Data cleaning and visualization

Zhiwei Lin

2023-01-20

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(skimr)
library(ggplot2)
library(ggrepel)
```

```
data <-read.csv('/Users/zhiweilin/Downloads/netflix_titles.csv', header =T, na.string=c("","NA"))
```

```
head(data) # observe first 6 rows
```

```
##   show_id    type                  title        director
## 1      s1   Movie  Dick Johnson Is Dead Kirsten Johnson
## 2      s2 TV Show          Blood & Water            <NA>
## 3      s3 TV Show              Ganglands Julien Leclercq
## 4      s4 TV Show Jailbirds New Orleans            <NA>
## 5      s5 TV Show           Kota Factory            <NA>
## 6      s6 TV Show          Midnight Mass   Mike Flanagan
##
## 1
## 2 Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Gr
## 3
## 4
## 5
## 6                                                                                    Kate Siegel, Zach Gilford,
##         country         date_added release_year rating  duration
## 1 United States September 25, 2021         2020  PG-13    90 min
## 2  South Africa September 24, 2021         2021  TV-MA 2 Seasons
## 3          <NA> September 24, 2021         2021  TV-MA  1 Season
## 4          <NA> September 24, 2021         2021  TV-MA  1 Season
## 5         India September 24, 2021         2021  TV-MA 2 Seasons
## 6          <NA> September 24, 2021         2021  TV-MA  1 Season
```

```
##                                                    listed_in
## 1                                              Documentaries
## 2              International TV Shows, TV Dramas, TV Mysteries
## 3 Crime TV Shows, International TV Shows, TV Action & Adventure
## 4                                    Docuseries, Reality TV
## 5        International TV Shows, Romantic TV Shows, TV Comedies
## 6                           TV Dramas, TV Horror, TV Mysteries
##
## 1 As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive ar
## 2      After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school s
## 3       To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of i
## 4      Feuds, flirtations and toilet talk go down among the incarcerated women at the Orleans Justice
## 5 In a city of coaching centers known to train India's finest collegiate minds, an earnest but unexce
## 6 The arrival of a charismatic young priest brings glorious miracles, ominous mysteries and renewed i
```

```
# convert variable type and rating to factor, and convert date_added to date variable
data <- mutate_at(data, vars(type,rating), as.factor)
data<-mutate(data,date_added = as.Date(date_added,format="%B %d, %Y"))
```

Data Summarization

```
summary(data)
```

```
##    show_id               type         title             director
##  Length:8807        Movie  :6131   Length:8807        Length:8807
##  Class :character   TV Show:2676   Class :character   Class :character
##  Mode  :character                  Mode  :character   Mode  :character
##
##
##
##
##      cast              country            date_added          release_year
##  Length:8807        Length:8807        Min.   :2008-01-01   Min.   :1925
##  Class :character   Class :character   1st Qu.:2018-04-20   1st Qu.:2013
##  Mode  :character   Mode  :character   Median :2019-07-12   Median :2017
##                                        Mean   :2019-05-23   Mean   :2014
##                                        3rd Qu.:2020-08-26   3rd Qu.:2019
##                                        Max.   :2021-09-25   Max.   :2021
##                                        NA's   :98
##      rating       duration          listed_in         description
##  TV-MA  :3207   Length:8807        Length:8807        Length:8807
##  TV-14  :2160   Class :character   Class :character   Class :character
##  TV-PG  : 863   Mode  :character   Mode  :character   Mode  :character
##  R      : 799
##  PG-13  : 490
##  (Other):1284
##  NA's   :   4
```

```
skim_without_charts(data) # another summary function
```

Table 1: Data summary

Name	data
Number of rows	8807
Number of columns	12
Column type frequency:	
character	8
Date	1
factor	2
numeric	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
show_id	0	1.00	2	5	0	8807	0
title	0	1.00	1	104	0	8807	0
director	2634	0.70	2	208	0	4528	0
cast	825	0.91	3	771	0	7692	0
country	831	0.91	4	123	0	748	0
duration	3	1.00	5	10	0	220	0
listed_in	0	1.00	6	79	0	514	0
description	0	1.00	61	248	0	8775	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date_added	98	0.99	2008-01-01	2021-09-25	2019-07-12	1699

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
type	0	1	FALSE	2	Mov: 6131, TV : 2676
rating	4	1	FALSE	17	TV-: 3207, TV-: 2160, TV-: 863, R: 799

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
release_year	0	1	2014.18	8.82	1925	2013	2017	2019	2021

From summary table above, there are missing values in director, cast, country, duration, date_added and rating variables. In particular, there are 2634 missing values in director column, dropping this huge amount of missing data will severely skewed the data analysis result. We'll drop or impute the missing values if it's nesscessarily in following analysis.

