

Data Challenge: Netflix

Mitchelle Mojekwu

5/7/2022

```
library(broom)
library(knitr)
library(tidyverse)

## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr 0.3.4
## v tibble 3.1.6       v dplyr 1.0.7
## v tidyr 1.1.4        v stringr 1.4.0
## v readr 2.1.1        v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(ggfortify)
library(readr)
library(stringi)
library(usethis)
```

EDA

```
#read csv
netflix <- read_csv("netflix_titles.csv")

## Rows: 8807 Columns: 12

## -- Column specification -----
## Delimiter: ","
## chr (11): show_id, type, title, director, cast, country, date_added, rating,...
## dbl (1): release_year

##
## 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.

#year added to netflix column added (ranges from 2014-2021)
netflix <- netflix %>%
  mutate(year_added = as.integer(str_sub(date_added, -4, -1)),
         release_year = as.integer(release_year),
         country = factor(country))
glimpse(netflix)
```

```
## Rows: 8,807
## Columns: 13
## $ show_id      <chr> "s1", "s2", "s3", "s4", "s5", "s6", "s7", "s8", "s9", "s1~
## $ type         <chr> "Movie", "TV Show", "TV Show", "TV Show", "TV Show", "TV ~
## $ title        <chr> "Dick Johnson Is Dead", "Blood & Water", "Ganglands", "Ja~
## $ director     <chr> "Kirsten Johnson", NA, "Julien Leclercq", NA, NA, "Mike F~
## $ cast         <chr> NA, "Ama Qamata, Khosi Ngema, Gail Mababane, Thabang Mola~
## $ country      <fct> "United States", "South Africa", NA, NA, "India", NA, NA,~
## $ date_added   <chr> "September 25, 2021", "September 24, 2021", "September 24~
## $ release_year <int> 2020, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 1993, 2021, 202~
## $ rating       <chr> "PG-13", "TV-MA", "TV-MA", "TV-MA", "TV-MA", "TV-MA", "PG~
## $ duration     <chr> "90 min", "2 Seasons", "1 Season", "1 Season", "2 Seasons~
## $ listed_in    <chr> "Documentaries", "International TV Shows, TV Dramas, TV M~
## $ description  <chr> "As her father nears the end of his life, filmmaker Kirst~
## $ year_added   <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 202~
```

```
#split data into shows and movies
```

```
shows <- netflix %>%
  filter(type == "TV Show") %>%
  mutate(num_seasons = as.integer(substring(duration,-5,1)),
         year_added = factor(year_added))
```

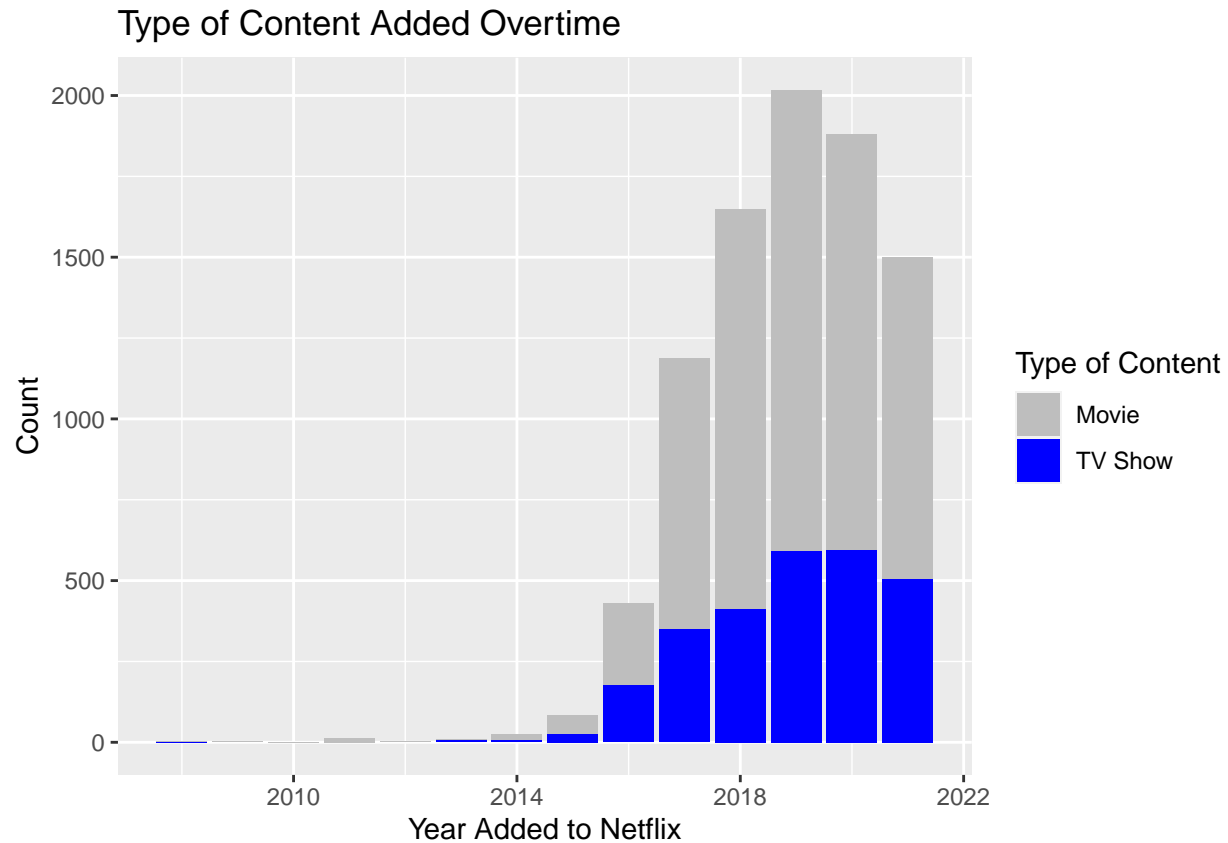
```
movies <- netflix %>%
  filter(type == "Movie",
         !is.na(duration)) %>%
  mutate(num_mins = as.integer(stri_sub(duration,1, -5)),
         year_added = factor(year_added))
```

```
#ggplot(data = netflix, aes(x = year_added, y = #duration))
```

```
#1 TV shows and movies overtime
```

```
ggplot(data = netflix, mapping = aes(x = year_added, fill = type)) + geom_bar() + scale_fill_manual(val
```

```
## Warning: Removed 10 rows containing non-finite values (stat_count).
```



```
#ggplot(netflix, aes(x = year_added, color = type)) + geom_density() + theme(axis.text = element_text(a
```

```
#2 countries overtime
```

```
#split countries (since some had multiple)
```

```
country_eda <- netflix %>%
  filter(!is.na(country)) %>%
  separate_rows(country, sep = ',')
country_eda$country <- trimws(country_eda$country)
```

```
country_eda %>%
  group_by(country) %>%
  count() %>%
  arrange(desc(n))
```

```
## # A tibble: 123 x 2
## # Groups:   country [123]
##   country      n
##   <chr>      <int>
## 1 United States 3690
## 2 India        1046
## 3 United Kingdom 806
## 4 Canada       445
## 5 France       393
## 6 Japan        318
## 7 Spain        232
```

```
## 8 South Korea      231
## 9 Germany          226
## 10 Mexico          169
## # ... with 113 more rows
```

```
#top 10 countries based on frequency of content
```

```
x <- list("United States", "India", "United Kingdom", "Canada", "France", "Japan", "Spain", "South Korea",
```

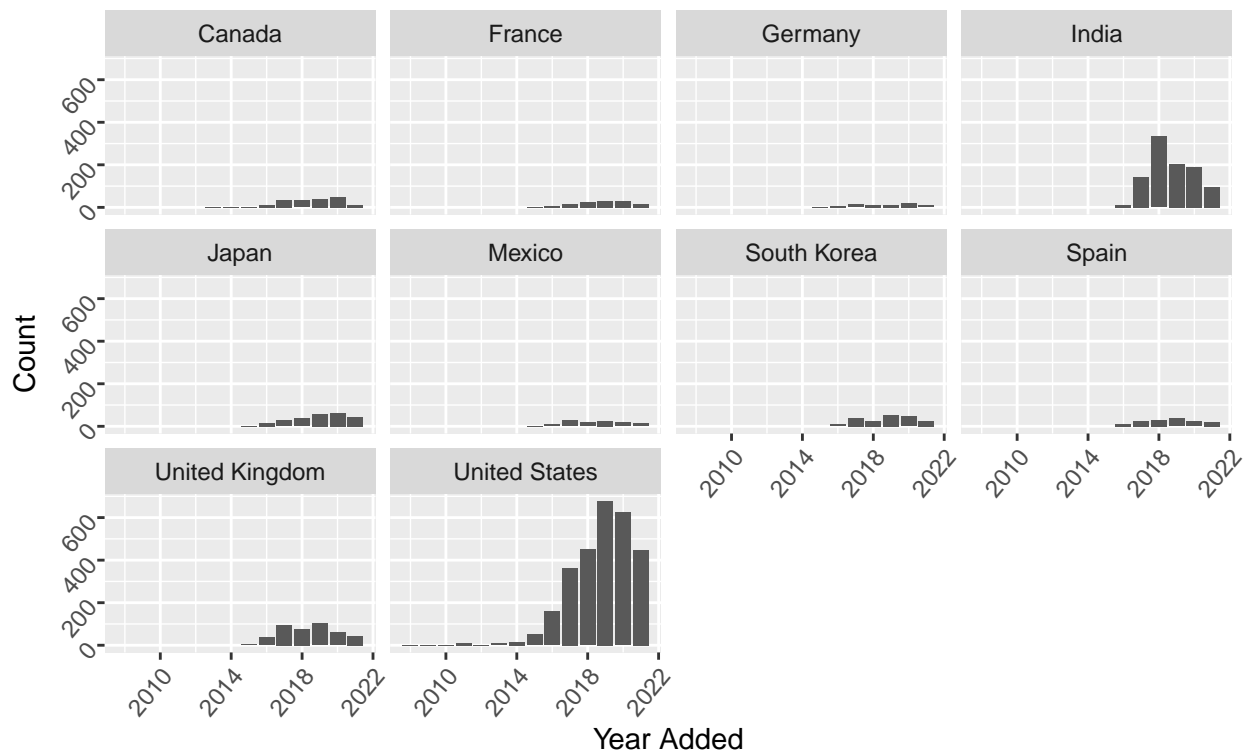
```
top_10_c <- netflix %>%
  filter(country %in% x,)
```

```
#overtime trends of content by country
```

```
ggplot(top_10_c, aes(x = year_added)) + geom_bar() + facet_wrap(~country) + theme(axis.text = element_text(size = 10))
```

```
## Warning: Removed 8 rows containing non-finite values (stat_count).
```

