_added = year(date_added))
```

Note: There are also null values for a number of movies (1545 fields) on additional variables such as **vote_average, vote_count, revenue, budget, & popularity,** but I chose to keep these records in the analysis because of the **genre, year_added, and rating** variables

Next, I had to join the netflix and shows table:

Preparing the shows table

```
#
#change the format of the release date column in shows from chr to a date time object
shows$first_air_date <- ymd(shows$first_air_date)

#show only the year for the release date column (in order to match the values in the netflix rel
ease year column)
shows$first_air_date <- year(shows$first_air_date)

#rename the title and date columns in the shows table to enable joining with the netflix table
shows <- shows %>% rename(release_year=first_air_date,title=name)
```

Creating a ̓netshows” table that joins the netflix and shows tables for additional show details

```
# Join the shows and netflix table according to show name and release date
netshows <- merge(x=netflix,y=shows,by=c("title","release_year"),all.x=TRUE)

# Remove duplicate entries by only keeping the unique show ids (and TV shows of the netflix tabl
e)
netshows <- netshows %>% distinct(show_id,.keep_all=TRUE) %>% filter(type=="TV Show")

# Adding a year released column
netshows <- netshows %>% mutate(year_added = year(date_added))
```

With that being done, I can now conduct my analysis on these two primary tables: *netmovies* and *netshows*

Analysis

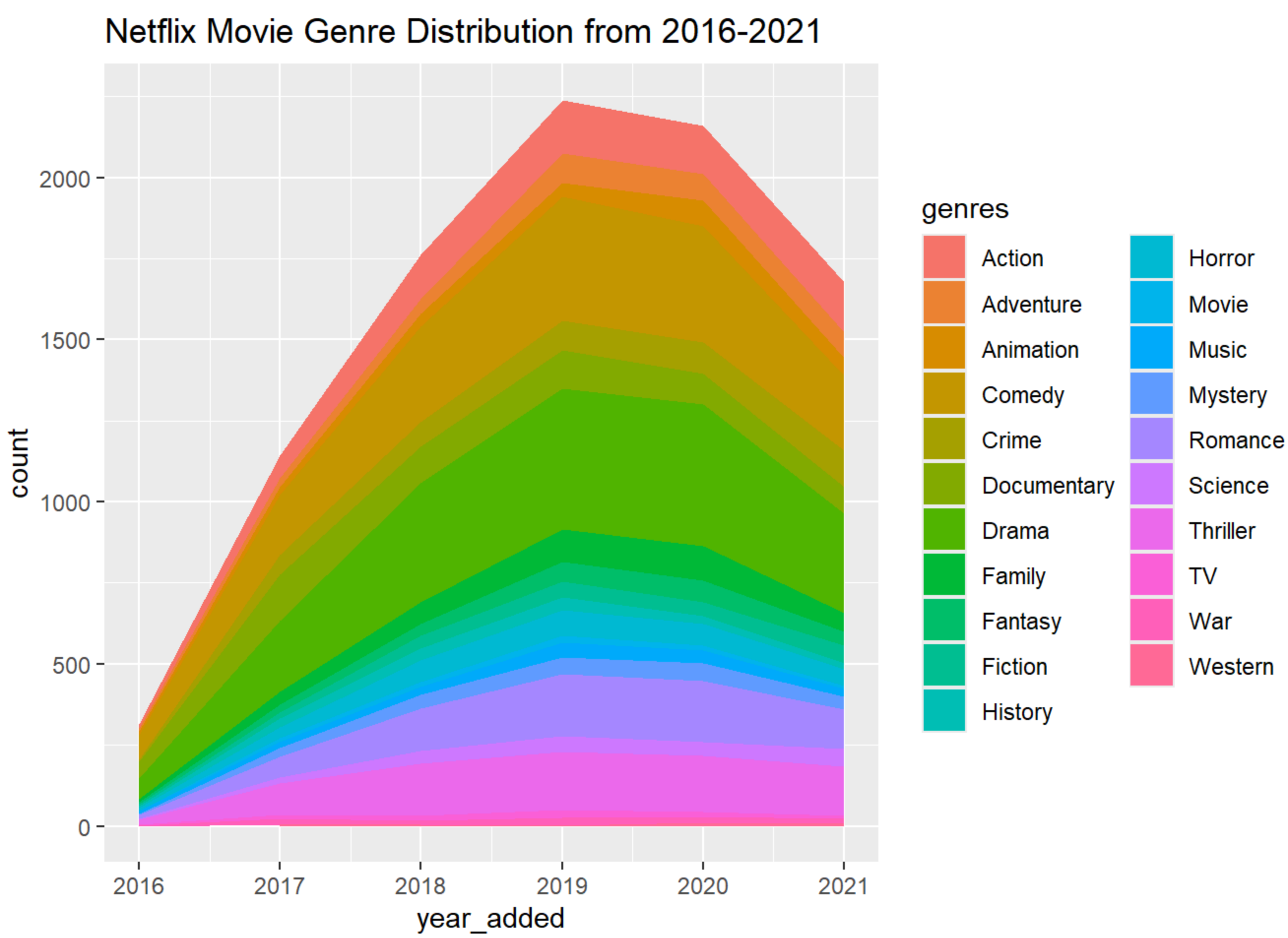
Distribution of Genres over Time

Plotting the genres of Netflix's content over time can be a general measure of users' entertainment preferences. It can show which genres perform well, are growing, or remain the same. Some quick insights on the plot below:

- *Crime/Comedy/Drama seems to be Netflix's "bread and butter" genres* (marked in yellow and green) for movies, since it comprises more than 60% of their releases every year
- 2017-2019 saw the most growth for content released by Netflix. Within that time period, Netflix added *Drama (+216 movies), Comedy (+195), and Romance (+127) movie genres the most.*

