Analysis Netflix

Emanuel Michele Soda

11/16/2021

Read data from the tidytuesday project

The datra are read from the tidytuesday repository using the package **tidytuesdayR**. The table is trasformed into a tibble for better visualizzation and all the character columns are
trasformed into factor.

Lets set the theme globally

```
theme_set(theme_light())
```

```
data <- read_csv(file = "tuesdata.csv") %>%
  tibble() %>%
  mutate_if(is_character, factor)
```

To have a first and quick look of the data we can use the summary function

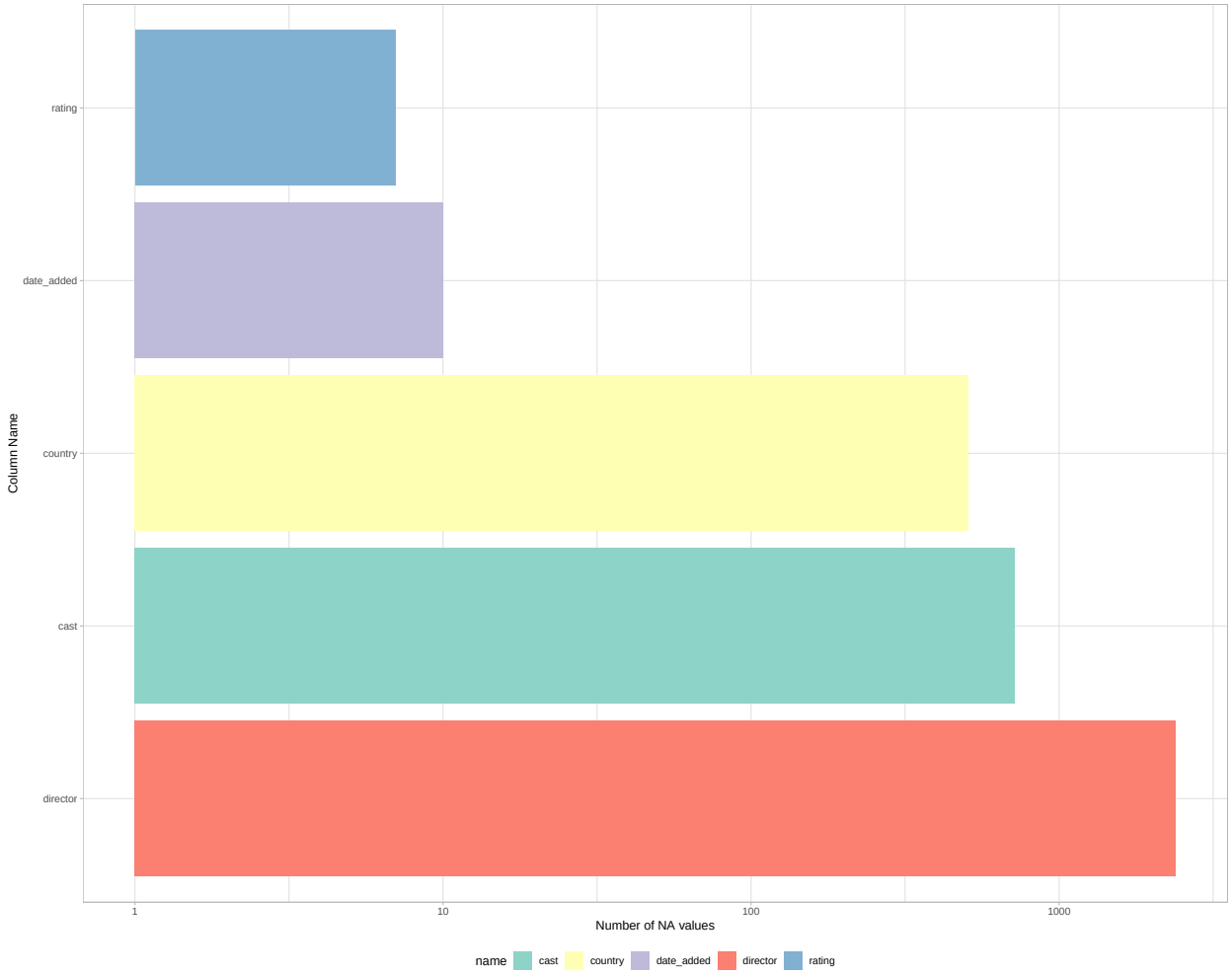
```
data %>%  summary()
```

```
##     show_id          type                                  title
##  s1     :   1   Movie  :5377   ¡Ay, mi madre!                :   1
##  s10    :   1   TV Show:2410   '89                           :   1
##  s100   :   1                  (T)ERROR                      :   1
##  s1000  :   1                  (Un)Well                      :   1
##  s1001  :   1                  #Alive                        :   1
##  s1002  :   1                  #AnneFrank - Parallel Stories:   1
##  (Other):7781                  (Other)                       :7781
##                    director                     cast                country
##  Raúl Campos, Jan Suter:  18   David Attenborough:  18   United States :2555
##  Marcus Raboy          :  16   Samuel West       :  10   India         : 923
##  Jay Karas             :  14   Jeff Dunham       :   7   United Kingdom: 397
##  Cathy Garcia-Molina   :  13   Craig Sechler     :   6   Japan         : 226
##  Jay Chapman           :  12   Kevin Hart        :   6   South Korea   : 183
##  (Other)               :5325   (Other)           :7022   (Other)       :2996
##  NA's                  :2389   NA's              : 718   NA's          : 507
##              date_added    release_year      rating          duration
##  January 1, 2020  : 119   Min.   :1925   TV-MA  :2863   1 Season :1608
##  November 1, 2019 :  96   1st Qu.:2013   TV-14  :1931   2 Seasons: 382
##  December 31, 2019:  76   Median :2017   TV-PG  : 806   3 Seasons: 184
##  March 1, 2018    :  76   Mean   :2014   R      : 665   90 min   : 136
##  October 1, 2018  :  72   3rd Qu.:2018   PG-13  : 386   93 min   : 131
```

```
##  (Other)          :7338   Max.   :2021   (Other):1129   91 min   : 125
##  NA's             :  10                  NA's   :   7   (Other)  :5221
##                                             listed_in
##  Documentaries                                    : 334
##  Stand-Up Comedy                                  : 321
##  Dramas, International Movies                     : 320
##  Comedies, Dramas, International Movies           : 243
##  Dramas, Independent Movies, International Movies: 215
##  Kids' TV                                         : 205
##  (Other)                                          :6149
##
##  A surly septuagenarian gets another chance at her 20s after having her photo snapped at a studio tha
##  Multiple women report their husbands as missing but when it appears they are looking for the same ma
##  A scheming matriarch plots to cut off her disabled stepson and his wife from the family fortune, cre
##  A young Han Solo tries to settle an old score with the help of his new buddy Chewbacca, a crew of sp
##  After growing up enduring criticism from his father, a young man finds his world shaken upon learnin
##  An affable, newly appointed college warden proves to be no ordinary man when an old enemy resurfaces
##  (Other)
```