Distribution of Content Overtime by Country

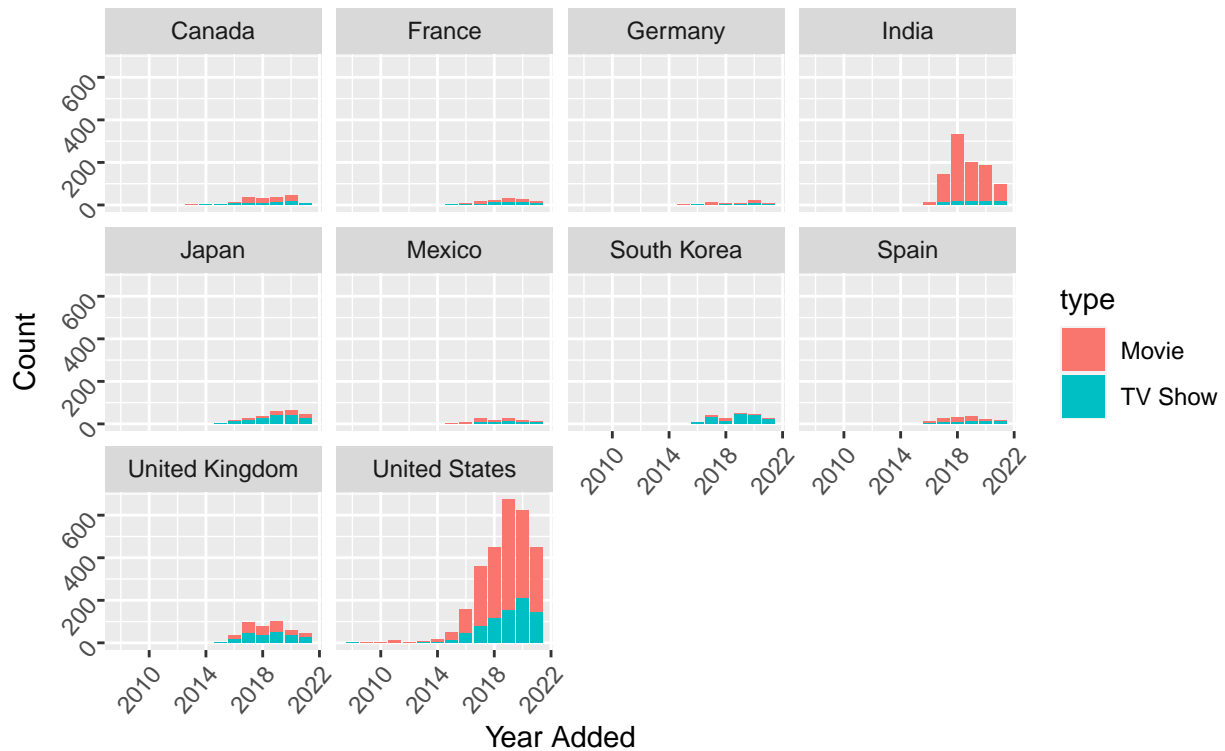


```
#overtime trends of movie/tv shows by country
```

```
ggplot(top_10_c, aes(x = year_added, fill = type)) + geom_bar() + facet_wrap(~country) + theme(axis.text = element_text(size = 10))
```

```
## Warning: Removed 8 rows containing non-finite values (stat_count).
```

Distribution of Content Overtime by Country



#3 genre overtime

```
netflix <- netflix %>%
  filter(!is.na(listed_in))
```

```
genres<-netflix%>%
  select(listed_in)%>%
  separate(listed_in, into = c('genre1','genre2','genre3'), ", ", convert = TRUE)
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 5078 rows [1, 4,
## 7, 9, 10, 13, 14, 16, 17, 19, 23, 24, 28, 29, 30, 32, 35, 38, 39, 40, ...].
```

```
genres<-genres%>%unlist()
```

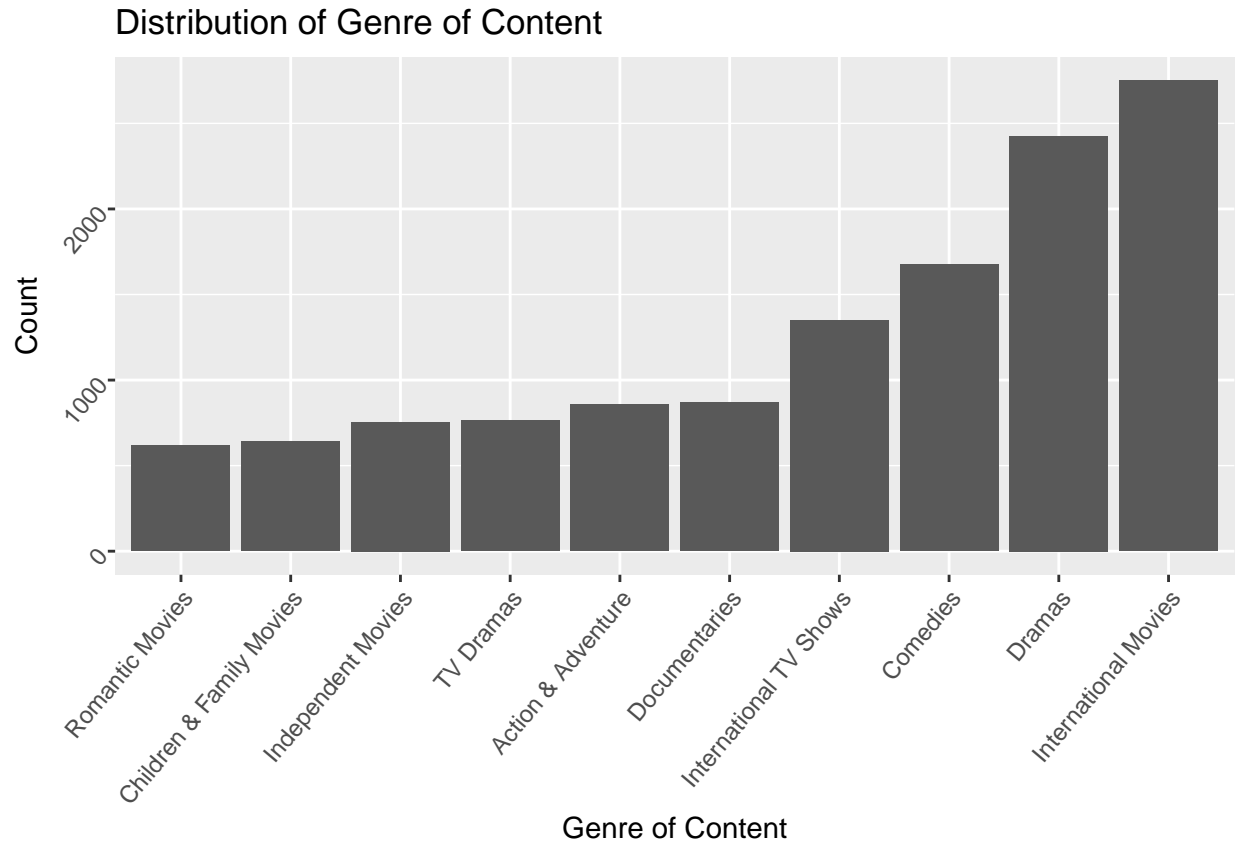
```
list_in<-tibble(
  list_in=genres)
```

```
genre_data <- list_in%>%
  group_by(list_in)%>%
  count()%>%
  filter(!is.na(list_in) && n>=600)
```

```
ggplot(genre_data, aes(n, reorder(list_in, fun=median, n)))+geom_histogram(stat = 'identity', show.legend=
  labs(
```

```
x='Count',
y='Genre of Content',
title='Distribution of Genre of Content') + coord_flip() + theme(axis.text = element_text(angle = 5
```

Warning: Ignoring unknown parameters: binwidth, bins, pad



```
genre_eda <- netflix %>%
  filter(!is.na(listed_in)) %>%
  separate_rows(listed_in, sep = ",")
genre_eda$listed_in <- trimws(genre_eda$listed_in)
```

```
genre_eda %>%
  group_by(listed_in) %>%
  count() %>%
  arrange(desc(n))
```

```
## # A tibble: 42 x 2
## # Groups:   listed_in [42]
##   listed_in          n
##   <chr>          <int>
## 1 International Movies 2752
## 2 Dramas             2427
## 3 Comedies           1674
## 4 International TV Shows 1351
## 5 Documentaries       869
## 6 Action & Adventure   859
```

```
## 7 TV Dramas 763
## 8 Independent Movies 756
## 9 Children & Family Movies 641
## 10 Romantic Movies 616
## # ... with 32 more rows
```