```
# Plot the genres of all movies throughout the years for Netflix
netmovies %>% separate_rows(genres) %>% filter(genres != "" & year_added > 2015 & year_added < 2
022) %>% group_by(year_added,genres) %>% summarize(count=n()) %>% arrange(desc(count)) %>% ggplo
t(aes(x=year_added,y=count,fill=genres)) + geom_area(stat="identity") + xlim(2016,2021) + labs(t
itle="Netflix Movie Genre Distribution from 2016-2021")
```

```
## `summarise()` has grouped output by 'year_added'. You can override using the
## `.groups` argument.
```



```r
# Observing which genres saw the most growth for movies
growthmovies <- netmovies %>% separate_rows(genres) %>% filter(genres != "" & year_added > 2016
& year_added < 2020) %>% group_by(genres, year_added) %>% summarize(count=n()) %>% print(n=70)
```

```
## `summarise()` has grouped output by 'genres'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 63 × 3
## # Groups:   genres [21]
##    genres      year_added count
##    <chr>            <dbl> <int>
##  1 Action            2017    72
##  2 Action            2018   136
##  3 Action            2019   166
##  4 Adventure         2017    25
##  5 Adventure         2018    46
##  6 Adventure         2019    90
##  7 Animation         2017    26
##  8 Animation         2018    37
##  9 Animation         2019    43
## 10 Comedy            2017   186
## 11 Comedy            2018   295
## 12 Comedy            2019   381
## 13 Crime             2017    60
## 14 Crime             2018    77
## 15 Crime             2019    92
## 16 Documentary       2017   143
## 17 Documentary       2018   113
## 18 Documentary       2019   119
## 19 Drama             2017   219
## 20 Drama             2018   367
## 21 Drama             2019   435
## 22 Family            2017    39
## 23 Family            2018    65
## 24 Family            2019   100
## 25 Fantasy           2017    23
## 26 Fantasy           2018    36
## 27 Fantasy           2019    60
## 28 Fiction           2017    20
## 29 Fiction           2018    40
## 30 Fiction           2019    49
## 31 History           2017    26
## 32 History           2018    38
## 33 History           2019    38
