

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)
# add any other packages if needed
```

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)
```

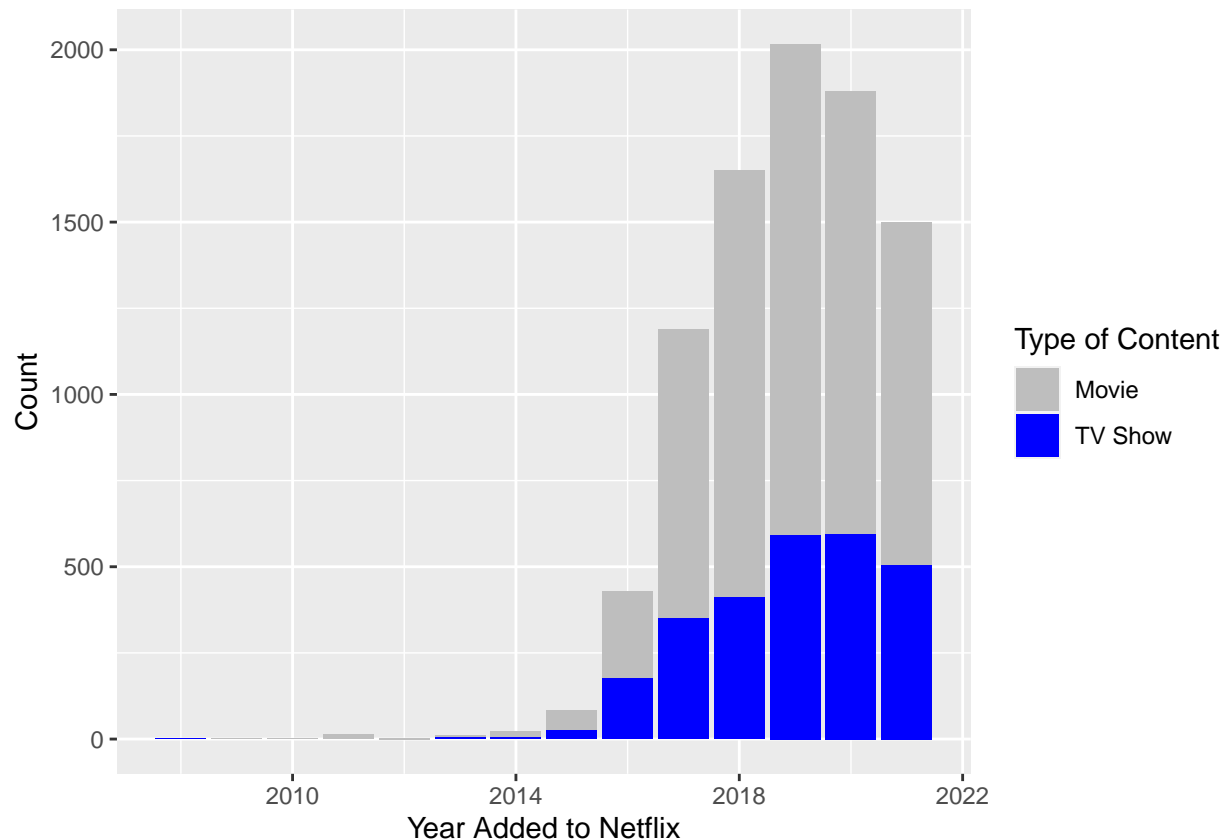
```
## Rows: 8,807
## Columns: 13
## $ show_id      <chr> "s1", "s2", "s3", "s4", "s5", "s6", "s7", "s8", "s9", "s1~
## $ type         <chr> "Movie", "TV Show", "TV Show", "TV Show", "TV Show", "TV ~
## $ title        <chr> "Dick Johnson Is Dead", "Blood & Water", "Ganglands", "Ja~
## $ director     <chr> "Kirsten Johnson", NA, "Julien Leclercq", NA, NA, "Mike F~
## $ cast         <chr> NA, "Ama Qamata, Khosi Ngema, Gail Mabalone, 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, 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~
```

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

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

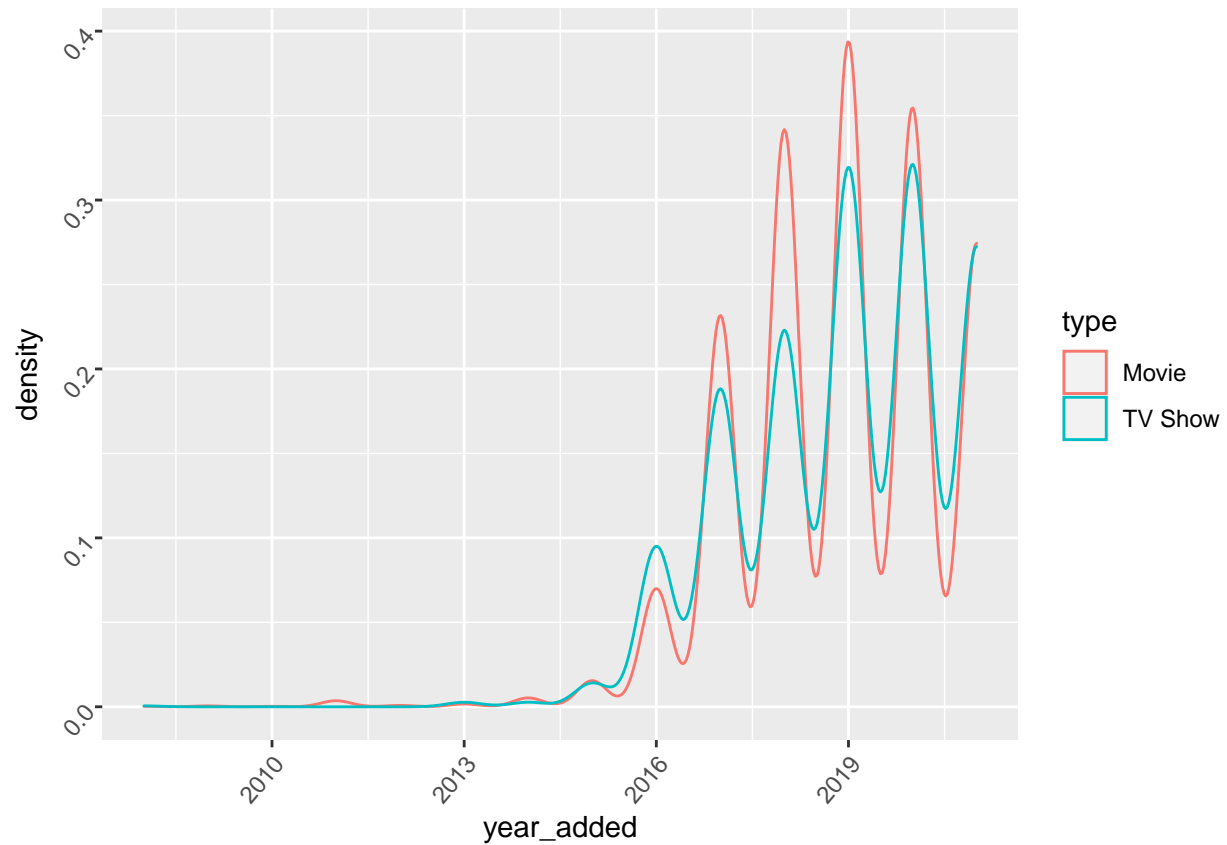
```
ggplot(data = netflix, mapping = aes(x = year_added, fill = type)) + geom_bar() + scale_fill_manual(val
```

```
## 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(ang
```