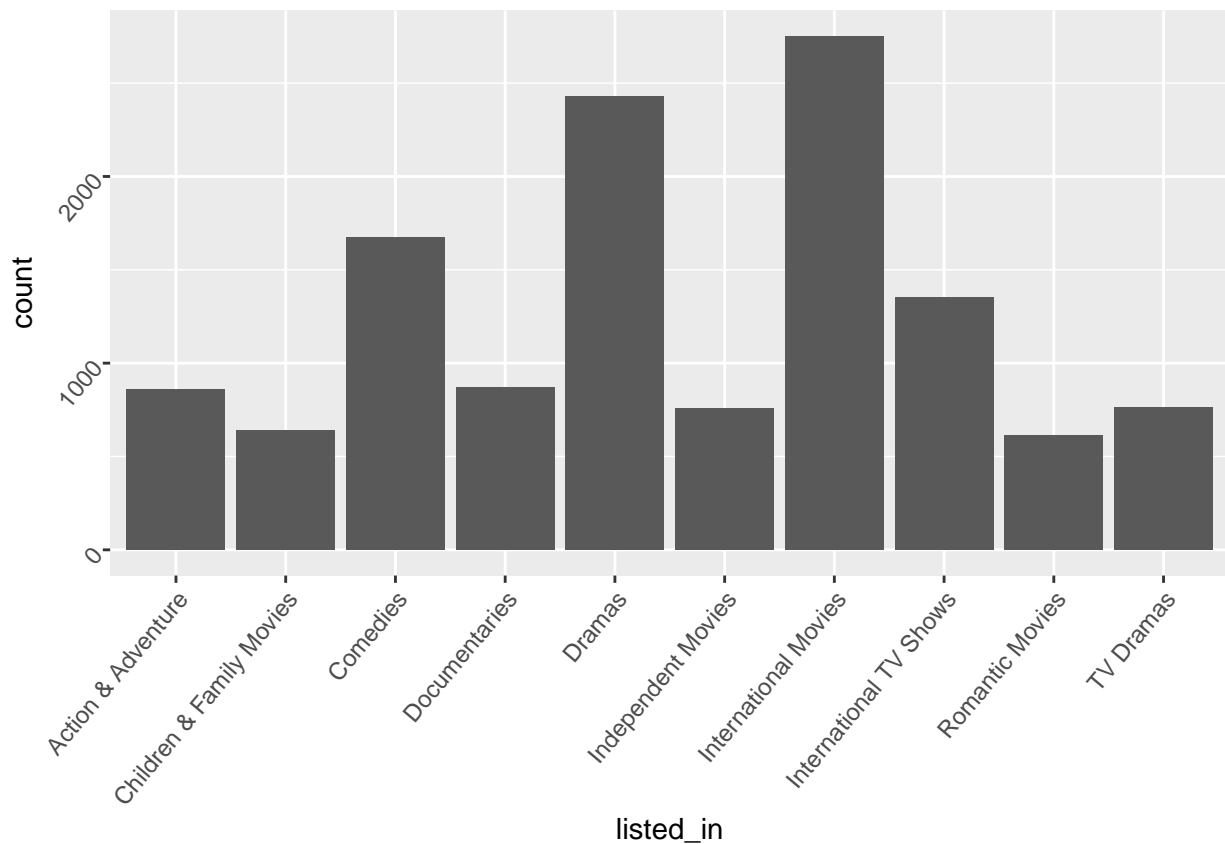
```
#top 10 genres
```

```
y <- list("International Movies", "Dramas", "Comedies", "International TV Shows", "Documentaries", "Act.
```

```
genre_eda %>%
```

```
  filter(listed_in %in% y) %>%
```

```
  ggplot(aes(x=listed_in)) + geom_bar() + theme(axis.text = element_text(angle = 50, hjust = 1))
```



```
genre_eda_movies <- movies %>%
  filter(!is.na(listed_in)) %>%
  separate_rows(listed_in, sep = ",")
genre_eda_movies$listed_in <- trimws(genre_eda_movies$listed_in)
```

```
genre_eda_movies %>%
  group_by(listed_in) %>%
  count() %>%
  arrange(desc(n))
```

```
## # A tibble: 20 x 2
## # Groups:   listed_in [20]
##   listed_in      n
##   <chr>      <int>
```

```
## 1 International Movies      2752
## 2 Dramas                   2427
## 3 Comedies                  1674
## 4 Documentaries             869
## 5 Action & Adventure        859
## 6 Independent Movies        756
## 7 Children & Family Movies   641
## 8 Romantic Movies           616
## 9 Thrillers                  577
## 10 Music & Musicals          375
## 11 Horror Movies             357
## 12 Stand-Up Comedy           343
## 13 Sci-Fi & Fantasy          243
## 14 Sports Movies             219
## 15 Classic Movies            116
## 16 LGBTQ Movies              102
## 17 Anime Features            71
## 18 Cult Movies                71
## 19 Faith & Spirituality       65
## 20 Movies                    54
```

```
#top 10 movie genres
```

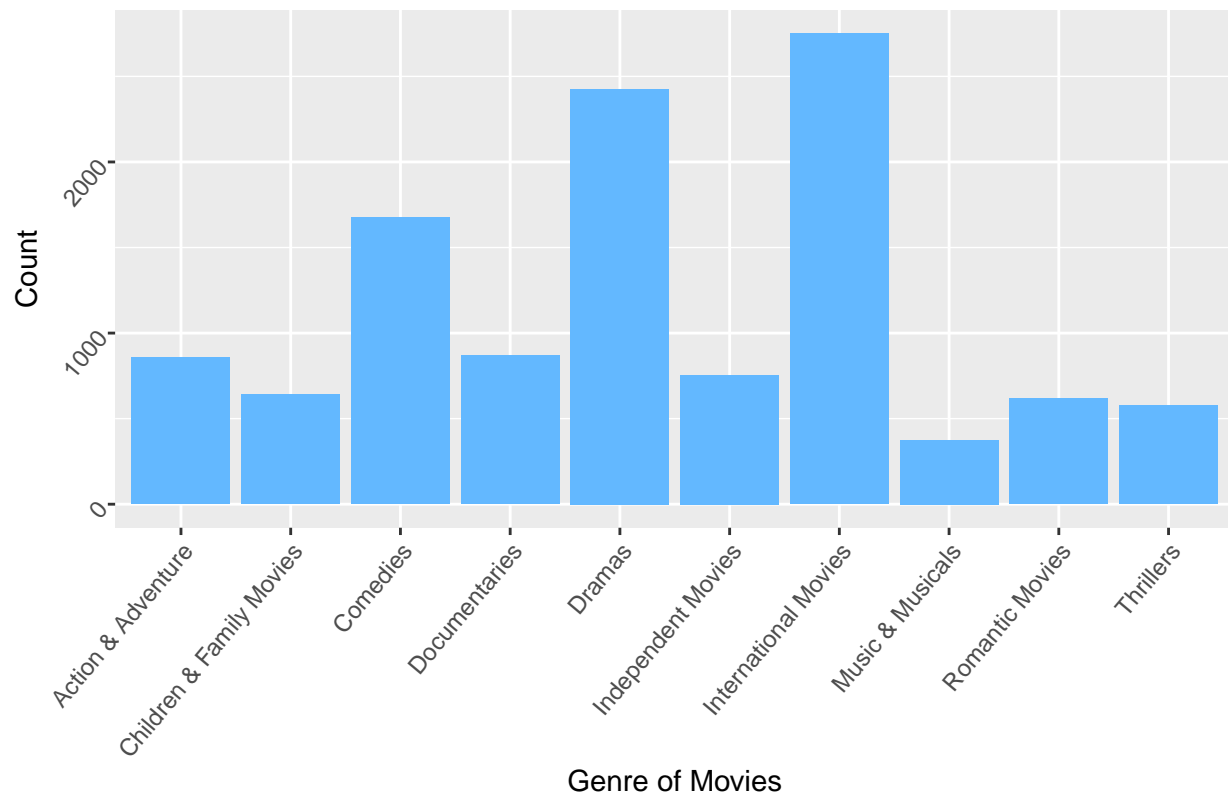
```
y1 <-list("International Movies", "Dramas", "Comedies", "Documentaries", "Action & Adventure", "Independent Movies", "Children & Family Movies", "Romantic Movies", "Thrillers", "Music & Musicals")
```

```
genre_eda_movies %>%
```

```
  filter(listed_in %in% y1) %>%
```

```
  ggplot(aes(x=listed_in)) + geom_bar(fill = "steelblue1") + theme(axis.text = element_text(angle = 50, l
```

Distribution of Top 10 Movie Genres




```
genre_eda_shows <- shows %>%
  filter(!is.na(listed_in)) %>%
  separate_rows(listed_in, sep = ",")
genre_eda_shows$listed_in <- trimws(genre_eda_shows$listed_in)
```

```
genre_eda_shows %>%
  group_by(listed_in) %>%
  count() %>%
  arrange(desc(n))
```

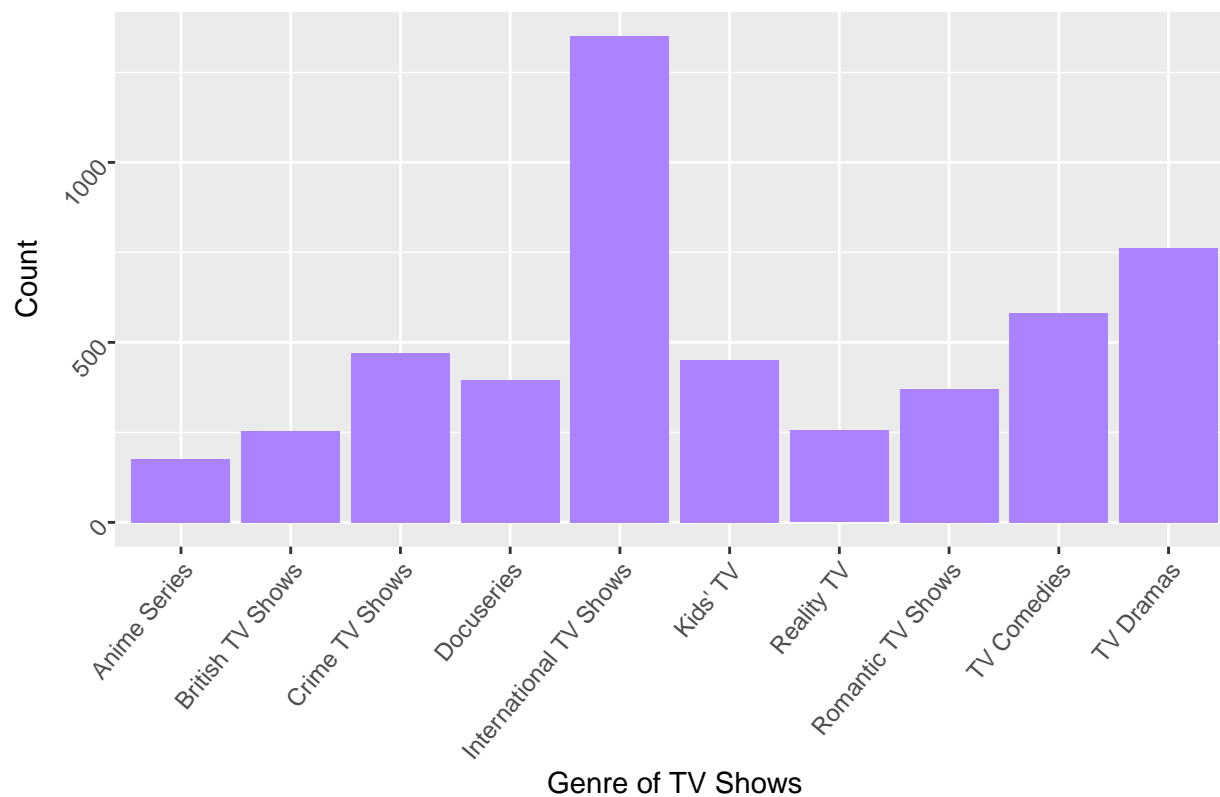
```
## # A tibble: 22 x 2
## # Groups:   listed_in [22]
##   listed_in          n
##   <chr>          <int>
## 1 International TV Shows 1351
## 2 TV Dramas          763
## 3 TV Comedies         581
## 4 Crime TV Shows      470
## 5 Kids' TV           451
## 6 Docuseries          395
## 7 Romantic TV Shows   370
## 8 Reality TV          255
## 9 British TV Shows    253
## 10 Anime Series       176
## # ... with 12 more rows
```

```
#top 10 show genres
```

```
y2 <- list("International TV Shows", "TV Dramas", "TV Comedies", "Crime TV Shows", "Kids' TV", "Docuseries")
```

```
genre_eda_shows %>%
  filter(listed_in %in% y2) %>%
  ggplot(aes(x = listed_in)) +geom_bar(fill = "mediumpurple1") +theme(axis.text = element_text(angle = 90))
```