```
data<-data[!duplicated(data$show_id), ] # drop any duplicated value based on show_id
```

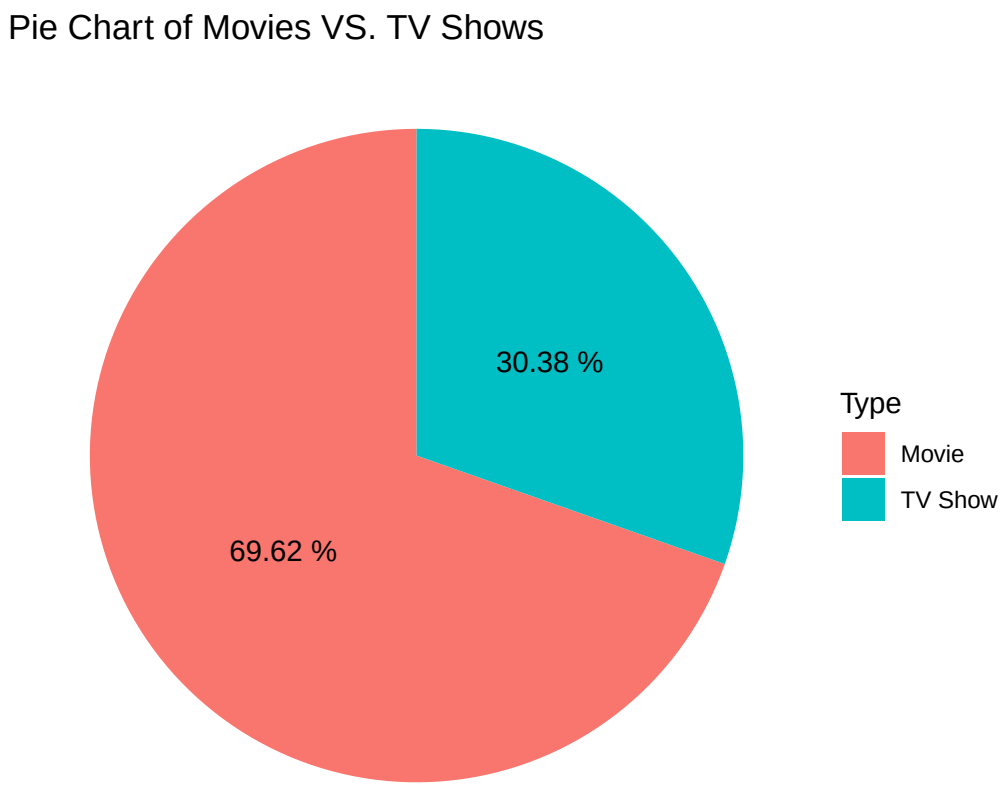
data visualization

Number & precentage of TV shows and Movies are on Netflix

```
df_type<-data %>%
  group_by(type) %>%
  summarise(
    count=n()
  )
df_type$percentage <- 100*prop.table(df_type$count)
print(df_type)
```

```
## # A tibble: 2 x 3
##   type    count percentage
##   <fct>   <int>      <dbl>
## 1 Movie    6131       69.6
## 2 TV Show  2676       30.4
```

```
ggplot(df_type, aes(x="", y=percentage, fill=type)) +
  geom_bar(width=1,stat="identity") +
  coord_polar(theta="y", start=0) +
  theme_void()+
  labs(title="Pie Chart of Movies VS. TV Shows", fill="Type")+
   geom_text(aes(label = paste(round(percentage,2), "%")),position = position_stack(vjust = 0.5),color =
```



number of movies and TV shows by rating

drop missing values in rating column becasue we want to analyze the rating of content. Missing value in rating is likely missing at random and only 4 missing values. Hence, Removing them would not severely affect the result of analysis.