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



```
#2 countries overtime
```

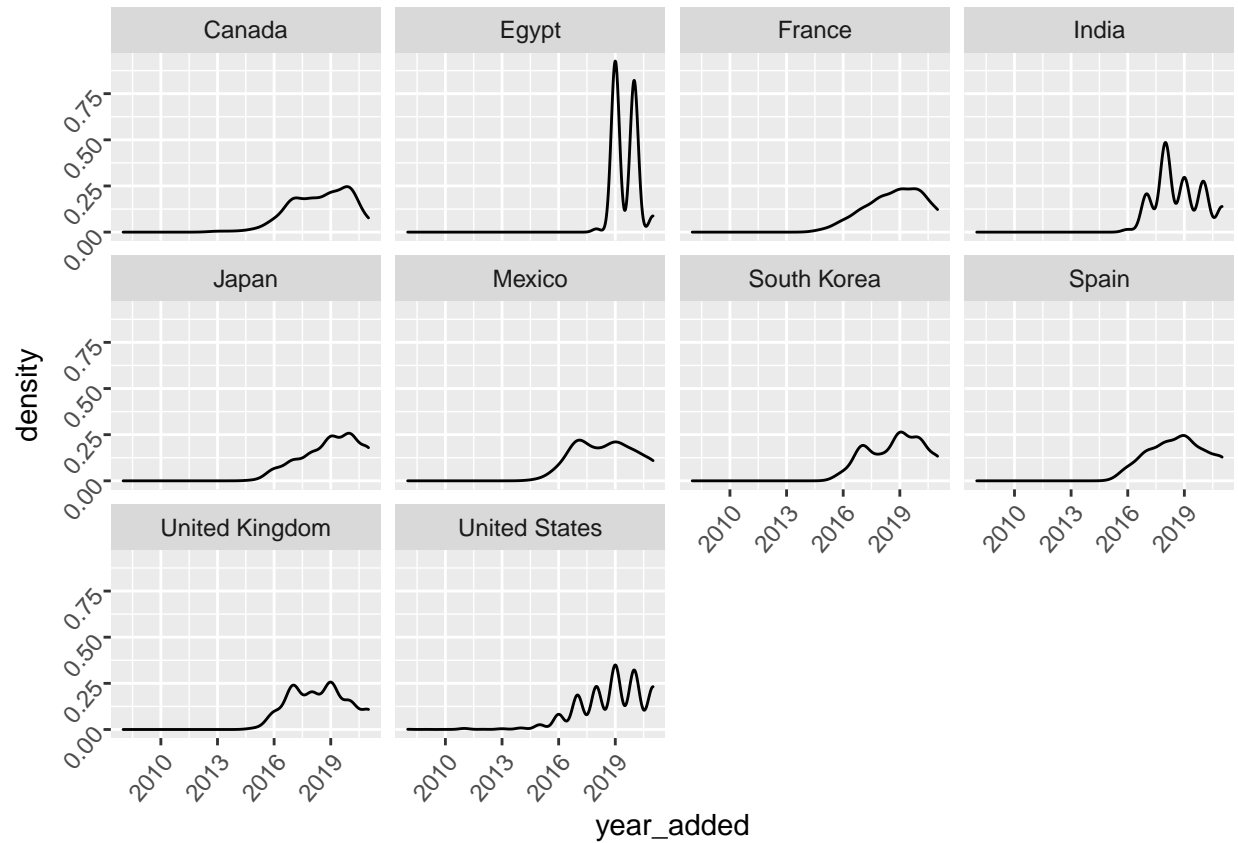
```
#top 10 countries based on frequency of content
```

```
x <- list("United States", "India", "United Kingdom", "Japan", "South Korea", "Canada", "Spain", "France")
top_10_c <- netflix %>%
  filter(country %in% x)
```