Distribution of Top 10 TV Shows Genres



```
US_genre <- country_eda %>%
  filter(country == "United States",
    !is.na(listed_in)) %>%
  separate_rows(listed_in, sep = ",")
US_genre$listed_in <- trimws(US_genre$listed_in)
```

```
US_movie_genre <- US_genre %>%
  filter(type == "Movie")
```

```
US_movie_genre %>%
  group_by(listed_in) %>%
  count() %>%
  arrange(desc(n))
```

```
## # A tibble: 20 x 2
## # Groups:   listed_in [20]
##   listed_in          n
##   <chr>          <int>
## 1 Dramas          835
## 2 Comedies         680
## 3 Documentaries    512
## 4 Action & Adventure 404
## 5 Children & Family Movies 390
## 6 Independent Movies 390
```

```
## 7 Thrillers 292
## 8 Romantic Movies 225
## 9 Stand-Up Comedy 216
## 10 Horror Movies 201
## 11 Sci-Fi & Fantasy 181
## 12 International Movies 166
## 13 Music & Musicals 147
## 14 Sports Movies 113
## 15 Classic Movies 81
## 16 LGBTQ Movies 63
## 17 Cult Movies 52
## 18 Faith & Spirituality 42
## 19 Movies 22
## 20 Anime Features 7
```

```
#top 10 movie genres in the U.S.
```

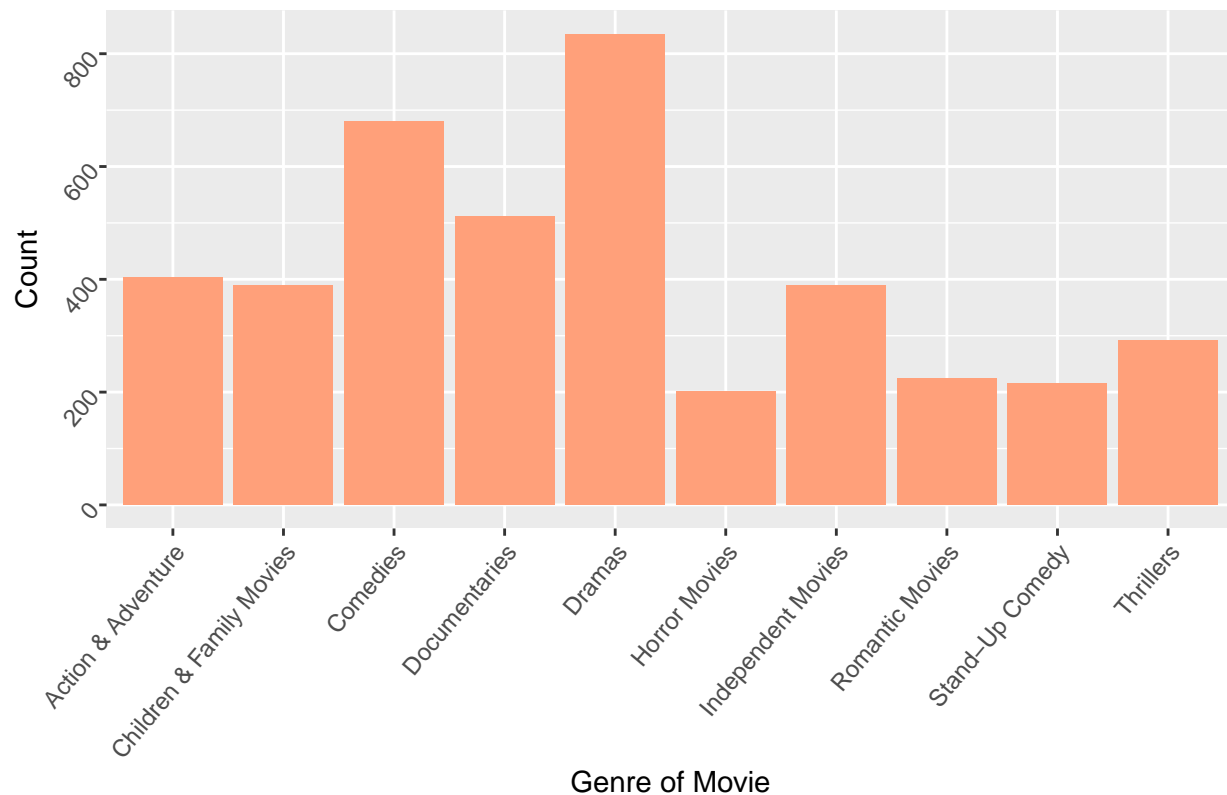
```
z <- list("Dramas", "Comedies", "Documentaries", "Action & Adventure", "Children & Family Movies", "Ind
```

```
US_movie_genre %>%
```

```
  filter(listed_in %in% z) %>%
```

```
  ggplot(aes(x = listed_in)) +geom_bar(fill = "lightsalmon") +theme(axis.text = element_text(angle = 50
```

Distribution of Top 10 Movie Genres in the U.S.



```
US_show_genre <- US_genre %>%
  filter(type == "TV Show")
```

```
US_show_genre %>%
```

```
  group_by(listed_in) %>%
```

```
count() %>%
arrange(desc(n))
```

```
## # A tibble: 22 x 2
## # Groups:   listed_in [22]
##   listed_in      n
##   <chr>        <int>
## 1 TV Comedies    258
## 2 TV Dramas     232
## 3 Kids' TV      214
## 4 Docuseries    192
## 5 Crime TV Shows 145
## 6 Reality TV    123
## 7 TV Action & Adventure 94
## 8 International TV Shows 74
## 9 TV Sci-Fi & Fantasy 60
## 10 TV Mysteries 51
## # ... with 12 more rows
```

#top 10 show genres in the U.S.

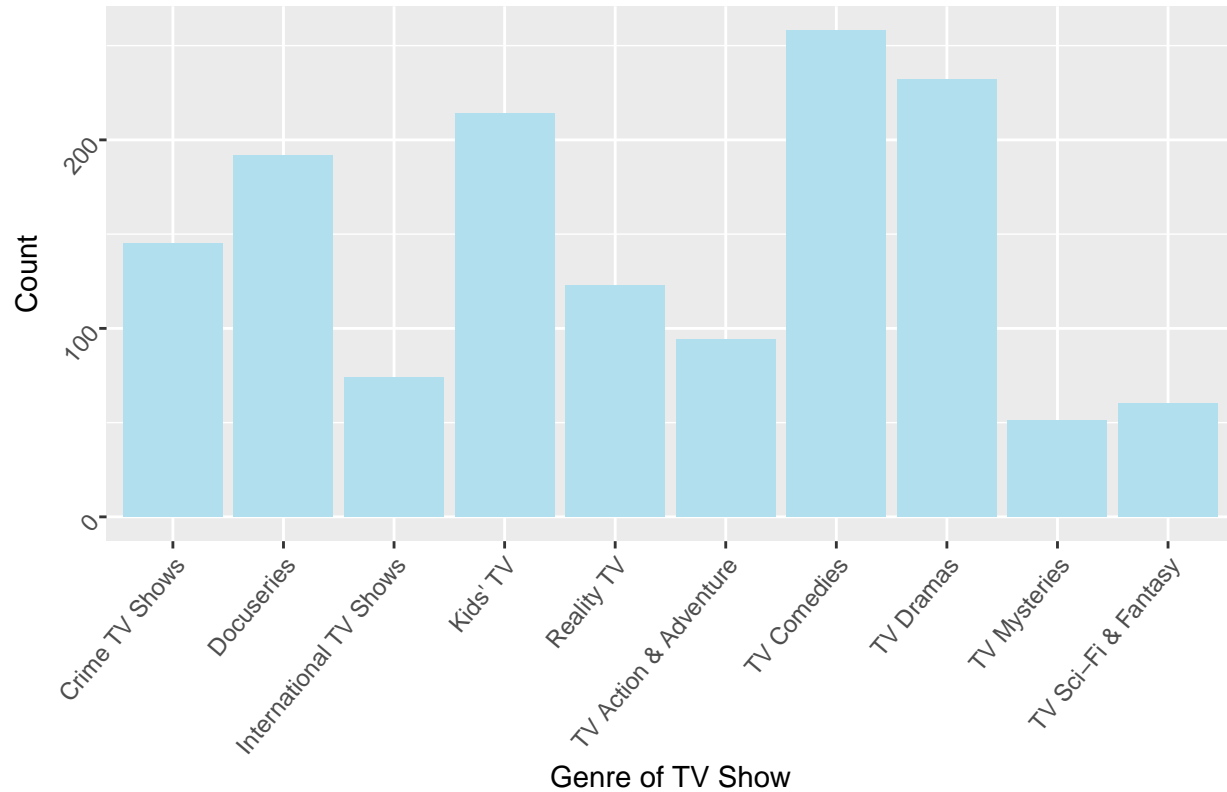
```
z1<- list("TV Comedies", "TV Dramas", "Kids' TV", "Docuseries", "Crime TV Shows", "Reality TV", "TV Act.
```

```
US_show_genre %>%
```

```
  filter(listed_in %in% z1) %>%
```

```
  ggplot(aes(x = listed_in)) +geom_bar(fill = "lightblue2") +theme(axis.text = element_text(angle = 50,
```

Distribution of Top 10 Show Genres in the U.S.