```
data <- drop_na(data, rating)
levels(data$rating) # we obsevered that rating has some strange levels such as 66mins, 74mins and 84 mi
```

```
##  [1] "66 min"   "74 min"   "84 min"   "G"        "NC-17"    "NR"
##  [7] "PG"       "PG-13"    "R"        "TV-14"    "TV-G"     "TV-MA"
## [13] "TV-PG"    "TV-Y"     "TV-Y7"    "TV-Y7-FV" "UR"
```

```
data = filter(data, rating != "66 min" & rating != "74 min" & rating != "84 min")
```

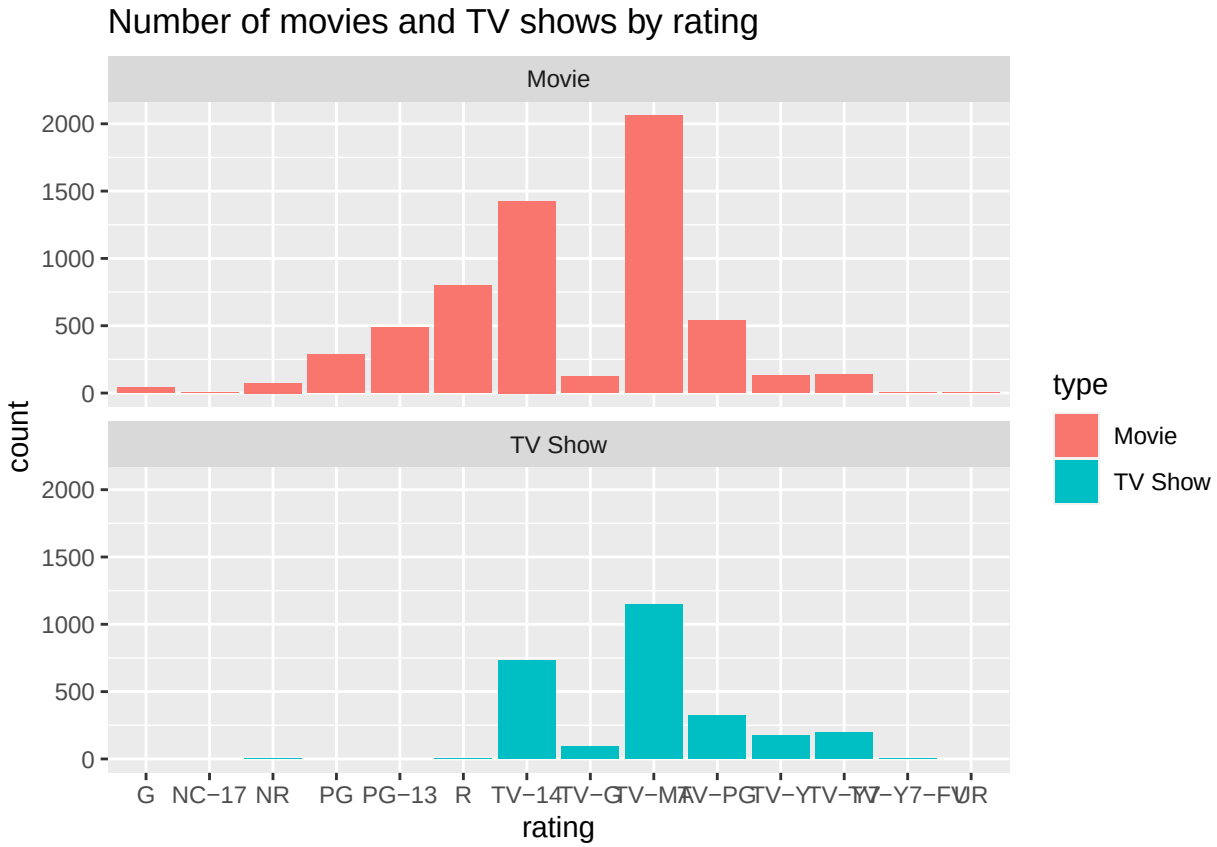
```
df_type_rating<-data %>%
  group_by(type,rating) %>%
  summarise(
    count=n()
  )
```

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

```
print(df_type_rating)
```

```
## # A tibble: 23 x 3
## # Groups:   type [2]
##    type  rating count
##    <fct> <fct>  <int>
##  1 Movie G         41
##  2 Movie NC-17      3
##  3 Movie NR        75
##  4 Movie PG       287
##  5 Movie PG-13    490
##  6 Movie R        797
##  7 Movie TV-14   1427
##  8 Movie TV-G     126
##  9 Movie TV-MA   2062
## 10 Movie TV-PG    540
## # ... with 13 more rows
```