Lets check if there are **NA** values. As can be seen the column which has more na values is director. NB: are plotted oly the column which contains at least 1 **NA**.

```
data %>%
  select(everything()) %>%  # replace to your needs
  summarise_all(funs(sum(is.na(.)))) %>%
  pivot_longer(cols = everything()) %>%
  filter(value > 0) %>%

  ggplot(., aes(x = value, y = reorder(name, -value), fill = name)) +
  geom_bar(stat="identity") +
  scale_fill_brewer(palette = "Set3") +
  scale_x_log10() +
  xlab("Number of NA values") +
  ylab("Column Name") +
  theme(legend.position = "bottom")
```

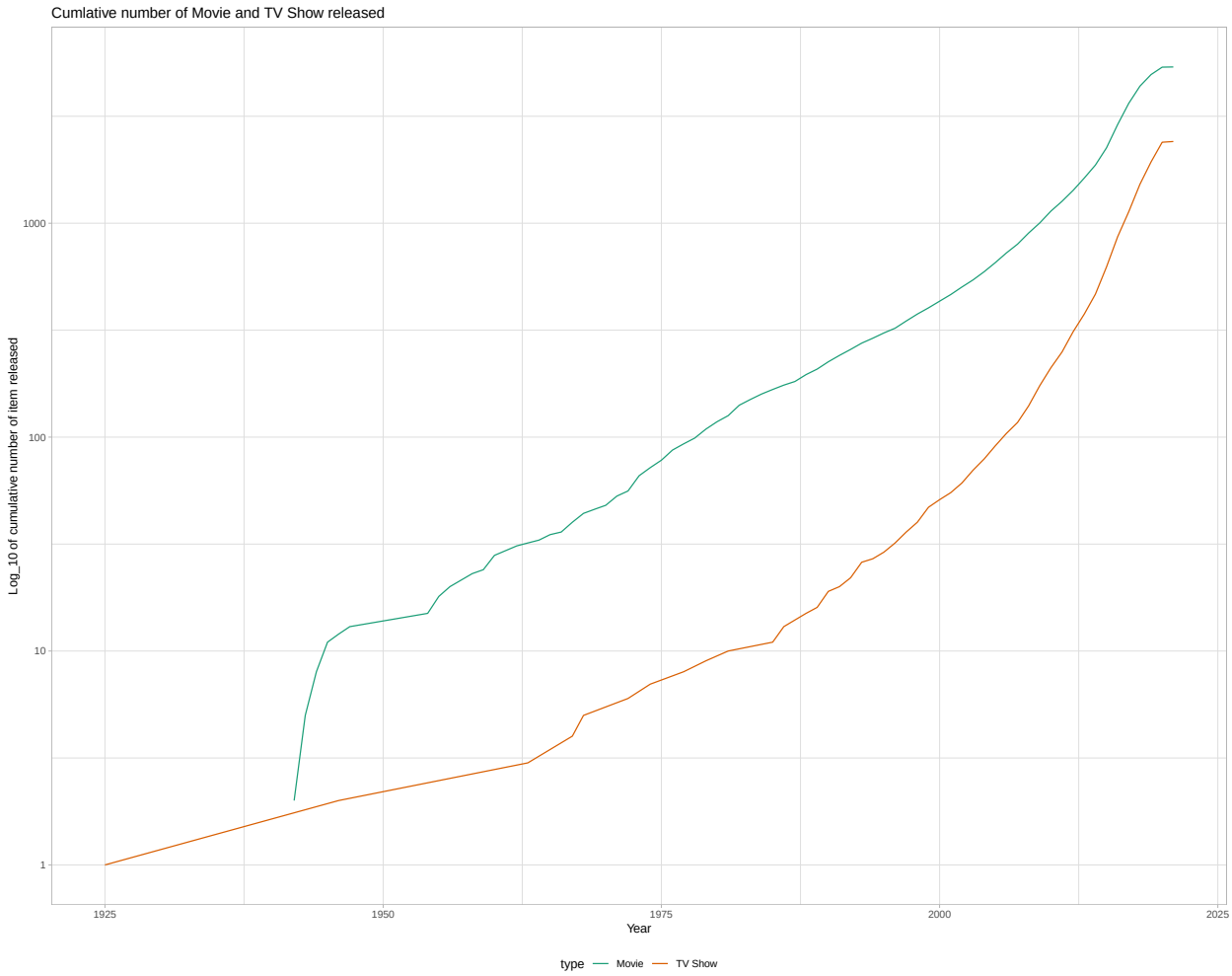


Data visualization

As can the summary functio shows there are two type of item in the Netflix collection **Movie** and **TV show**, can be interesting to see the number of those item released over year. To do this we can compute the cumulative sum over time divided in the two groups. As can be seen from the line plot the number of film release is always bigger than the number of TV Show. Moreover, as reported the y axis is in log scale, for this reason the trend which is linear in the log scale is actually exponential.