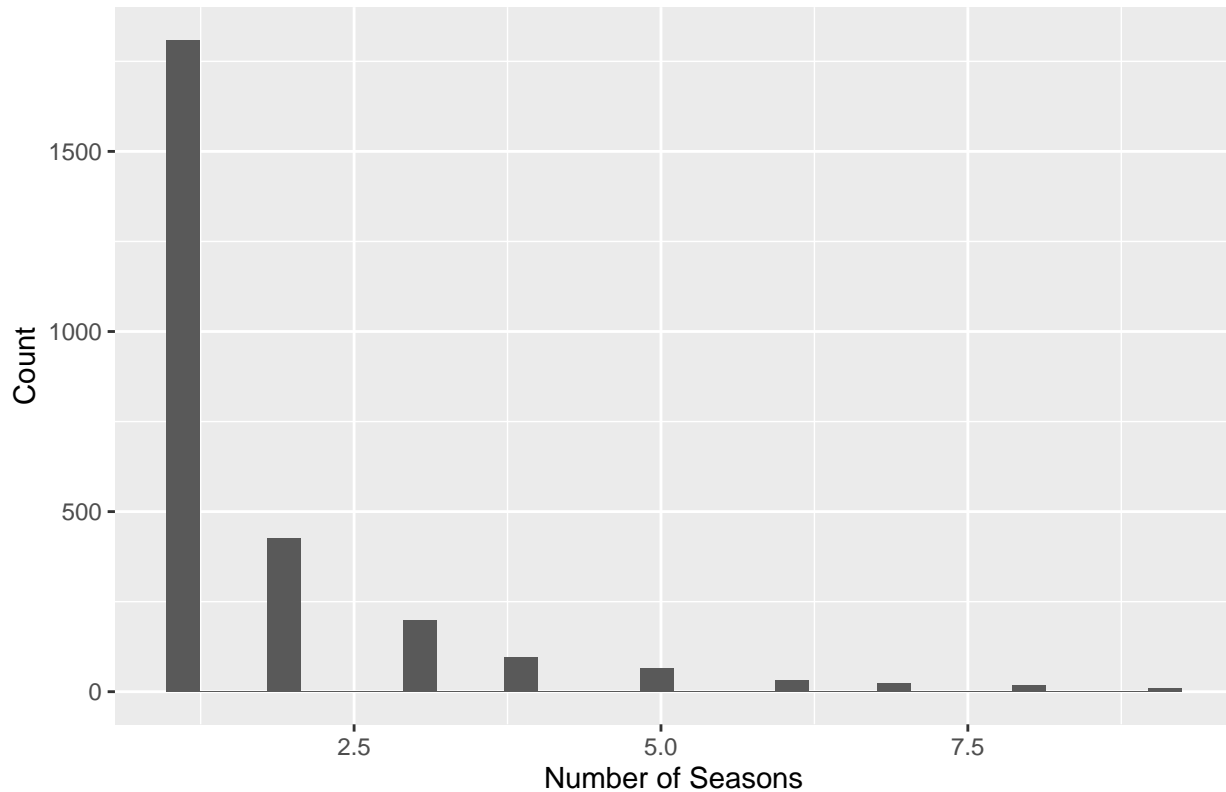
```
#4 duration overtime (shows then movies)
```

```
movies <- movies %>%  
  group_by(year_added) %>%  
  mutate(mean_dur = mean(num_mins),  
         sd_dur = sd(num_mins))
```

```
ggplot(data = shows, mapping = aes(x = num_seasons)) + geom_histogram() + labs(title = "Distribution of
```

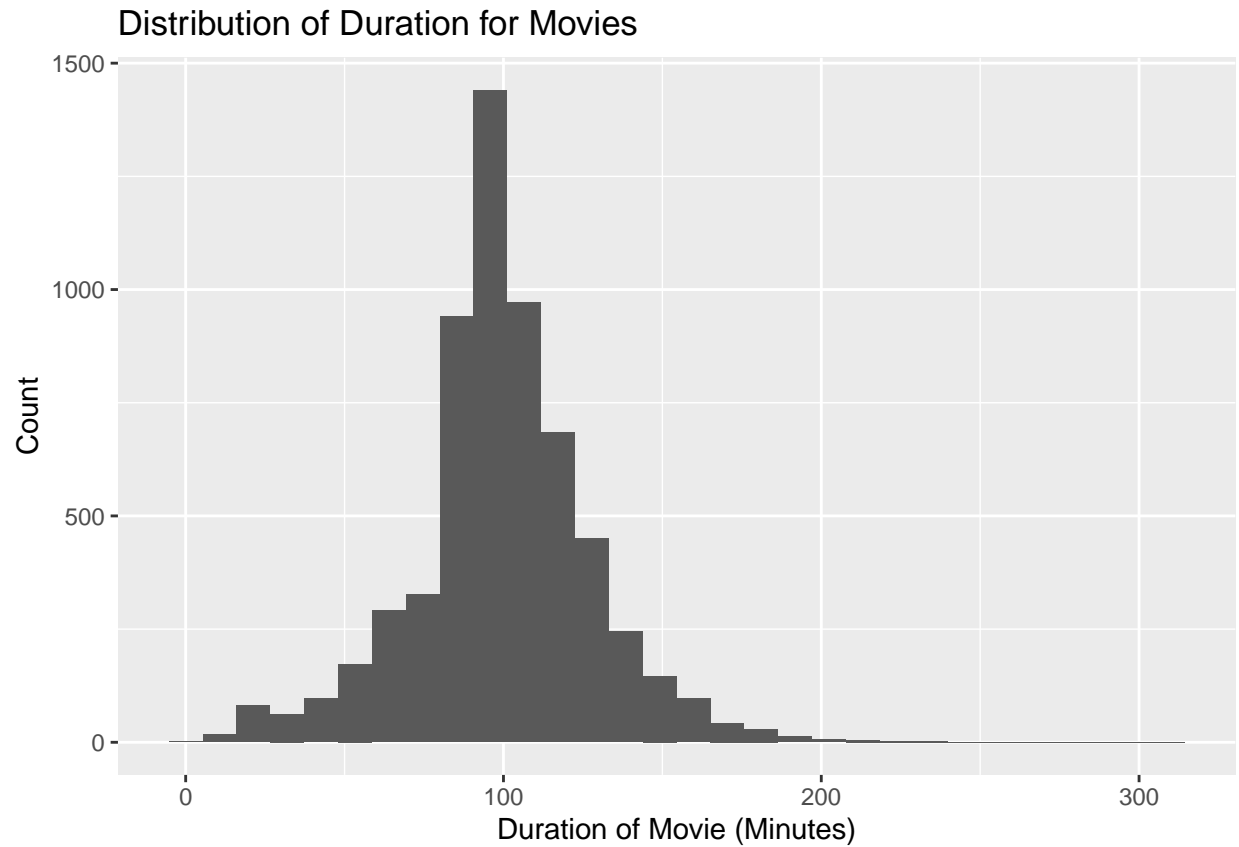
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Distribution of Duration for TV Shows