## 34 Horror            2017    34
## 35 Horror            2018    65
## 36 Horror            2019    78
## 37 Movie             2017    11
## 38 Movie             2018    15
## 39 Movie             2019    24
## 40 Music             2017    18
## 41 Music             2018    25
## 42 Music             2019    43
## 43 Mystery           2017    27
## 44 Mystery           2018    43
## 45 Mystery           2019    53
## 46 Romance           2017    64
## 47 Romance           2018   131
## 48 Romance           2019   191
```

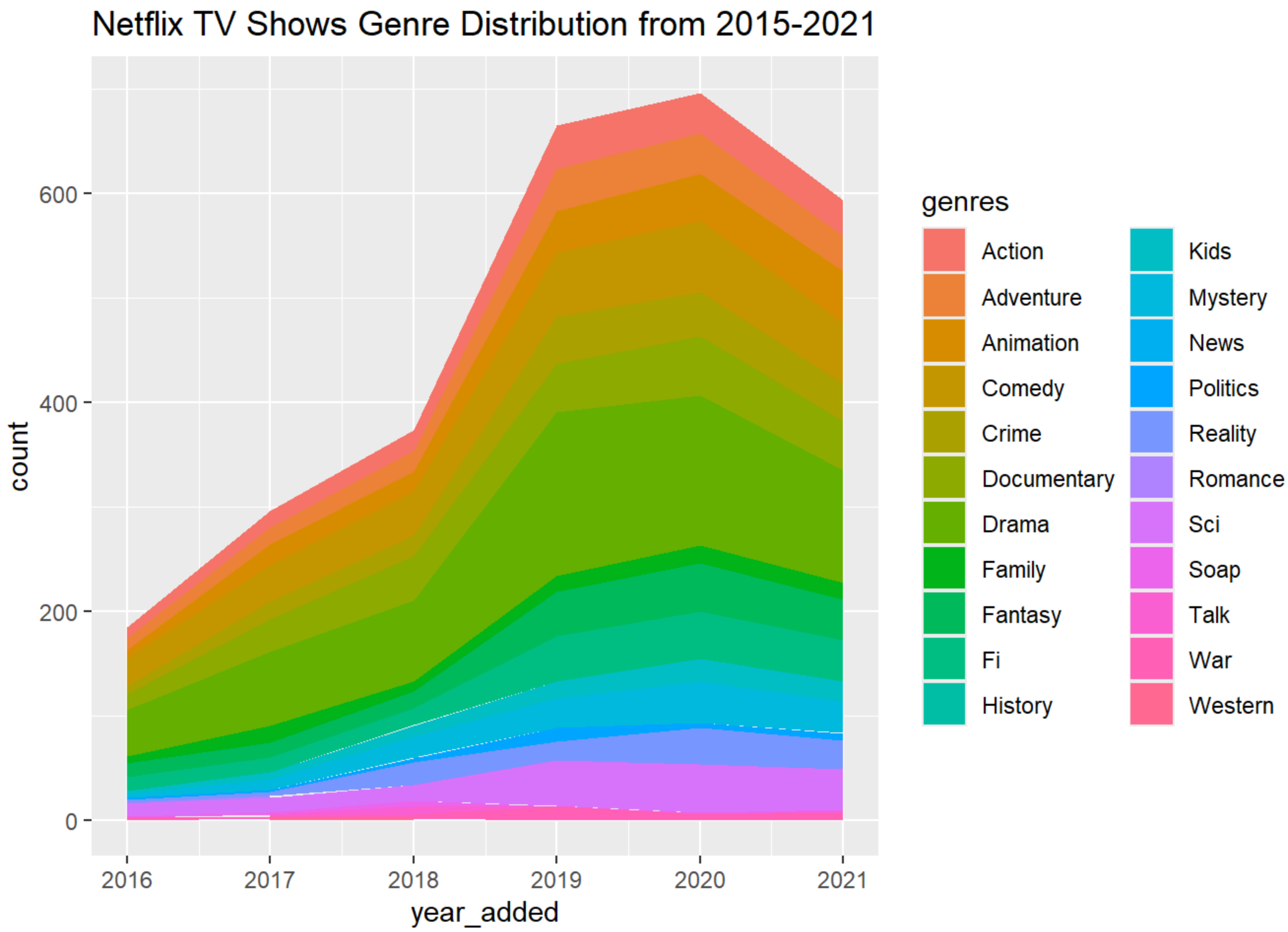
```
## 49 Science          2017    20
## 50 Science          2018    40
## 51 Science          2019    49
## 52 TV               2017    11
## 53 TV               2018    15
## 54 TV               2019    24
## 55 Thriller         2017    97
## 56 Thriller         2018   160
## 57 Thriller         2019   179
## 58 War              2017    19
## 59 War              2018    12
## 60 War              2019    23
## 61 Western          2017     4
## 62 Western          2018     5
## 63 Western          2019     3
```

Looking at the plot for TV Shows,

**Drama was a top priority*, accounting for the most releases out of all genres every year* 2017-2019 saw the most growth for content released by Netflix. Within that time period, *Drama (+80 shows), Sci-Fi (+27), and Crime (+25) saw the biggest growth amongst all TV shows*

```
# Plot the genres of all shows throughout the years

netshows %>% separate_rows(genres) %>% filter(genres != "" & year_added > 2015 & year_added < 20
22) %>% group_by(year_added,genres) %>% summarize(count=n()) %>% ggplot(aes(x=year_added,y=coun
t,fill=genres)) + geom_area(stat="identity") + xlim(2016,2021) + labs(title="Netflix TV Shows Ge
nre Distribution from 2015-2021")
```

```
## `summarise()` has grouped output by 'year_added'. You can override using the
## `.groups` argument.
```




```
# Observing which genres saw the most growth for tv shows
growthtv <- netshows %>% separate_rows(genres) %>% filter(genres != "" & year_added > 2017 & yea
r_added < 2020) %>% group_by(genres, year_added) %>% summarize(count=n()) %>% print(n=40)
```