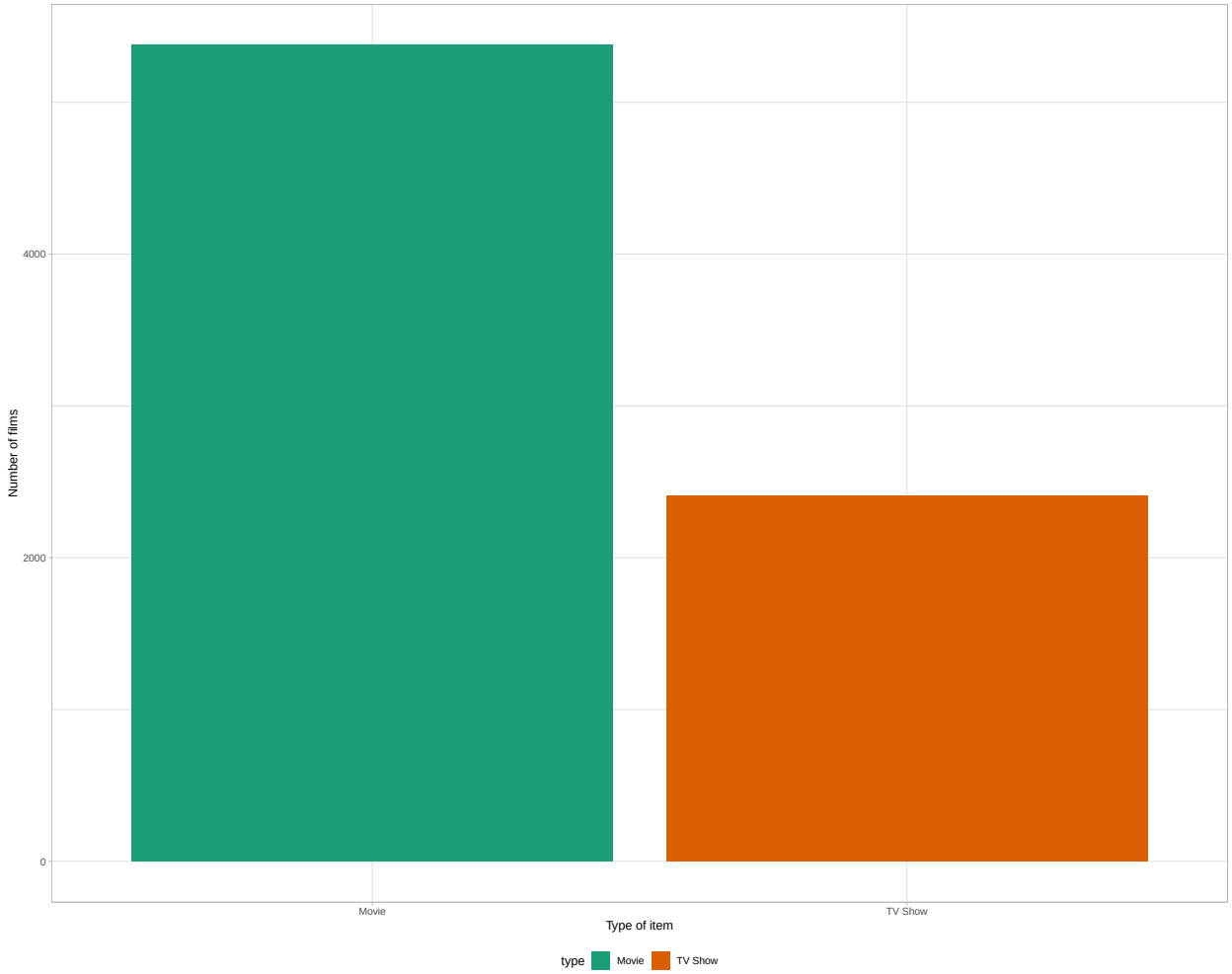
```
data %>%
  group_by(type, release_year) %>%
  summarize(n=n()) %>%
  mutate(cum = cumsum(n)) %>%

 ggplot(., aes(x=release_year, y=cum, col=type)) +
  geom_line() +
  scale_y_log10()  +
  scale_color_brewer(palette = "Dark2") +
  ylab("Log_10 of cumulative number of item released") +
  xlab("Year") +
  theme_light() +
```

```
  theme(legend.position = "bottom") +
  ggtitle("Cumlative number of Movie and TV Show released")
```