```
ggplot(df_type_rating)+
  geom_bar(aes(x=rating,y=count,fill=type),stat="identity",position="dodge")+
  facet_wrap(~type,ncol=1)+
  labs(title="Number of movies and TV shows by rating")
```



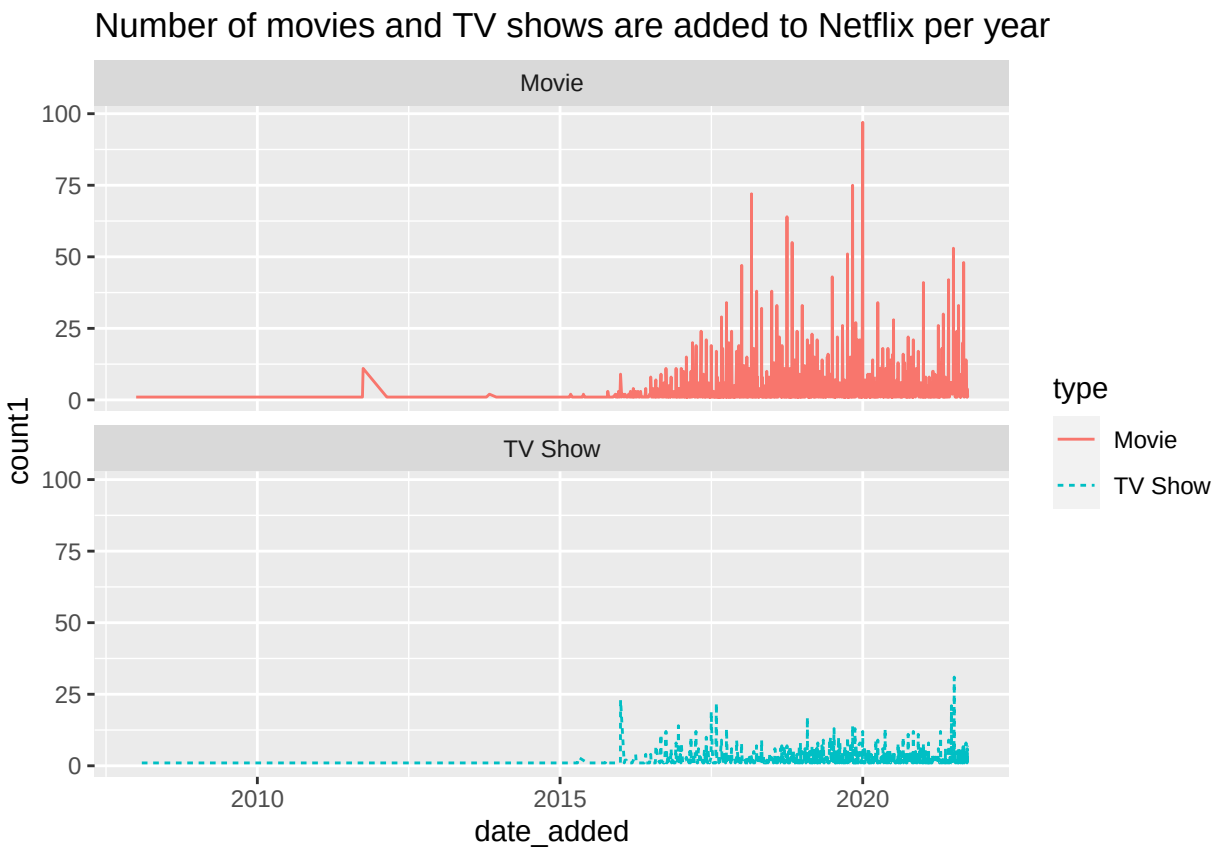
number of movies and TV shows are added to Netflix per year

```
df_type_year<-data %>%
  group_by(type,date_added) %>%
  summarise(
    count1=n(),
  )
```

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

```
ggplot(df_type_year,aes(x=date_added,y=count1,gruop=type))+
  geom_line(aes(linetype=type,color=type))+
  facet_wrap(~type,ncol=1)+
  labs(title="Number of movies and TV shows are added to Netflix per year")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