```
## `summarise()` has grouped output by 'genres'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 38 × 3
## # Groups:   genres [21]
##    genres      year_added count
##    <chr>            <dbl> <int>
##  1 Action            2018    20
##  2 Action            2019    41
##  3 Adventure         2018    20
##  4 Adventure         2019    41
##  5 Animation         2018    19
##  6 Animation         2019    39
##  7 Comedy            2018    42
##  8 Comedy            2019    62
##  9 Crime             2018    20
## 10 Crime             2019    45
## 11 Documentary       2018    43
## 12 Documentary       2019    46
## 13 Drama             2018    77
## 14 Drama             2019   157
## 15 Family            2018    10
## 16 Family            2019    15
## 17 Fantasy           2018    16
## 18 Fantasy           2019    43
## 19 Fi                2018    16
## 20 Fi                2019    43
## 21 History           2018     1
## 22 Kids              2018    10
## 23 Kids              2019    16
## 24 Mystery           2018    20
## 25 Mystery           2019    29
## 26 News              2018     1
## 27 Politics          2018     4
## 28 Politics          2019    13
## 29 Reality           2018    21
## 30 Reality           2019    18
## 31 Sci               2018    16
## 32 Sci               2019    43
## 33 Soap              2018     5
## 34 Talk              2018     8
## 35 Talk              2019     1
## 36 War               2018     4
## 37 War               2019    13
## 38 Western           2018     1
```

Ratings for Content

Generally, ratings are an indicator of the *minimum** appropriate age demographic for a form of entertainment. Analyzing the ratings of Netflix's content over the years may reveal things about its target market (age-wise).

Using Netflix's maturity ratings (https://help.netflix.com/en/node/2064) as a guide, *a large percentage of their content (43% of TV shows and 46% of movies) are recommended for adults.*

```
# A general overview of the most common ratings on Netflix movies
movieratings <- netmovies %>% group_by(rating) %>% summarize(movies=n()) %>% arrange(desc(movie
s)) %>% filter(rating != "" & movies > 2)
movieratings <- movieratings %>% mutate(percentage_of_total=round((movies/sum(movies))*100,digit
s=2))
```

```
# A general overview of the most common ratings on Netflix TV shows
showratings <- netshows %>% filter(rating != "") %>% group_by(rating) %>% summarize(shows=n()) %
>% arrange(desc(shows))
showratings <- showratings %>% mutate(percentage_of_total=round((shows/sum(shows))*100,digits=
2))
```

Identifying the characteristics of an adult-rated movie:

```
# What constitutes an adult movie? Create a table of all movies rated TV-MA or R

adultmovies <- netmovies %>% filter(rating=="TV-MA" | rating=="R")

# Most common genres with an R or TV-MA rating
adultmovies %>% separate_rows(genres,sep=",") %>% group_by(genres) %>% summarize(movies=n()) %>%
filter(genres != "" | !is.na(genres)) %>% arrange(desc(movies)) %>% head(n=10)
```

```
## # A tibble: 10 × 2
##    genres        movies
##    <chr>          <int>
##  1 "Comedy"         574
##  2 "Drama"          519
##  3 " Drama"         413
##  4 " Thriller"      370
##  5 "Documentary"    232
##  6 "Action"         224
##  7 " Crime"         211
##  8 " Romance"       160
##  9 "Horror"         141
## 10 "Thriller"       139
```

```
# Most common keywords in movies with an R or TV-MA rating
adultmovies %>% separate_rows(keywords,sep=",") %>% group_by(keywords) %>% summarize(movies=n())
%>% filter(keywords != "" | !is.na(keywords)) %>% arrange(desc(movies)) %>% head(n=10)
```

```
## # A tibble: 10 × 2
##    keywords                    movies
##    <chr>                        <int>
##  1 ""                             573
##  2 "stand-up comedy"              174
##  3 " woman director"              104
##  4 " murder"                       90
##  5 " based on novel or book"       63
##  6 " based on true story"          56
##  7 " revenge"                      52
##  8 " california"                   46
##  9 " biography"                    44
## 10 " lgbt"                         41
```

```
# Most common words found in descriptions of R or TV-MA rated movies
moviedescriptions <- adultmovies %>% separate_rows(description, sep=" ") %>% group_by(descriptio
n) %>% summarize(movies=n()) %>% filter(description != "" | !is.na(description)) %>% arrange(des
c(movies)) %>% head(n=100)
```

Identifying the characteristics of an adult-rated show:

```
# What constitutes an adult show? Create a table of all shows rated TV-MA or R
adultshows <- netshows %>% filter(rating=="TV-MA" | rating=="R")