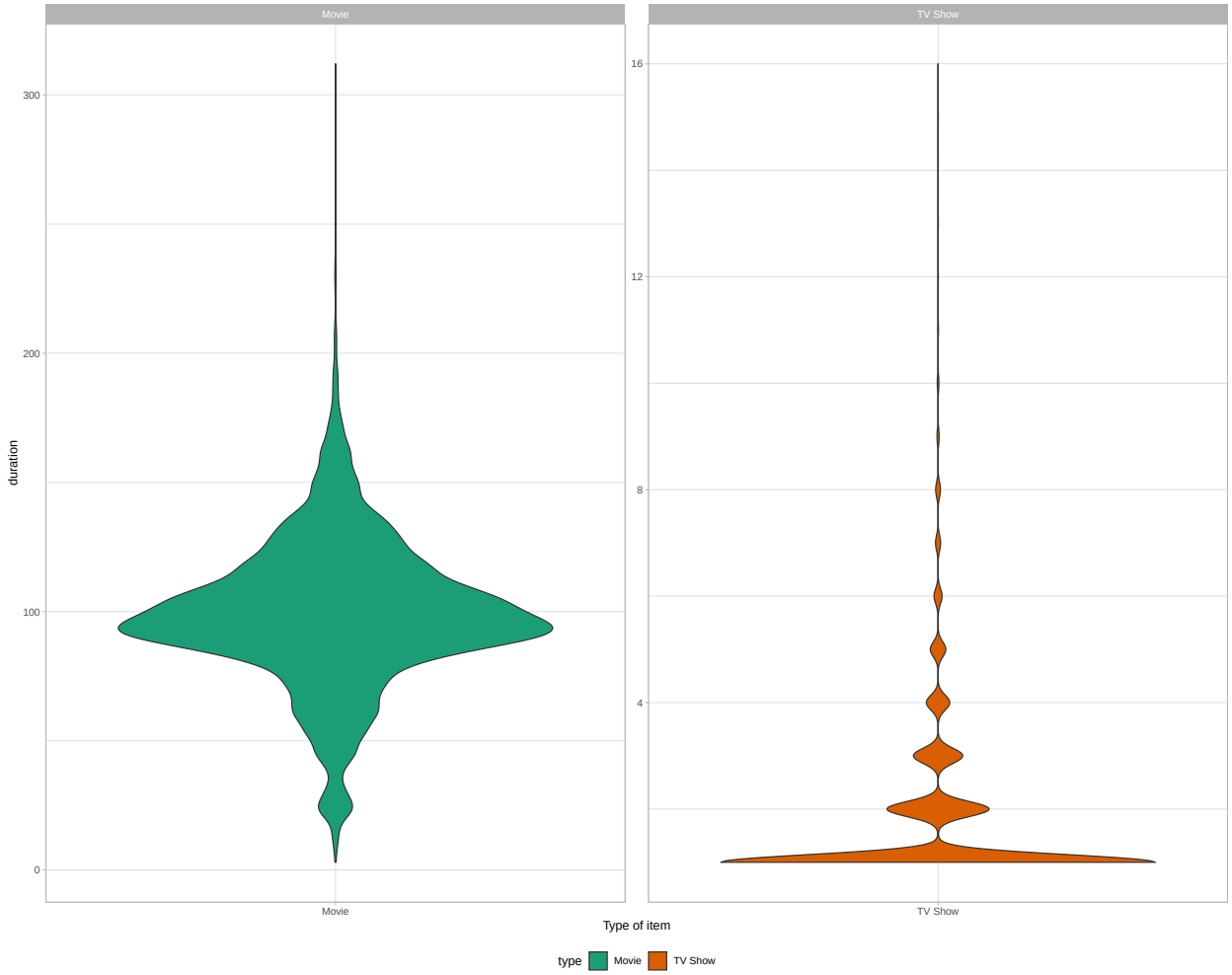
So, as can be imagine from the first plot the number of film is much bigger than the number of TV show

```
data %>%
  group_by(type) %>%
  count() %>%

  ggplot(., aes(x = type, y = n, fill = type)) +
  geom_bar(stat="identity") +
  scale_fill_brewer(palette = "Dark2") +
  theme_light() +
  ylab("Number of films") +
  xlab("Type of item") +
  theme(legend.position = "bottom")
```