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

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

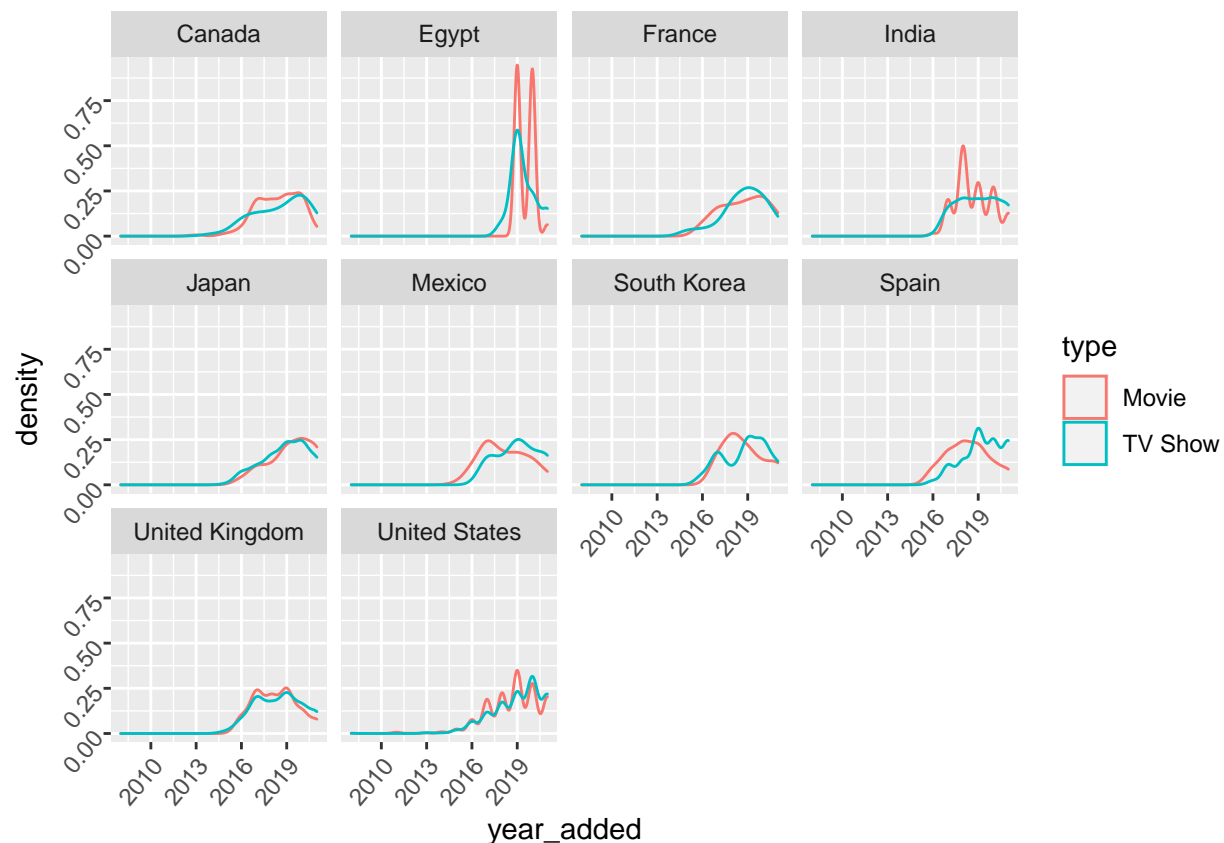
```
## Warning: Removed 8 rows containing non-finite values (stat_density).
```



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

```
ggplot(top_10_c, aes(x = year_added, color = type)) + geom_density() + facet_wrap(~country) + theme(ax
```

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



#3 genre overtime

#top 10 genres

```
netflix %>%
  filter(!is.na(listed_in)) %>%
  group_by(listed_in) %>%
  count() %>%
  arrange(desc(n))
```