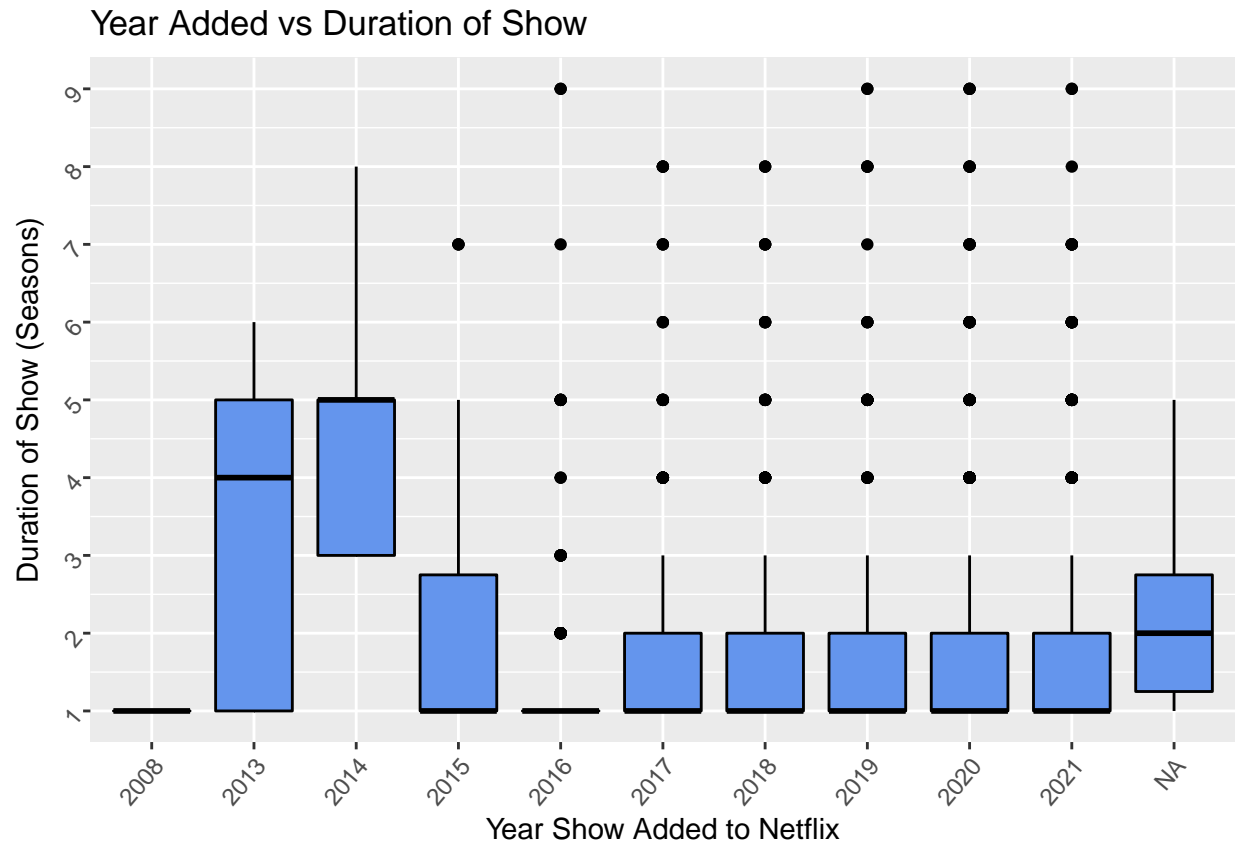


```
ggplot(data = movies, mapping = aes(x = num_mins)) + geom_histogram() + labs(title = "Distribution of
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



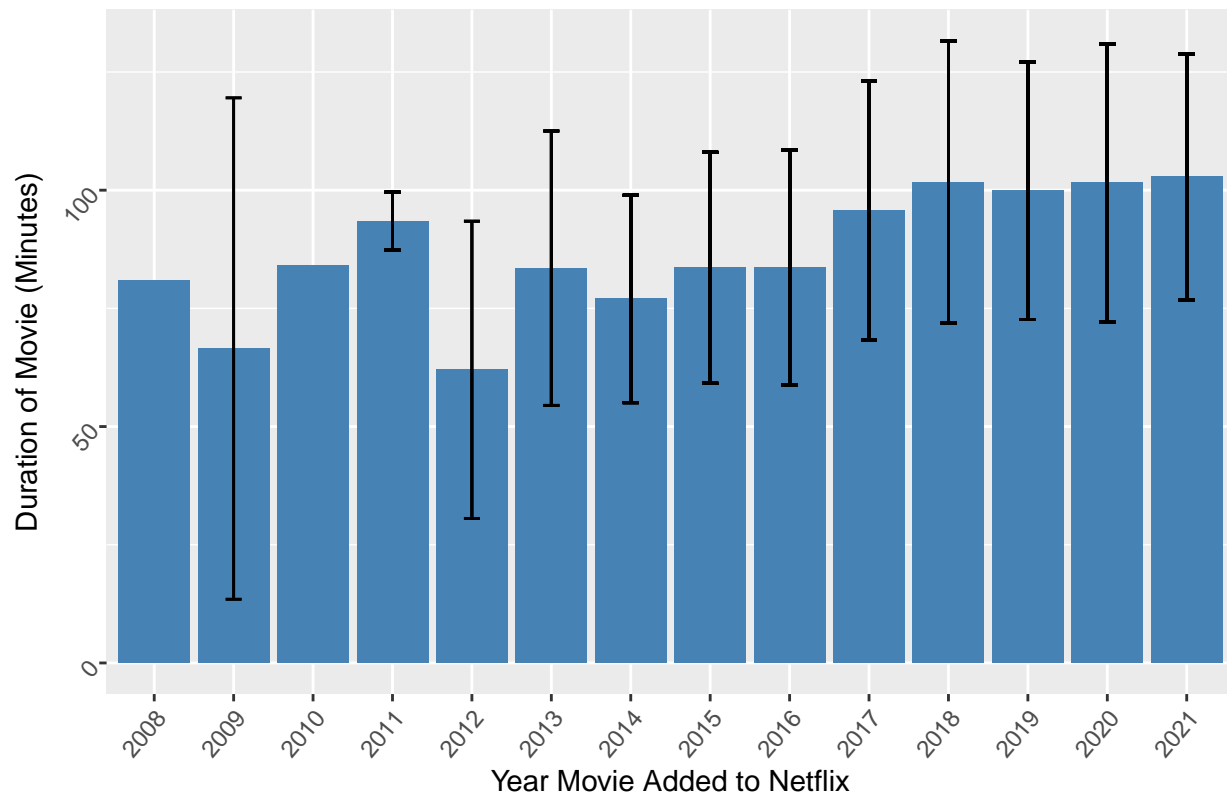
#because the distribution of shows is skewed we should use a box plot because it showcases the median d
`ggplot(data = shows, mapping = aes(x = year_added, y = num_seasons)) + geom_boxplot(color = "black", fil`



#because the distribution of movies is relatively normal we can use a histogram of the mean duration of

```
ggplot(movies, aes(x=year_added, y=mean_dur)) +
  geom_bar(position=position_dodge(), stat="identity",
    fill="steelblue") +
  geom_errorbar(aes(ymin=mean_dur-sd_dur, ymax=mean_dur+sd_dur), width=.2) + labs(title = "Year Added vs Duration of Show")
```

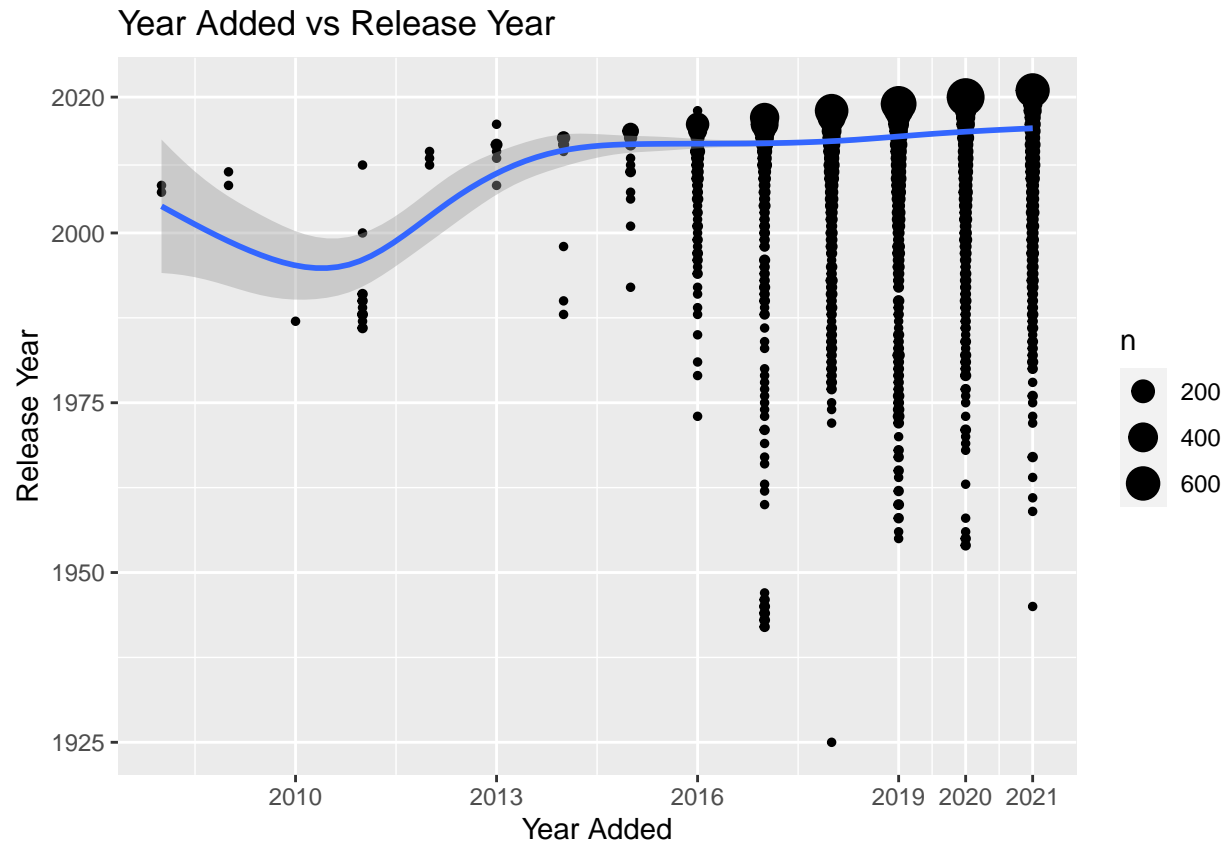
Year Added vs Duration of Movie



```
#5 release year
releaseyr_eda <- netflix %>%
  filter(!is.na(release_year),
         !is.na(year_added))

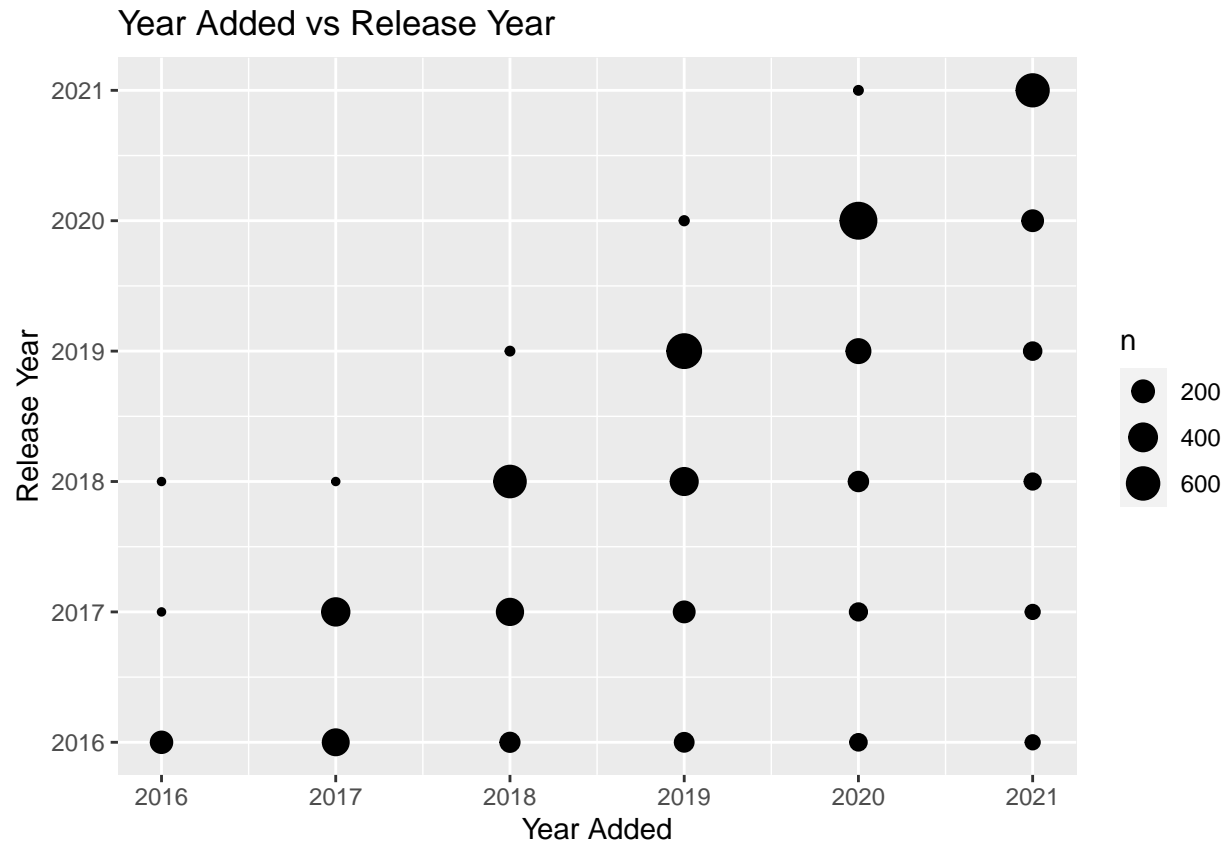
ggplot(releaseyr_eda, mapping = aes(x = year_added, y = release_year)) + geom_count() + geom_smooth() + s