Could be of interest to see the duration of each item. On this topic as to be clarify that the duration is in minutes for what concern the Movie while is in season for the TV Shows.

```
data %>%
  mutate(duration = as.integer(str_split_fixed(duration," ", 2)[, 1])) %>%
  group_by(type, duration)   %>%
  summarise(duration, .groups = "drop") %>%

  ggplot(., aes(x = type, y = duration, fill = type)) +
  geom_violin() +
  #geom_boxplot() +
  scale_fill_brewer(palette = "Dark2") +
  theme_light() +
  xlab("Type of item") +
  theme(legend.position = "bottom")   +
  facet_wrap("type", scales = "free")
```



From the plot can be seen that some TV shows have more than 13 season. Lets find out which are.

```
data %>%
mutate(duration = as.integer(str_split_fixed(duration," ", 2)[, 1])) %>%
  filter(type == "TV Show") %>%
  filter(duration > 13) %>%
  summarise(title, duration, cast, country, release_year, description)
```

```
## # A tibble: 3 x 6
##   title          duration cast          country   release_year description
##   <fct>             <int> <fct>         <fct>            <dbl> <fct>
## 1 Grey's Anatomy       16 Ellen Pompeo~ United S~         2019 Intern (and even~
## 2 NCIS                 15 Mark Harmon,~ United S~         2017 Follow the quirk~
## 3 Supernatural         15 Jared Padale~ United S~         2019 Siblings Dean an~
```

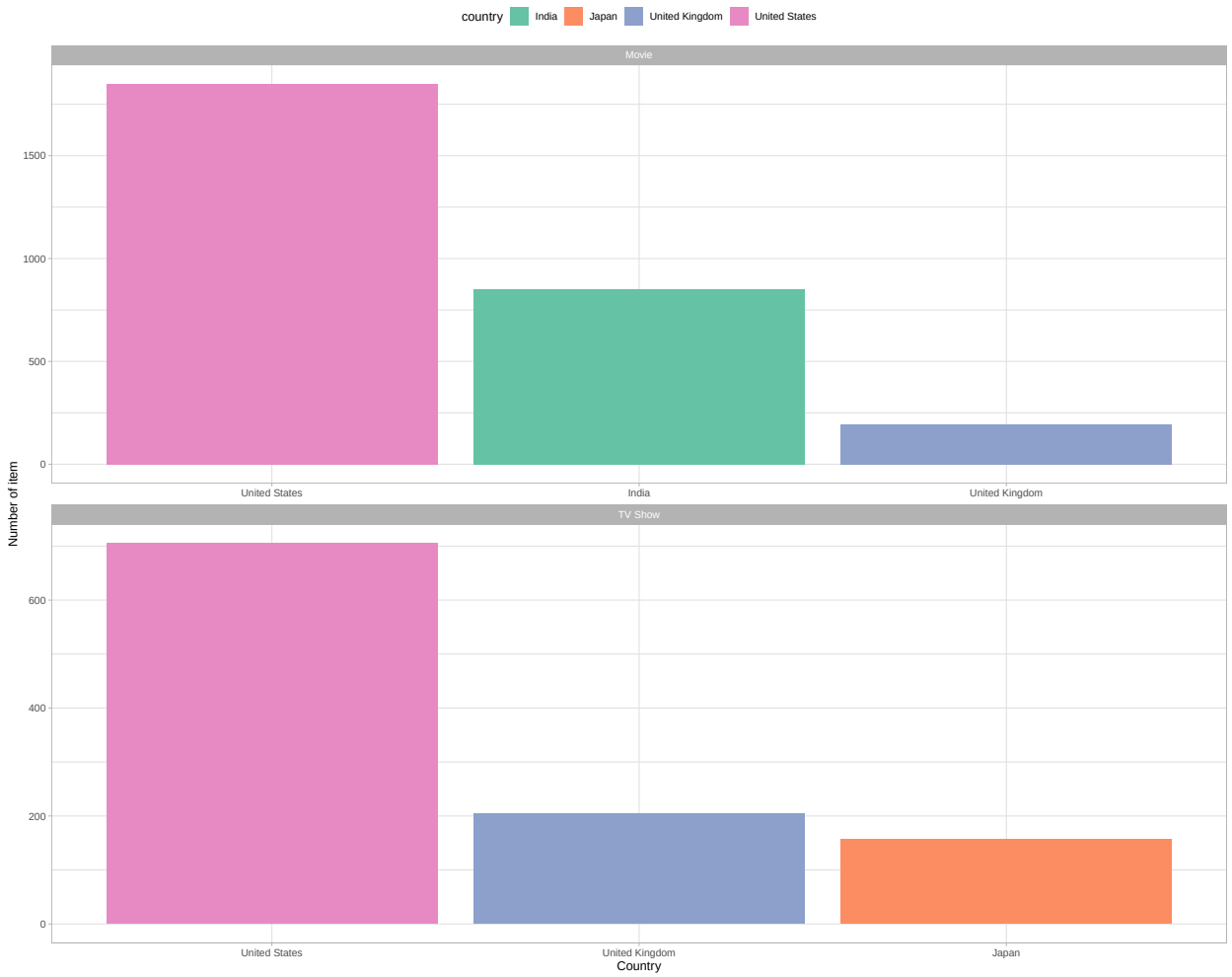
From the plot can be seen that some Movies are longer than 250 minutes Lets find out which are.

```
data %>%
mutate(duration = as.integer(str_split_fixed(duration," ", 2)[, 1])) %>%
  filter(type == "Movie") %>%
  filter(duration > 250) %>%
  summarise(title, duration, cast, country, release_year, description)
```

```
## # A tibble: 2 x 6
##   title                       duration cast    country  release_year description
##   <fct>                          <int> <fct>   <fct>           <dbl> <fct>
## 1 Black Mirror: Bandersnatch       312 Fionn ~ United ~         2018 In 1984, a ~
## 2 The School of Mischief           253 Suhair~ Egypt            1973 A high scho~
```