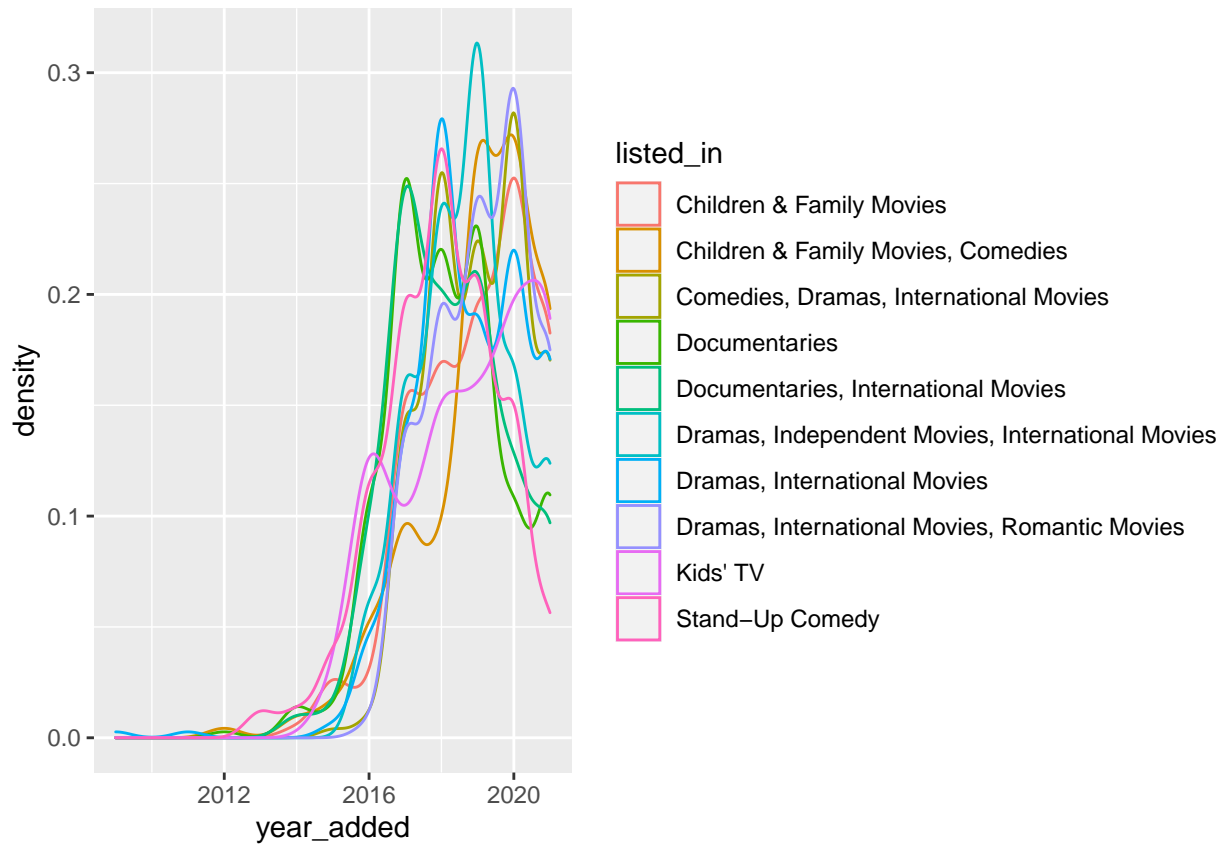
```
## # A tibble: 514 x 2
## # Groups:   listed_in [514]
##   listed_in          n
##   <chr>          <int>
## 1 Dramas, International Movies 362
## 2 Documentaries 359
## 3 Stand-Up Comedy 334
## 4 Comedies, Dramas, International Movies 274
## 5 Dramas, Independent Movies, International Movies 252
## 6 Kids' TV 220
## 7 Children & Family Movies 215
## 8 Children & Family Movies, Comedies 201
## 9 Documentaries, International Movies 186
## 10 Dramas, International Movies, Romantic Movies 180
## # ... with 504 more rows
```

```
x1 <- list("Dramas, International Movies", "Documentaries", "Stand-Up Comedy", "Comedies, Dramas, Internat
```

```
top_10_g <-netflix %>%
  filter(listed_in %in% x1)

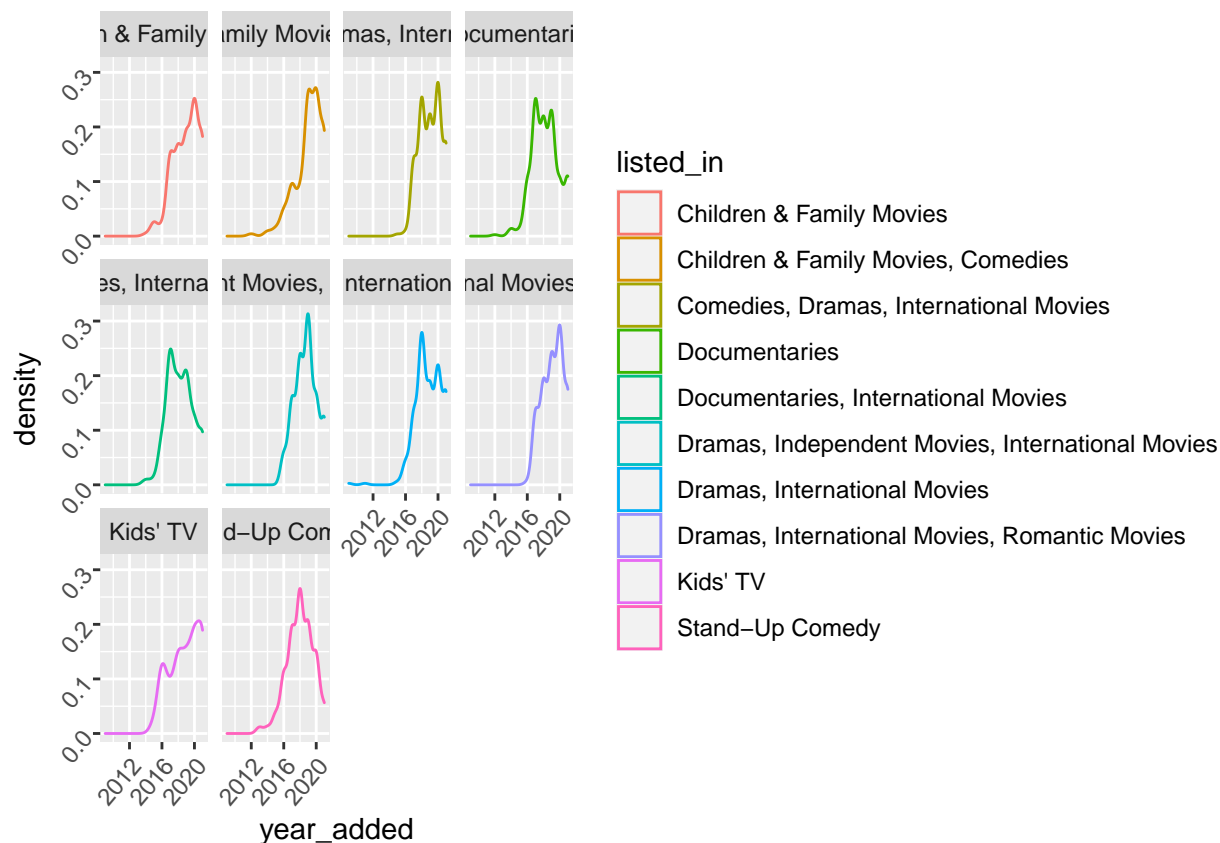
ggplot(data = top_10_g, mapping = aes(x = year_added, color = listed_in)) +geom_density()

## Warning: Removed 1 rows containing non-finite values (stat_density).
```



```
ggplot(data = top_10_g, mapping = aes(x = year_added, color = listed_in)) +geom_density() + facet_wrap(

## Warning: Removed 1 rows containing non-finite values (stat_density).
```



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

```
#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") %>%
  mutate(num_mins = as.integer(stri_sub(duration,1, -5)))
```

```
#glimpse(shows)
```

```
#glimpse(movies)
```