# Most common genres in shows with an R or TV-MA rating
adultshows %>% separate_rows(genres,sep=",") %>% group_by(genres) %>% summarize(shows=n()) %>% f
ilter(genres != "" | !is.na(genres)) %>% arrange(desc(shows)) %>% head(n=10)
```

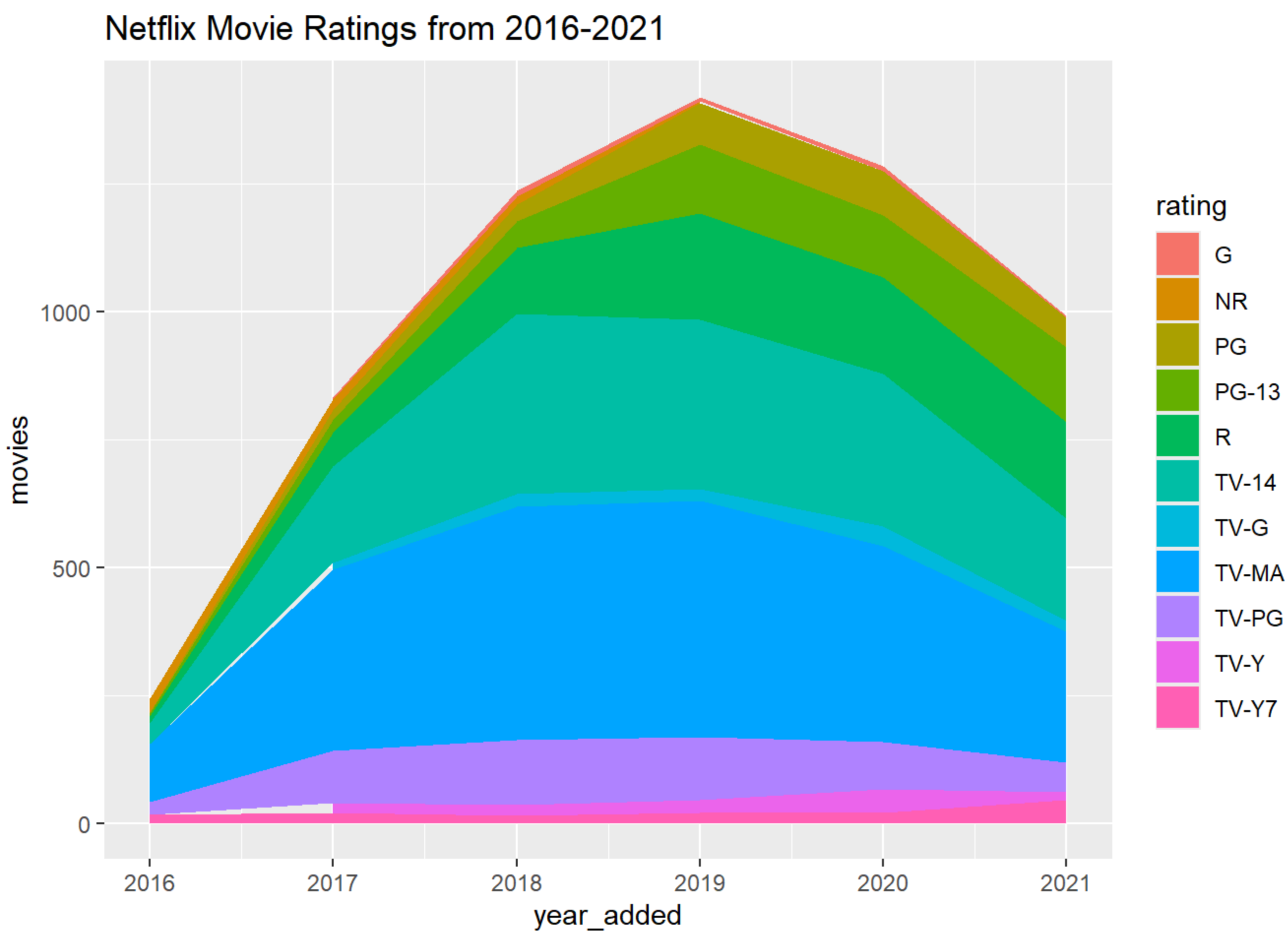
```
## # A tibble: 10 × 2
##    genres                 shows
##    <chr>                  <int>
##  1 "Drama"                  207
##  2 " Drama"                 141
##  3 "Documentary"            105
##  4 "Comedy"                  95
##  5 " Crime"                  84
##  6 " Mystery"                73
##  7 "Crime"                   51
##  8 " Sci-Fi & Fantasy"       50
##  9 " Action & Adventure"     43
## 10 " Comedy"                 38
```