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
ggplot(releaseyr_eda, mapping = aes(x = year_added, y = release_year)) + geom_count() + geom_smooth() + x
```

```
## Warning: Removed 3144 rows containing non-finite values (stat_sum).
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 3144 rows containing non-finite values (stat_smooth).
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```



```
#6 directors overtime

#top 10 directors in all content

director_eda <- netflix %>%
  filter(!is.na(director)) %>%
  separate_rows(director, sep = ",")
director_eda$director <- trimws(director_eda$director)
```

```
director_eda %>%
  group_by(director) %>%
  count() %>%
  arrange(desc(n))
```

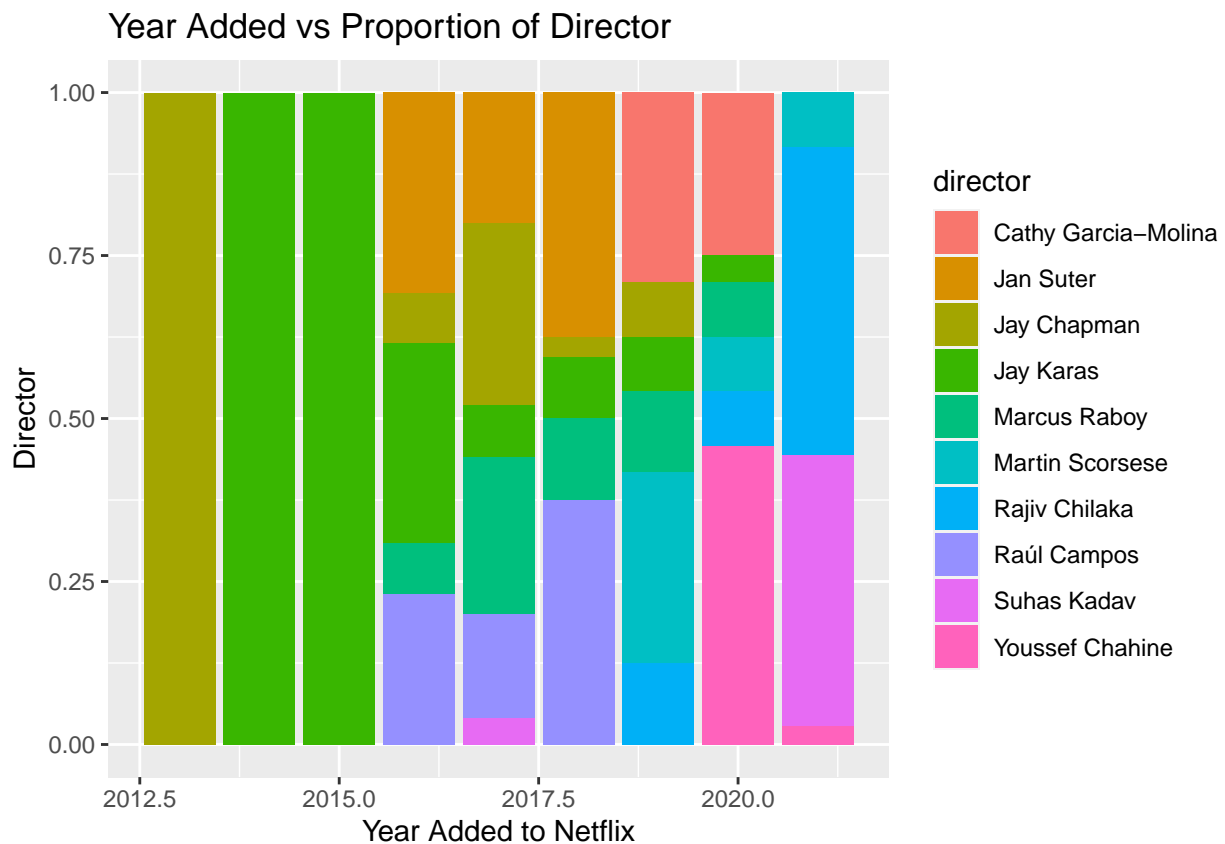
```
## # A tibble: 4,993 x 2
## # Groups:   director [4,993]
##   director      n
##   <chr>      <int>
## 1 Rajiv Chilaka    22
## 2 Jan Suter        21
## 3 Raúl Campos     19
## 4 Marcus Raboy     16
## 5 Suhas Kadav      16
## 6 Jay Karas        15
```

```
## 7 Cathy Garcia-Molina      13
## 8 Jay Chapman               12
## 9 Martin Scorsese          12
## 10 Youssef Chahine          12
## # ... with 4,983 more rows

x2 <- list("Rajiv Chilaka", "Jan Suter", "Raúl Campos", "Marcus Raboy", "Suhas Kadav", "Jay Karas", "Cathy Garcia-Molina")

top_10_d <- director_eda %>%
  filter(director %in% x2)

ggplot(data = top_10_d, mapping = aes(x = year_added, fill = director)) + geom_bar(position = "fill") +
```