Lets plot the top 3 cuntry by Movie and TV Shows.As can be seen from the plot the US has the most Movie and TV Show released

```
data %>%
  group_by(country, type) %>%
  count() %>%
  drop_na() %>%
  group_by(type) %>%
  slice_max(order_by = n, n = 3)   %>%

  ggplot(., aes(x = reorder(country, -n), y = n, fill= country)) +
  geom_bar(stat="identity") +
  #scale_y_log10() +
  scale_fill_brewer(palette = "Set2") +
  theme_light() +
  xlab("Country") +
  ylab("Number of item") +
  theme(legend.position = "top") +
  facet_wrap("type", scales = "free", ncol = 1)
```



Lets see the item on the Netflix catalog in which is present **Leonardo DiCaprio** as part of the cast.

```
data  %>%
  filter(str_detect(string = cast, regex('DiCaprio', ignore_case = T))) %>%
  summarise(title, type, cast)
```

```
## # A tibble: 8 x 3
##   title                       type  cast
##   <fct>                       <fct> <fct>
## 1 Before the Flood            Movie Leonardo DiCaprio
## 2 Catch Me If You Can         Movie Leonardo DiCaprio, Tom Hanks, Christopher W~
## 3 Django Unchained            Movie Jamie Foxx, Christoph Waltz, Leonardo DiCap~
## 4 Gangs of New York           Movie Leonardo DiCaprio, Daniel Day-Lewis, Camero~
## 5 Inception                   Movie Leonardo DiCaprio, Joseph Gordon-Levitt, El~
## 6 Revolutionary Road          Movie Leonardo DiCaprio, Kate Winslet, Kathy Bate~
## 7 The Departed                Movie Leonardo DiCaprio, Matt Damon, Jack Nichols~
## 8 What's Eating Gilbert Grape Movie Johnny Depp, Leonardo DiCaprio, Juliette Le~
```