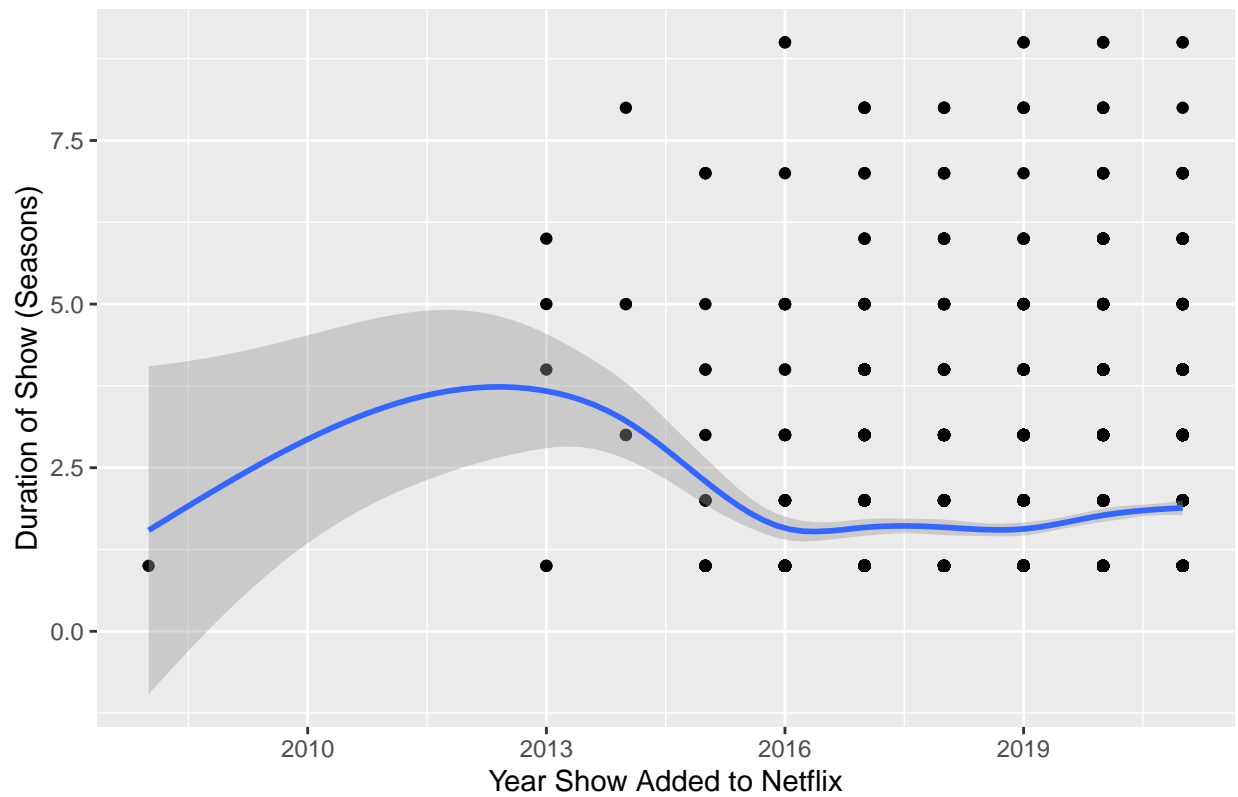
```
ggplot(data = shows,aes(x = year_added, y = num_seasons)) + geom_point() + geom_smooth() + labs(title =
```

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

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

```
## Warning: Removed 10 rows containing missing values (geom_point).
```

Year Added vs Duration of Show



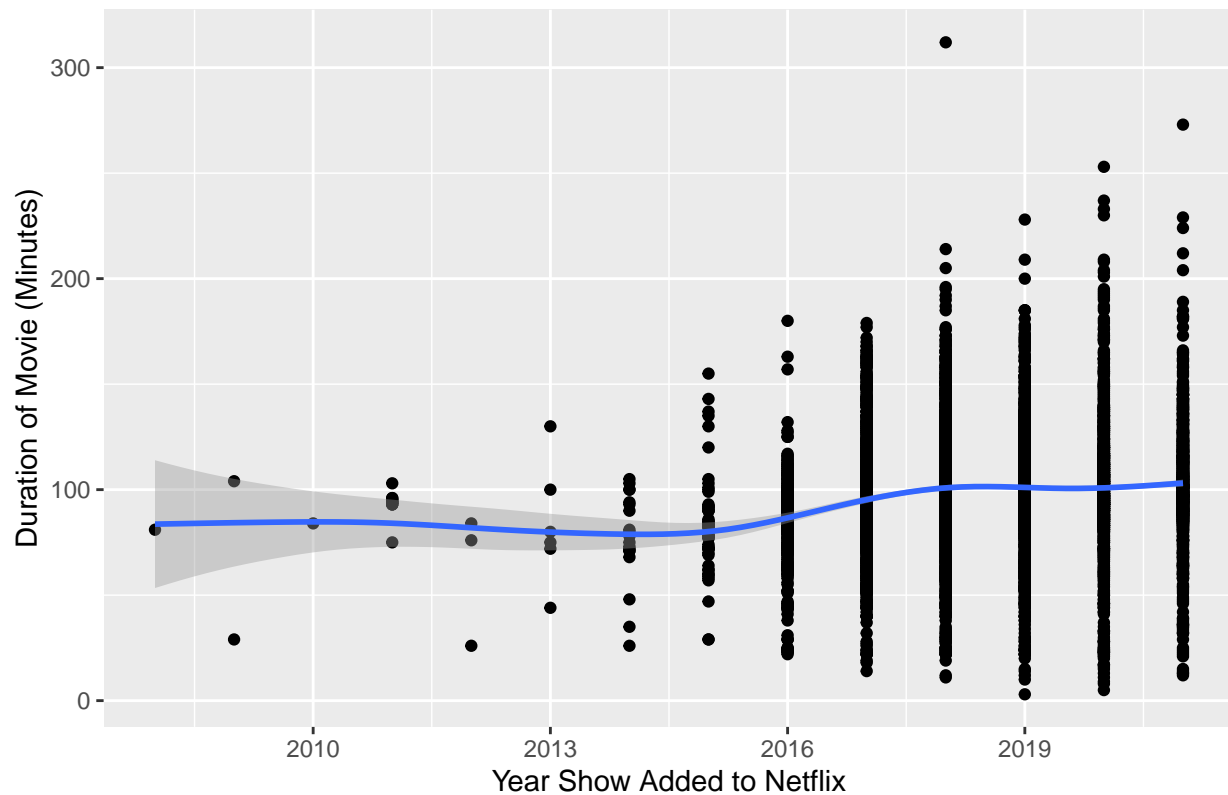
```
ggplot(data = movies, aes(x = year_added, y = num_mins)) + geom_point() + geom_smooth() + labs(title = "Year Added vs Duration of Show")
```

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

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```


Year Added vs Duration of Movie



```
#5 directors overtime

#top 10 directors in all content

netflix %>%
  group_by(director) %>%
  filter(!is.na(director)) %>%
  count() %>%
  arrange(desc(n))
```