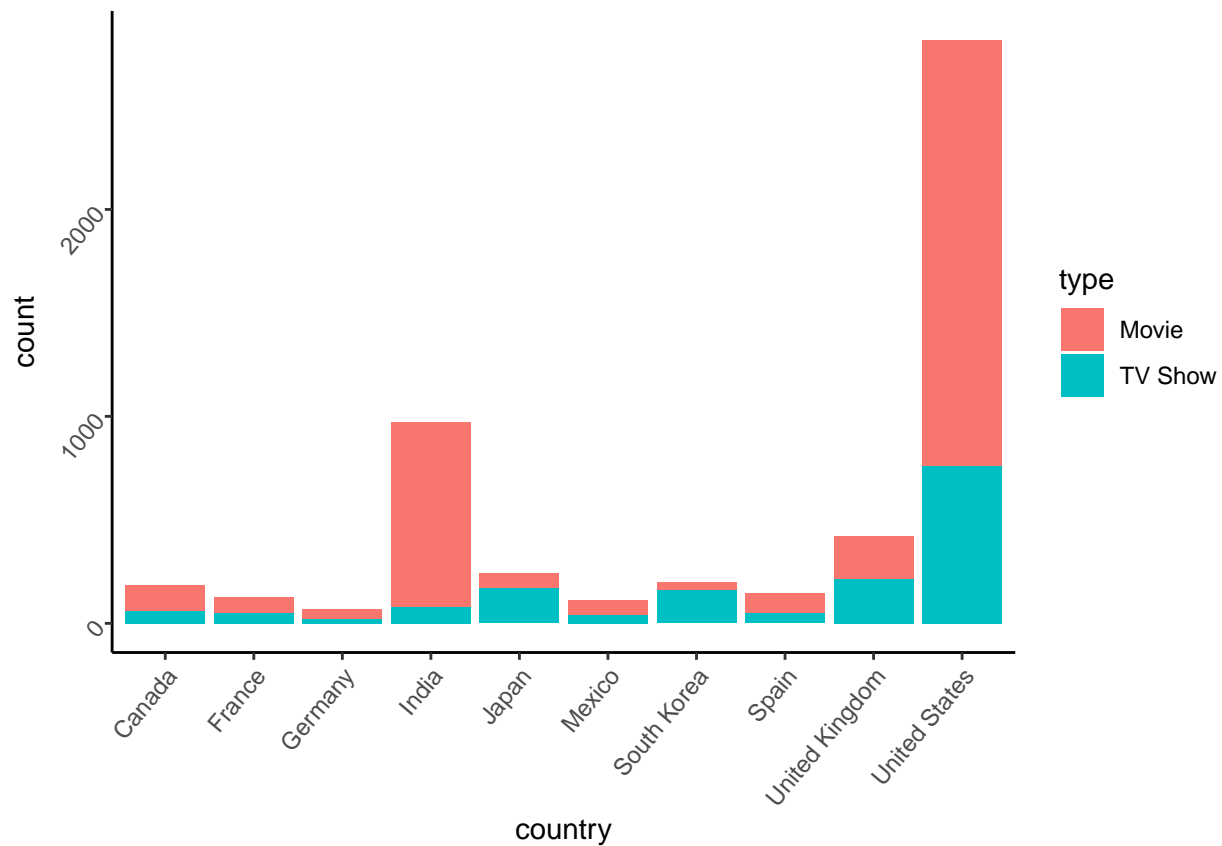


```
#6 actors overtime
#HAVE TO FIGURE OUT HOW TO SPLIT UP CAST BY ACTORS!!

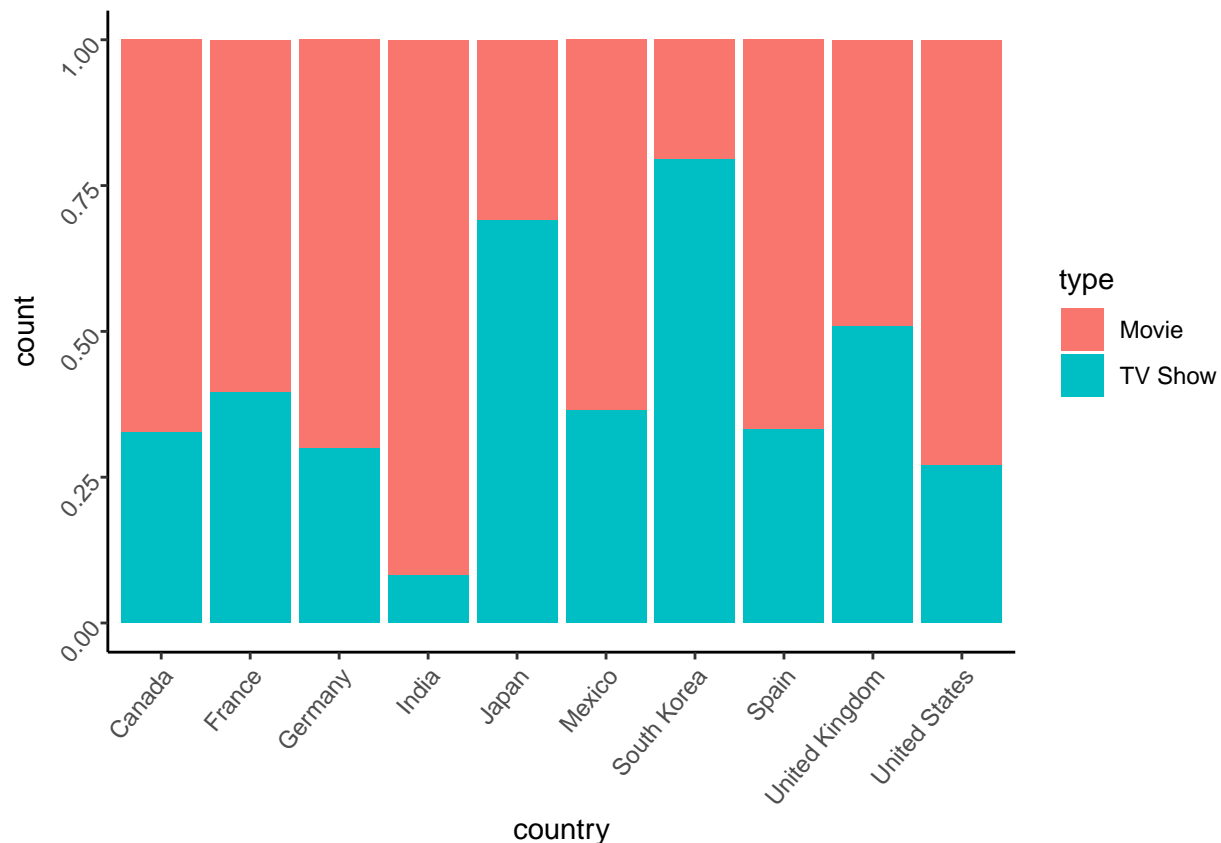
#create model that shows which attributes of a show lead to more seasons!
# figure out how to split up countries to use to create model that shows which aspects of a movie leads

###
#both_country_count <- netflix %>%
# filter(!is.na(country)) %>%
# group_by(country) %>%
# count() %>%
# arrange(desc(n))
#glimpse(both_country_count)
```

```
#bar plot on the count of movies and shows within top 10 countries
ggplot(top_10_c, aes(x = country, fill = type)) +
  geom_bar() +
  theme_classic() + theme(axis.text = element_text(angle = 50, hjust = 1))
```



```
#bar plot on the proportions of movies and shows within top 10 countries
ggplot(top_10_c, aes(x = country, fill = type)) +
  geom_bar(position = "fill") +
  theme_classic() + theme(axis.text = element_text(angle = 50, hjust = 1))
```



```
#ggplot(data.frame(both_country_count), aes(x=country)) +
#geom_bar()
```

```
#split data into shows and movies
shows <- netflix %>%
  filter(type == "TV Show") %>%
  mutate(num_seasons = as.integer(substring(duration,-5,1)))
```

```
movies <- netflix %>%
  filter(type == "Movie")
glimpse(shows)
```

```
## Rows: 2,676
## Columns: 14
## $ show_id      <chr> "s2", "s3", "s4", "s5", "s6", "s9", "s11", "s12", "s15", ~
## $ type         <chr> "TV Show", "TV Show", "TV Show", "TV Show", "TV Show", "T~
## $ title        <chr> "Blood & Water", "Ganglands", "Jailbirds New Orleans", "K~
## $ director     <chr> NA, "Julien Leclercq", NA, NA, "Mike Flanagan", "Andy Dev~
## $ cast         <chr> "Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, ~
## $ country      <fct> "South Africa", NA, NA, "India", NA, "United Kingdom", NA~
## $ date_added   <chr> "September 24, 2021", "September 24, 2021", "September 24~
## $ release_year <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 202~
## $ rating       <chr> "TV-MA", "TV-MA", "TV-MA", "TV-MA", "TV-MA", "TV-14", "TV~
## $ duration     <chr> "2 Seasons", "1 Season", "1 Season", "2 Seasons", "1 Seas~
## $ listed_in    <chr> "International TV Shows, TV Dramas, TV Mysteries", "Crime~
## $ description  <chr> "After crossing paths at a party, a Cape Town teen sets o~
```

```
## $ year_added    <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 202~
## $ num_seasons   <int> 2, 1, 1, 2, 1, 9, 1, 1, 1, 4, 2, 1, 1, 5, 2, 1, 3, 1, 1, ~
```

```
glimpse(movies)
```

```
## Rows: 6,131
## Columns: 13
## $ show_id       <chr> "s1", "s7", "s8", "s10", "s13", "s14", "s17", "s19", "s23~
## $ type          <chr> "Movie", "Movie", "Movie", "Movie", "Movie", "Movie", "Movie", "Mo~
## $ title         <chr> "Dick Johnson Is Dead", "My Little Pony: A New Generation~
## $ director      <chr> "Kirsten Johnson", "Robert Cullen, José Luis Ucha", "Hail~
## $ cast          <chr> NA, "Vanessa Hudgens, Kimiko Glenn, James Marsden, Sofia ~
## $ country       <fct> "United States", NA, "United States, Ghana, Burkina Faso,~
## $ date_added    <chr> "September 25, 2021", "September 24, 2021", "September 24~
## $ release_year  <int> 2020, 2021, 1993, 2021, 2021, 2021, 2020, 2021, 1996, 202~
## $ rating        <chr> "PG-13", "PG", "TV-MA", "PG-13", "TV-MA", "TV-PG", "TV-MA~
## $ duration      <chr> "90 min", "91 min", "125 min", "104 min", "127 min", "91 ~
## $ listed_in     <chr> "Documentaries", "Children & Family Movies", "Dramas, Ind~
## $ description   <chr> "As her father nears the end of his life, filmmaker Kirst~
## $ year_added    <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 202~
```

```
#all data entries are <chr>"
```

Possible Useful Variables

Shows: - country - date_added - release_year - duration (# of seasons) *dependent (measure of success) - rating - listed_in (genre) Most popular: - Drama - Documentary - Comedy

*most shows did not have a director listed

Movies: - director - country *dependent (measure of success) - date_added - release_year - rating - duration (# of minutes) - listed_in (genre) Most Popular: - Comedy - Animation - Drama

<https://www.whats-on-netflix.com/news/what-movie-tv-genres-perform-well-in-the-netflix-top-10s/>

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

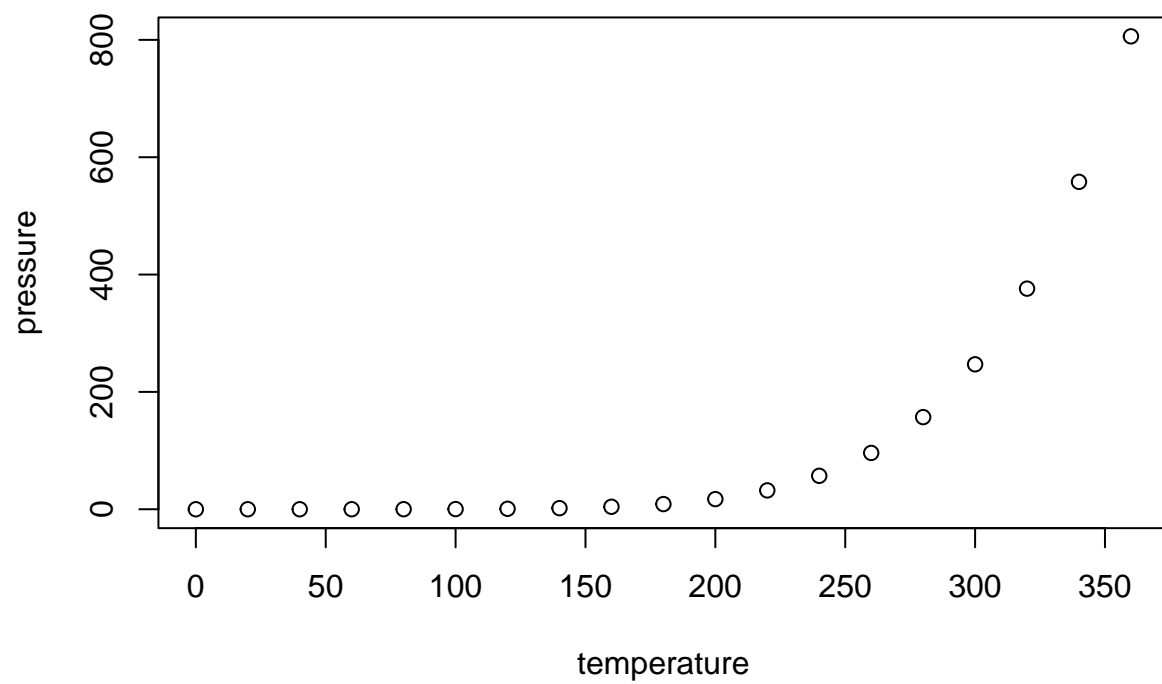
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed          dist
##  Min.   : 4.0      Min.   :  2.00
## 1st Qu.:12.0      1st Qu.: 26.00
##  Median :15.0      Median : 36.00
##   Mean  :15.4      Mean   : 42.98
## 3rd Qu.:19.0      3rd Qu.: 56.00
##   Max.  :25.0      Max.    :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.