```
# Most common words found in descriptions of R or TV-MA rated shows
showdescriptions <- adultshows %>% separate_rows(description, sep=" ") %>% group_by(description)
%>% summarize(shows=n()) %>% filter(description != "" | !is.na(description)) %>% arrange(desc(sh
ows)) %>% head(n=100)
```

Looking at a distribution of the ratings over time, we can see below that Netflix chose to *expand on content appropriate for adults and teenagers (not suitable for ages under 14).*

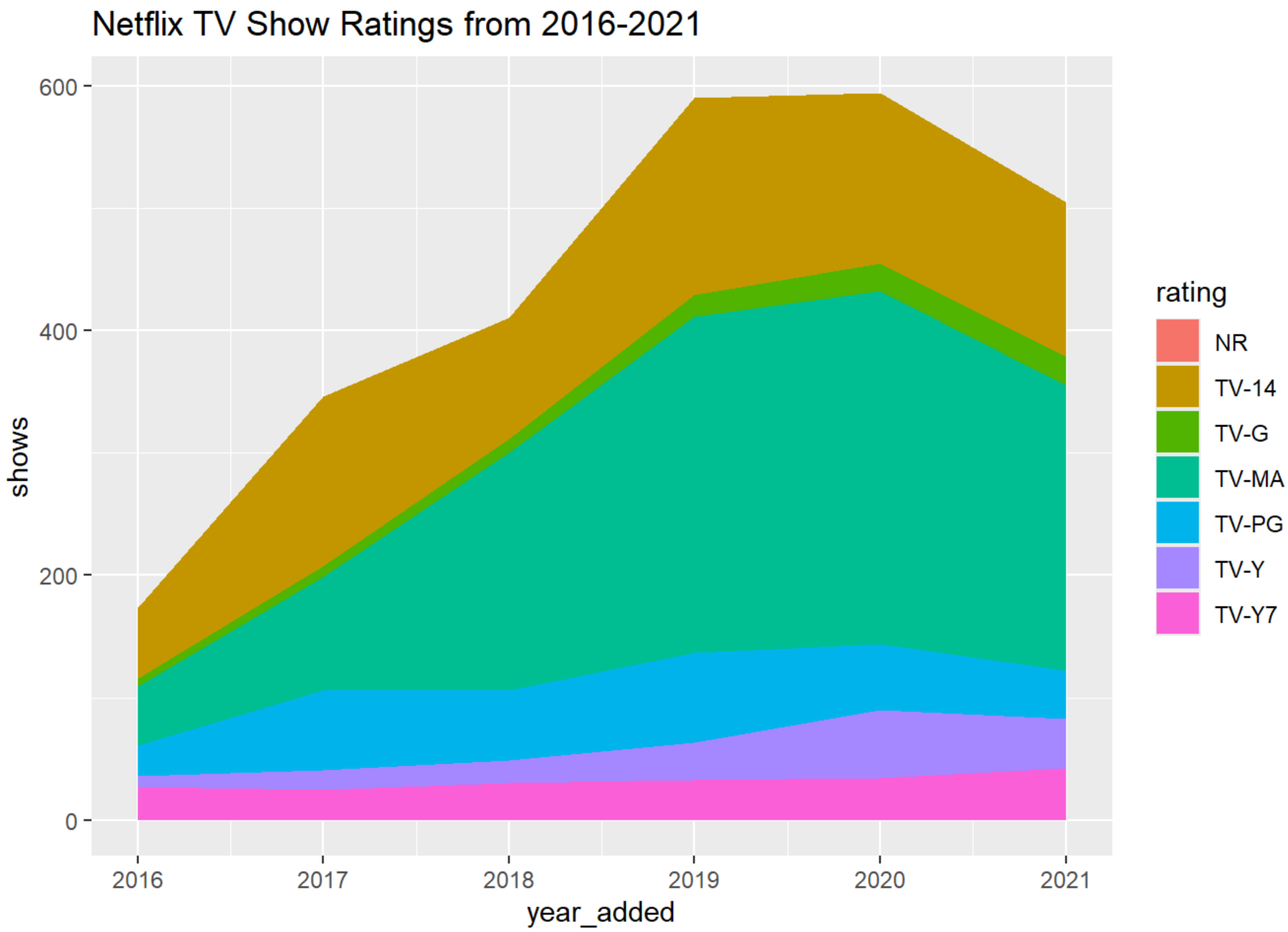
```
# Plotting the ratings of Netflix movie releases over time

netmovies %>% group_by(rating, year_added) %>% summarize(movies=n()) %>%  filter(rating != "" & year_added > 2015 & year_added < 2022 & movies > 2) %>% ggplot(aes(x=year_added,y=movies,fill=rating)) + geom_area(stat="identity") + xlim(2016,2021) + labs(title="Netflix Movie Ratings from 2016-2021")
```

```
## `summarise()` has grouped output by 'rating'. You can override using the
## `.groups` argument.
```




```
# Plotting the ratings of Netflix TV show releases over time
netshows %>% group_by(rating, year_added) %>% summarize(shows=n()) %>%  filter(rating != "" & year_added > 2015 & year_added < 2022 & shows > 2) %>% ggplot(aes(x=year_added,y=shows,fill=rating)) + geom_area(stat="identity") + xlim(2016,2021) + labs(title="Netflix TV Show Ratings from 2016-2021")
```