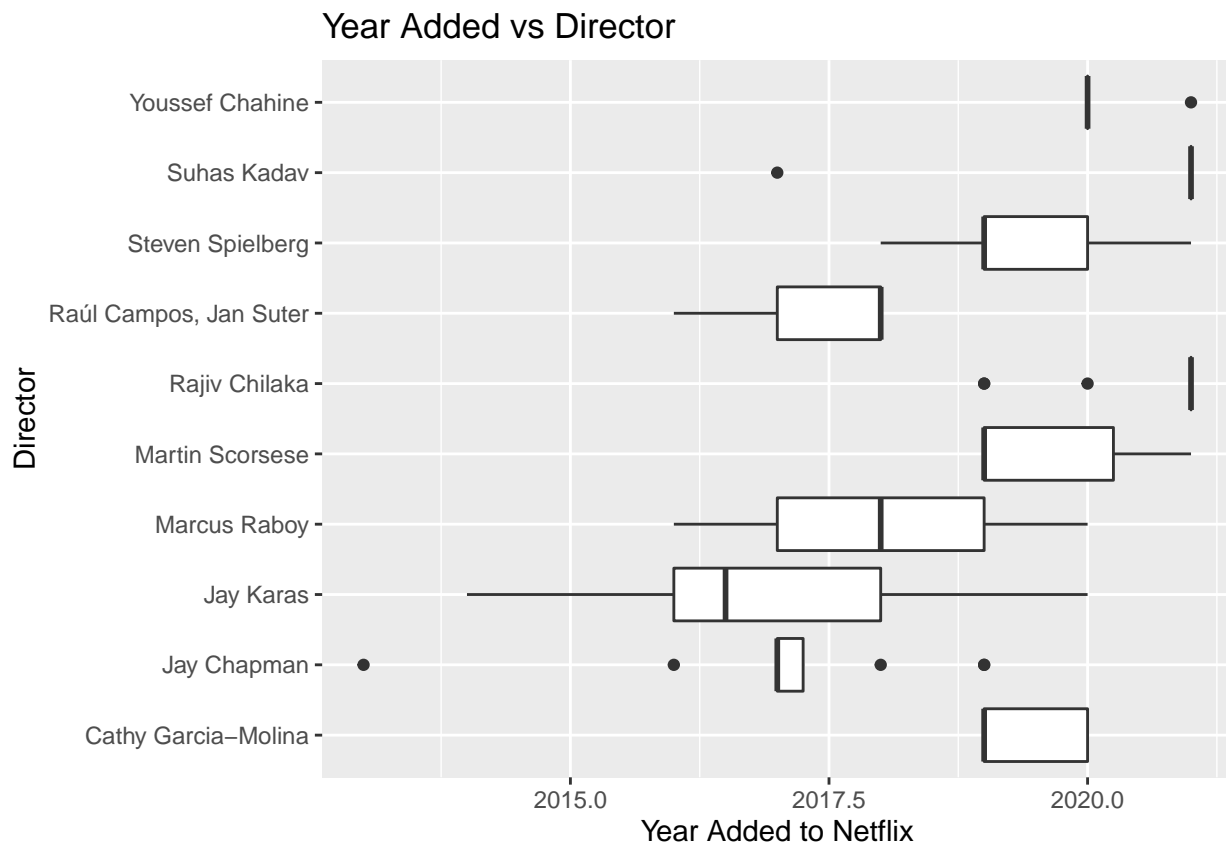
```
## # A tibble: 4,528 x 2
## # Groups:   director [4,528]
##   director          n
##   <chr>          <int>
## 1 Rajiv Chilaka    19
## 2 Raúl Campos, Jan Suter 18
## 3 Marcus Raboy    16
## 4 Suhas Kadav     16
## 5 Jay Karas       14
## 6 Cathy Garcia-Molina 13
## 7 Jay Chapman     12
## 8 Martin Scorsese  12
## 9 Youssef Chahine  12
## 10 Steven Spielberg 11
## # ... with 4,518 more rows
```

```

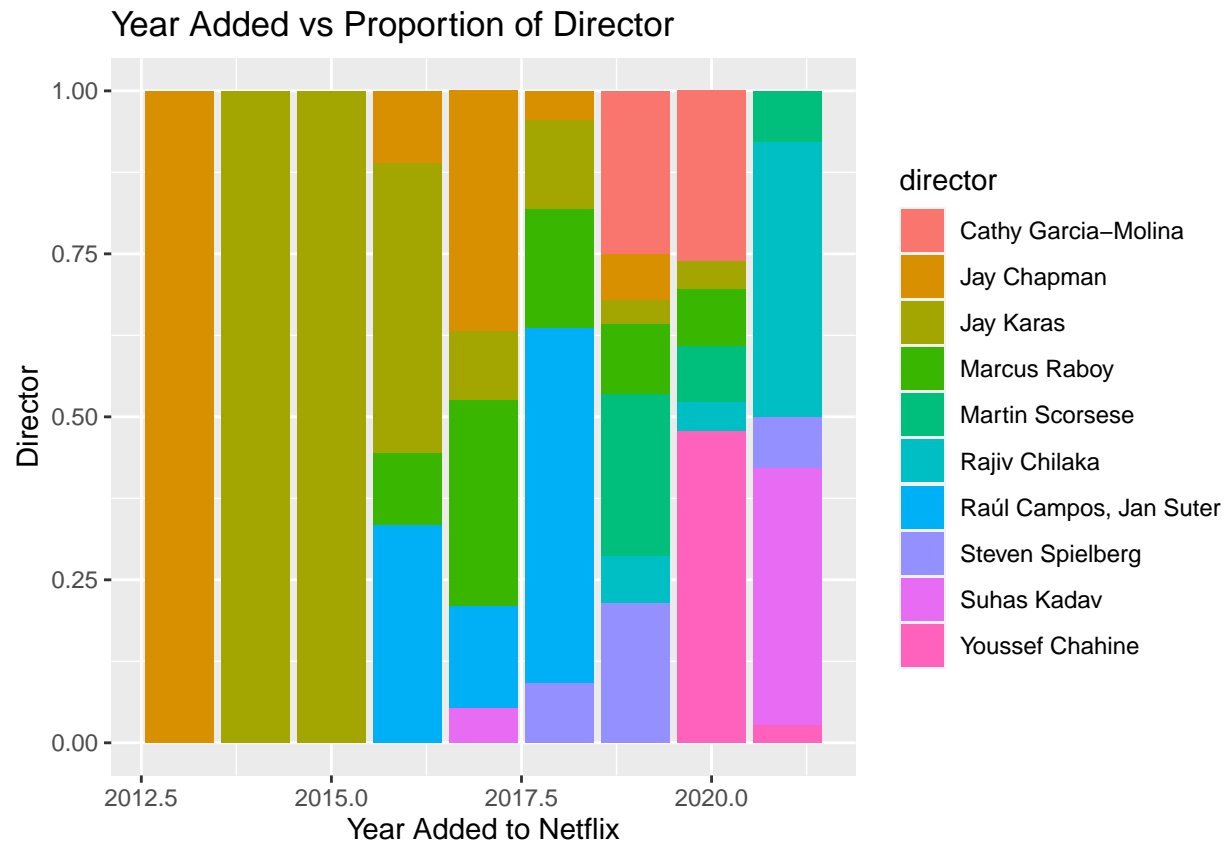
x2 <- list("Rajiv Chilaka", "Raúl Campos, Jan Suter", "Marcus Raboy", "Suhas Kadav", "Jay Karas", "Cath
top_10_d <-netflix %>%
  filter(director %in% x2)
glimpse(top_10_d)

## Rows: 143
## Columns: 13
## $ show_id      <chr> "s42", "s330", "s407", "s408", "s409", "s410", "s411", "s~
## $ type         <chr> "Movie", "Movie", "Movie", "Movie", "Movie", "Movie", "Mo~
## $ title        <chr> "Jaws", "Catch Me If You Can", "Chhota Bheem - Neeli Paha~
## $ director     <chr> "Steven Spielberg", "Steven Spielberg", "Rajiv Chilaka", ~
## $ cast         <chr> "Roy Scheider, Robert Shaw, Richard Dreyfuss, Lorraine Ga~
## $ country      <fct> "United States", "United States, Canada", NA, NA, NA, NA,~
## $ date_added   <chr> "September 16, 2021", "August 1, 2021", "July 22, 2021", ~
## $ release_year <int> 1975, 2002, 2013, 2009, 2011, 2010, 2013, 2013, 2013, 201~
## $ rating       <chr> "PG", "PG-13", "TV-Y7", "TV-Y7", "TV-Y7", "TV-Y7", "TV-Y7~
## $ duration     <chr> "124 min", "142 min", "64 min", "68 min", "67 min", "66 m~
## $ listed_in    <chr> "Action & Adventure, Classic Movies, Dramas", "Dramas", "~
## $ description  <chr> "When an insatiable great white shark terrorizes Amity Is~
## $ year_added   <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 202~
ggplot(data = top_10_d, aes(x = year_added, y=director)) +geom_boxplot() + labs(title = "Year Added vs I