World Map

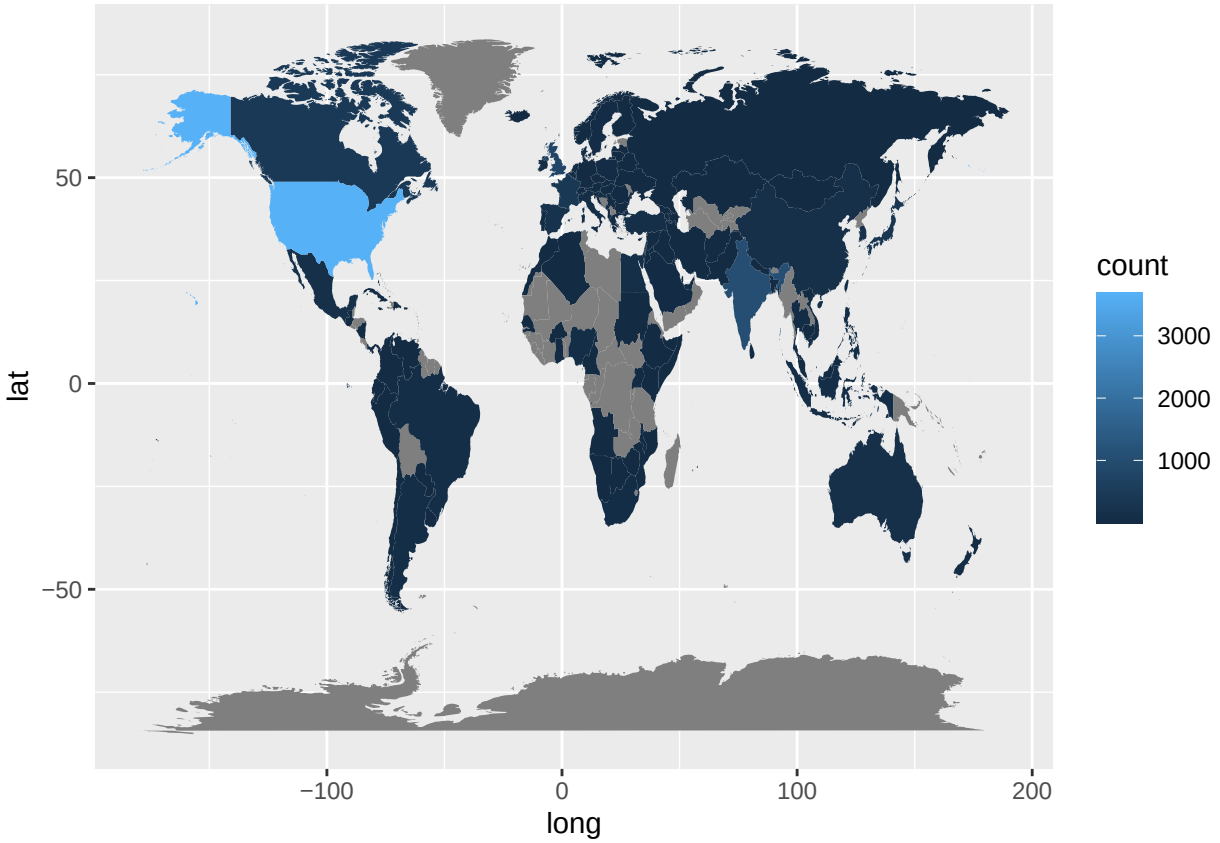
```
df<-data %>%
  separate_rows(country, sep=", |,") %>%
  group_by(country) %>%
  summarise(
    count=n()
  )
```

```
world <- map_data('world')
world_map <- left_join(df, world, by = c("country" = "region"))
not_matched<-world_map[is.na(world_map$long),"country"]
print(not_matched)
```

```
## # A tibble: 9 x 1
##   country
##   <chr>
## 1 ""
## 2 "East Germany"
## 3 "Hong Kong"
## 4 "Soviet Union"
## 5 "United Kingdom"
## 6 "United States"
## 7 "Vatican City"
## 8 "West Germany"
## 9  <NA>
```

```
df$country[df$country == 'East Germany'] <- 'Germany'
df$country[df$country == 'Hong Kong'] <- 'China'
df$country[df$country == 'United Kingdom'] <- 'UK'
df$country[df$country == 'United States'] <- 'USA'
df$country[df$country == 'Vatican City'] <- 'Vatican'
df$country[df$country == 'West Germany'] <- 'Germany'
```

```
world %>%
  merge(df, by.x = "region", by.y="country", all.x =T) %>%
  arrange(group,order) %>%
  ggplot(aes(x=long, y=lat, group = group, fill = count)) + geom_polygon()
```


50
0
−50
lat
−100
0
100
200
long
count
3000
2000
1000