```
## `summarise()` has grouped output by 'rating'. You can override using the
## `.groups` argument.
```



Analyzing Keywords of Netflix Content

Keywords is a column that identifies common words or phrases associated with a specific movie or TV show. Studying the most frequently occurring keywords, or even its correlations with metrics such as popularity or audience scores can be an approximate measure of user preference (and thus performance)

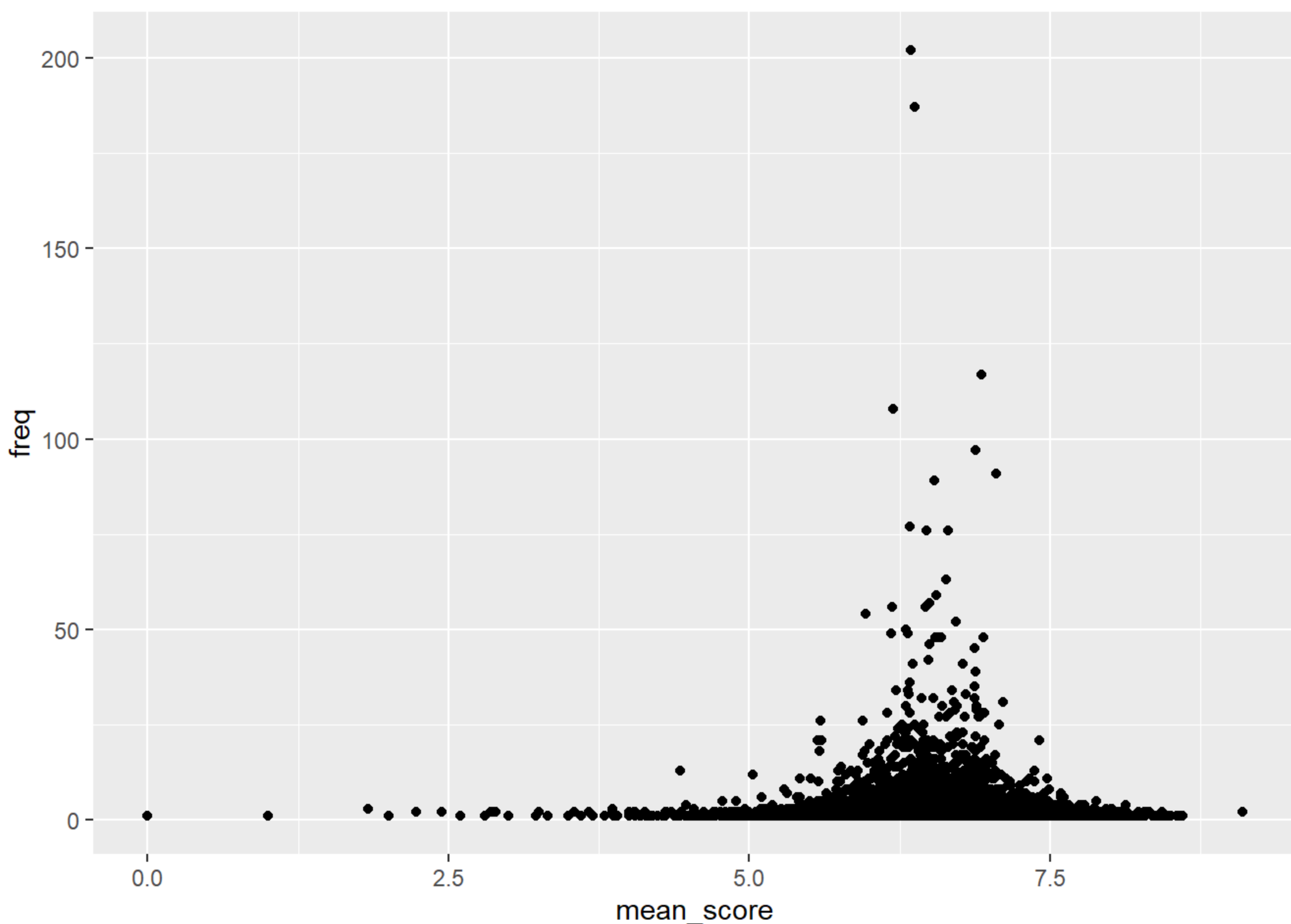
```
# Creating a table of the most commonly used keywords and their average audience scores for Netflix movies

moviekeywords <- netmovies %>% separate_rows(keywords, sep=",") %>% group_by(keywords) %>% summarize(total_votes=sum(vote_count),mean_score=mean(vote_average), avg_revenue=mean(revenue),avg_release_yr=mean(release_year),freq=n()) %>% arrange(desc(freq)) %>% filter(keywords != "" & !is.na(keywords))