```



```
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
```

```
###2
```

```
#both_country_count <- netflix %>%
```

```
# filter(!is.na(country)) %>%
```

```
# group_by(country) %>%
```

```
# count() %>%
```

```
# arrange(desc(n))
```

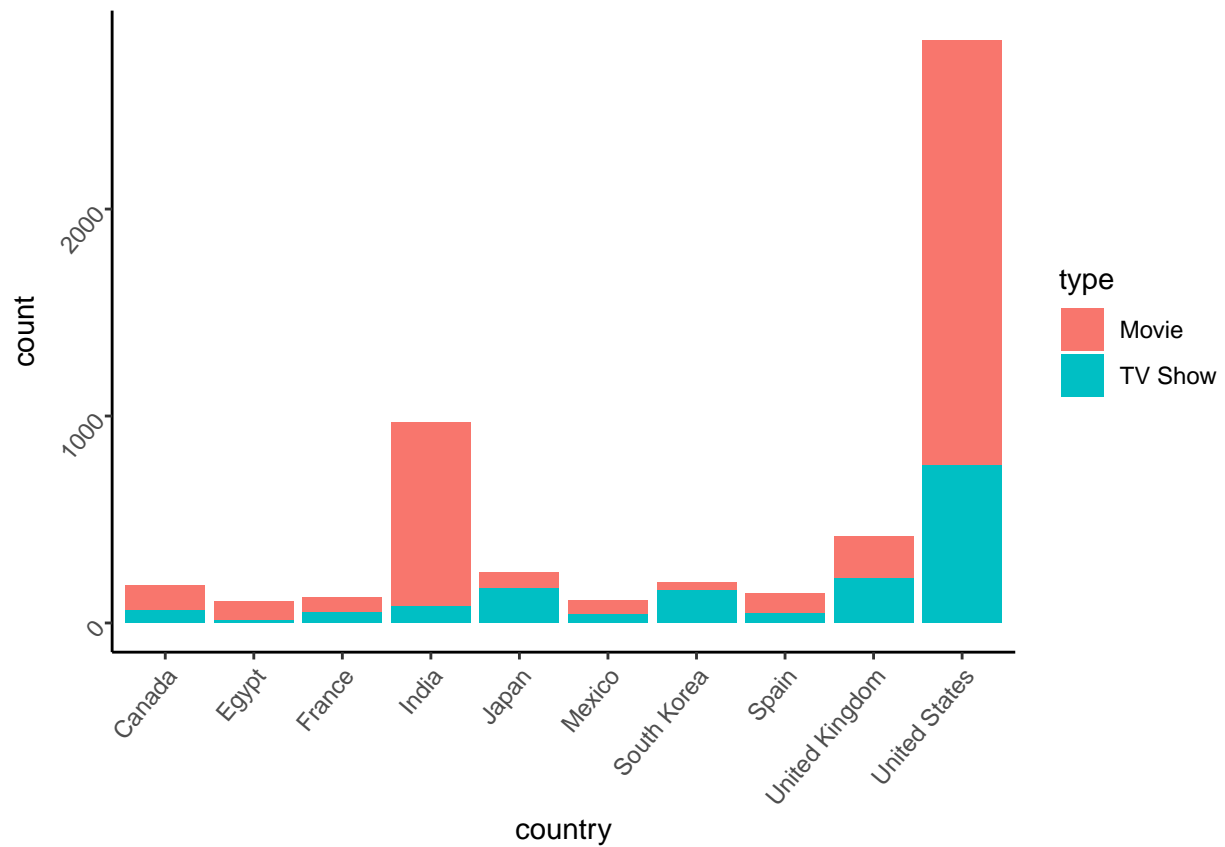
```
#glimpse(both_country_count)
```

```
#bar plot on the count of movies and shows within top 10 countries
```