head(moviekeywords)
```

```
## # A tibble: 6 × 6
##   keywords               total_votes mean_score avg_revenue avg_release_yr  freq
##   <chr>                        <int>      <dbl>       <dbl>          <dbl> <int>
## 1 "stand-up comedy"             9446       6.34     116832.          2016.   202
## 2 " woman director"           174155       6.37   25384136.          2013.   187
## 3 " based on novel or b…      457896       6.93  114477661.          2005.   117
## 4 " murder"                   153831       6.20   23647976.          2012.   108
## 5 " based on true story"      226735       6.88   50860198.          2013.    97
## 6 " biography"                147199       7.05   32488232.          2014.    91
```

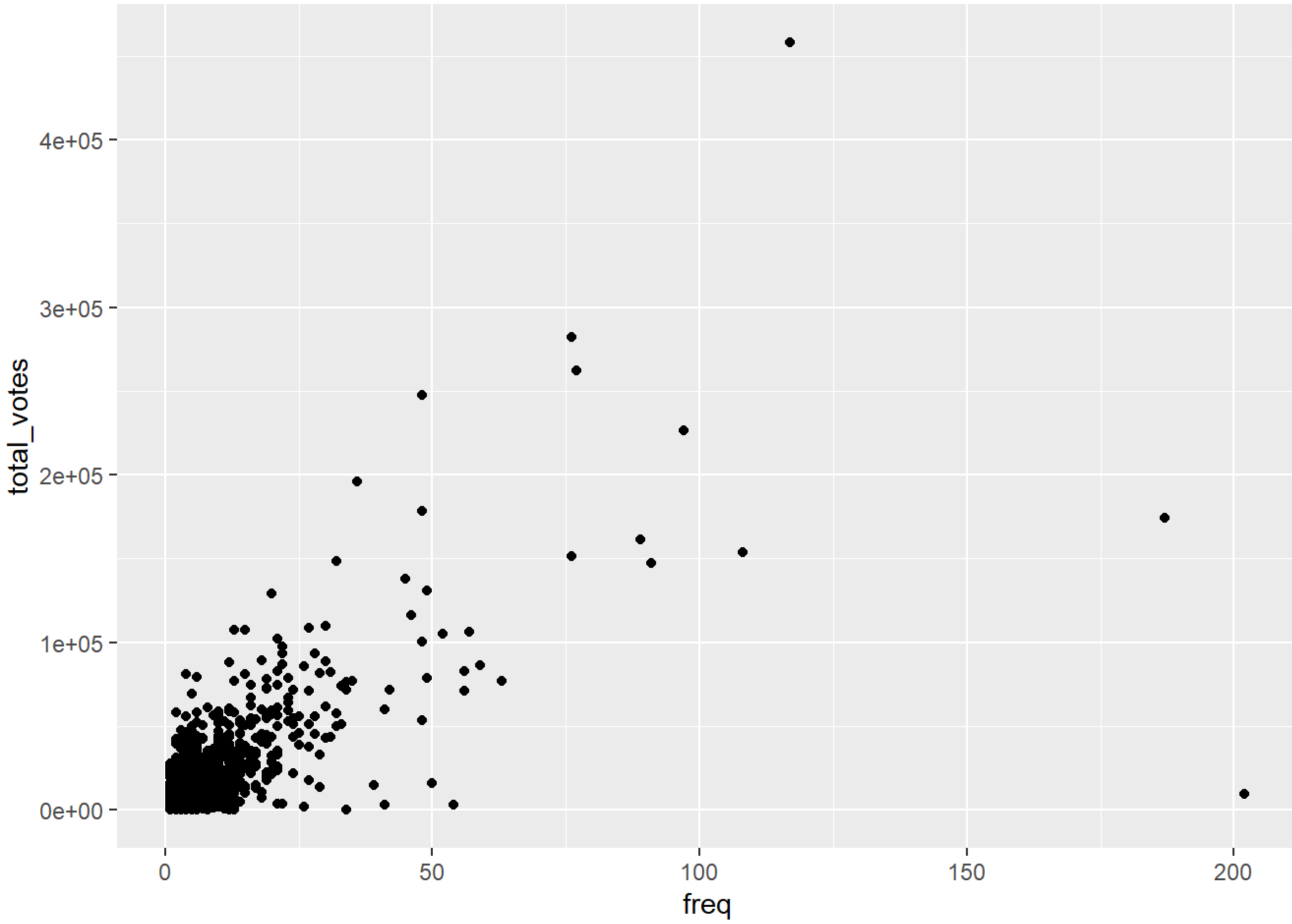
The most common keywords amongst Netflix titles are **stand-up comedy, woman director, based on novel or book, murder, or based on true story.**

With the plots below, it's evident that the commonly preferred movies on Netflix are *more popular (high engagement from audiences/high vote counts) than good (high audience scores).*

```
# Plotting the correlation of common keywords with the movie's average audience score
ggplot(data=moviekeywords) + geom_point(mapping=aes(x=mean_score,y=freq))
```




```
# Plotting the correlation of common keywords with a movie's popularity
ggplot(data=moviekeywords) + geom_point(mapping=aes(x=freq,y=total_votes))
```


4e+05
3e+05
2e+05
1e+05
0e+00
total_votes
0
50
100
150
200
freq