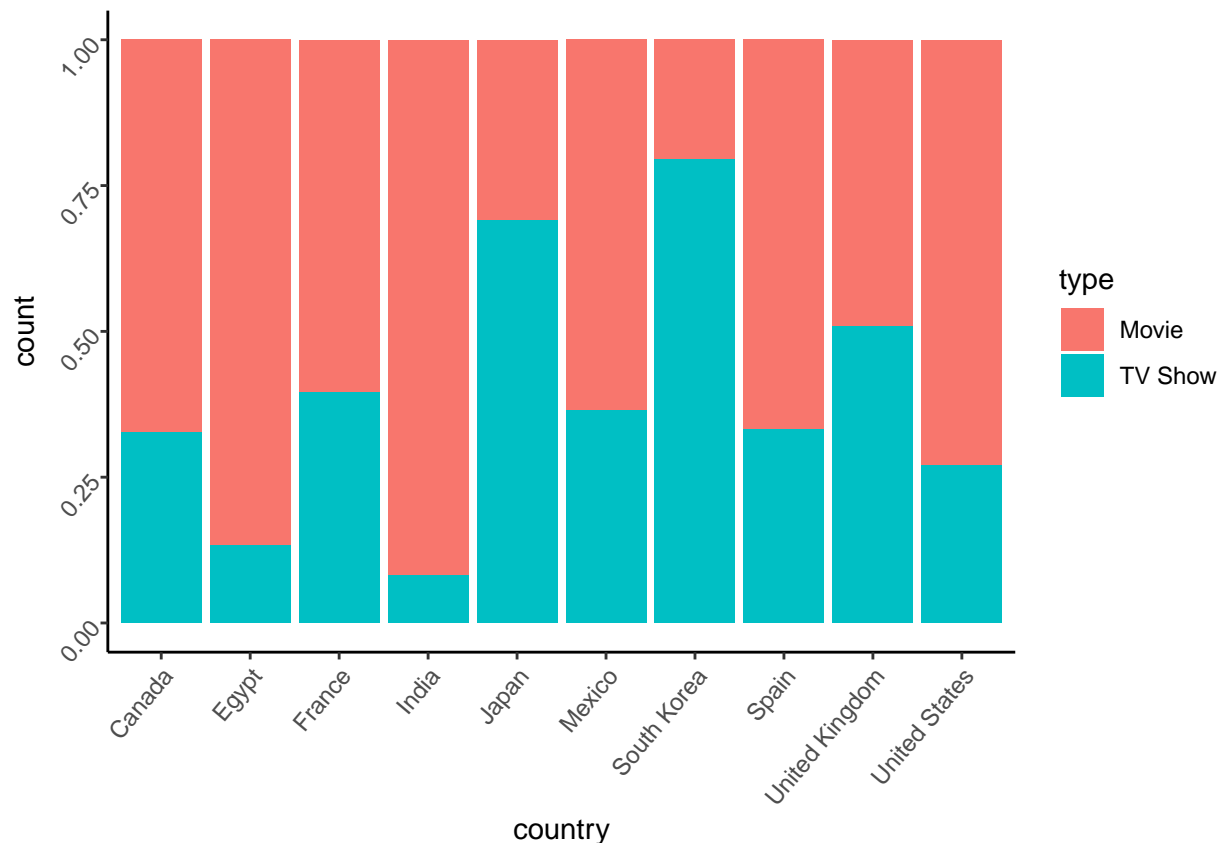
```
ggplot(top_10_c, aes(x = country, fill = type)) +
```

```
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>.

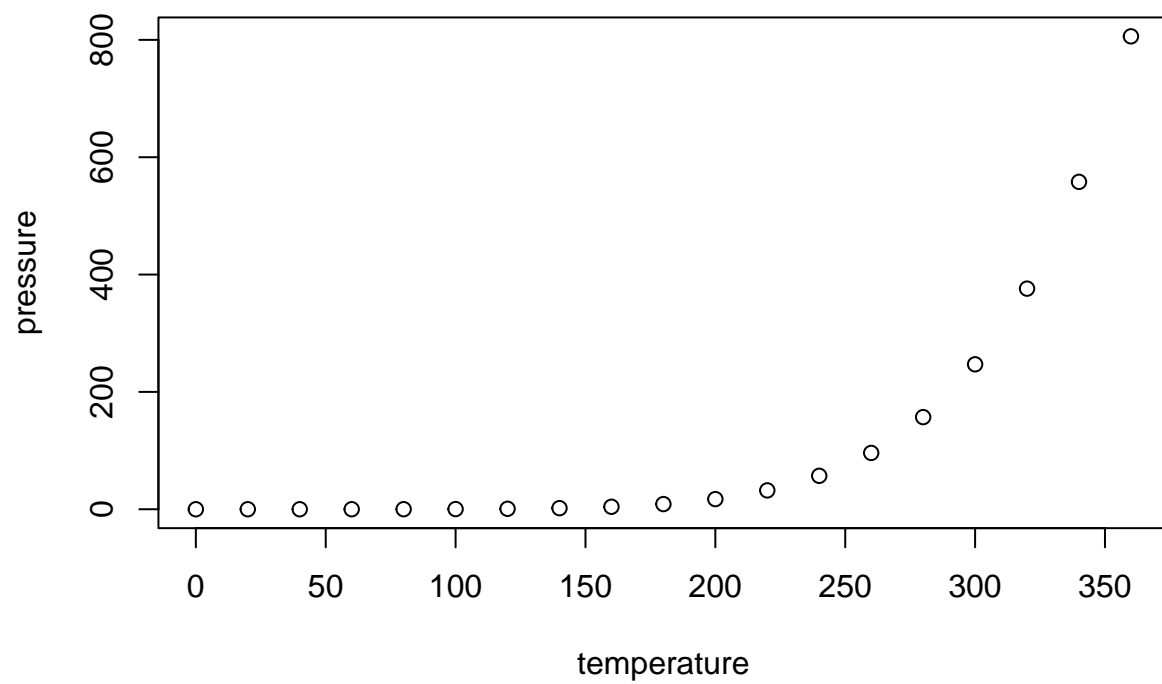
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.