Netflix Titles

Dataset

Group Members:
Prince 20/49053
Kriti Lal 20/49023

OBJECTIVES

- We will visualize the proportion of Netflix content by type (Movie/Tv shows).
- We will visualize the Top 12 Countries by Amount of content they have produced.
- We will visualize Netflix's growth in amount of content as a function of time.
- We will visualize the amount of content added per month.
- We will visualize the distribution of Netflix content by rating classes.
- We will visualize movie duration in top 12 countries.
- We will also visualize Tv shows duration (in seasons) in top 12 countries.

TABLE OF CONTENTS

INTRODUCTION

This dataset consists of listing of all the movies and Tv shows available on Netflix along with details such as cast, directors, ratings, release year, a Unique Id for movie/tv show, country where the movie/tv show was produced, date Tv show/movie was added on Netflix and total duration, etc. This dataset analysis helps you find the most popular and best rated Tv shows and movies according to different contents available in different countries. And in this dataset, we will analyze what Netflix users prefer more- Tv shows or movies.

1

BASIC DETAILS OF DATASET

```
Netflix_data = read.csv("netflix_titles.csv", na.strings = c("","NA"),
stringsAsFactors = FALSE)
print(class(Netflix_data))
```

```
> Netflix_data = read.csv("netflix_titles.csv", na.strings = c("","NA"), stringsAsFactors = FALSE)
> print(class(Netflix_data))
[1] "data.frame"
```

```
print(dim(Netflix_data))
```

```
> print(dim(Netflix_data))
[1] 6234   12
>
```

print(str(Netflix_data))

```
> print(str(Netflix_data))
'data.frame':   6234 obs. of  12 variables:
 $ show_id     : int  81145628 80117401 70234439 80058654 80125979 80163890 70304989 80164077 80117902 70304990 ...
 $ type        : chr  "Movie" "Movie" "TV Show" "TV Show" ...
 $ title       : chr  "Norm of the North: King Sized Adventure" "Jandino: Whatever it Takes" "Transformers Prime" "Transformers: Robots in Disguise" ...
 $ director    : chr  "Richard Finn, Tim Maltby" NA NA NA ...
 $ cast        : chr  "Alan Marriott, Andrew Toth, Brian Dobson, Cole Howard, Jennifer Cameron, Jonathan Holmes, Lee Tockar, Lisa Duru"| __truncated__ "Jandi
no Asporaat" "Peter Cullen, Sumalee Montano, Frank Welker, Jeffrey Combs, Kevin Michael Richardson, Tania Gunadi, Josh Keaton"| __truncated__ "Will Friedle,
 Darren Criss, Constance Zimmer, Khary Payton, Mitchell Whitfield, Stuart Allan, Ted McGinley, Peter Cullen" ...
 $ country     : chr  "United States, India, South Korea, China" "United Kingdom" "United States" "United States" ...
 $ date_added  : chr  "September 9, 2019" "September 9, 2016" "September 8, 2018" "September 8, 2018" ...
 $ release_year: int  2019 2016 2013 2016 2017 2016 2014 2017 2017 2014 ...
 $ rating      : chr  "TV-PG" "TV-MA" "TV-Y7-FV" "TV-Y7" ...
 $ duration    : chr  "90 min" "94 min" "1 Season" "1 Season" ...
 $ listed_in   : chr  "Children & Family Movies, Comedies" "Stand-Up Comedy" "Kids' TV" "Kids' TV" ...
 $ description : chr  "Before planning an awesome wedding for his grandfather, a polar bear king must take back a stolen artifact from"| __truncated__ "Jandi
no Asporaat riffs on the challenges of raising kids and serenades the audience with a rousing rendition of"| __truncated__ "With the help of three human alli
es, the Autobots once again protect Earth from the onslaught of the Decepticon"| __truncated__ "When a prison ship crash unleashes hundreds of Decepticons on
 Earth, Bumblebee leads a new Autobot force to protect humankind." ...
NULL
```

2

DATA CLEANING

```
Netflix_data$rating<- as.factor(Netflix_data$rating)

#printing missing values by creating a new data frame.
a<-data.frame("Variable"=c(colnames(Netflix_data)), "Missing
Values"=sapply(Netflix_data, function(x) sum(is.na(x))), row.names = NULL)
print(a)
```

```
> print(a)
       Variable Missing.Values
1       show_id              0
2          type              0
3         title              0
4      director           1969
5          cast            570
6       country            476
7    date_added             11
8  release_year              0
9        rating             10
10     duration              0
11    listed_in              0
12  description              0
>
```

```
mode<- function(v){
  uniqv<- unique(v)
  uniqv[which.max(tabulate(match(v,uniqv)))]
}
Netflix_data$rating[is.na(Netflix_data$rating)] = mode(Netflix_data$rating)
b<-data.frame("Variable"=c(colnames(Netflix_data)), "Missing
Values"=sapply(Netflix_data, function(x) sum(is.na(x))), row.names = NULL)
print(b)
```

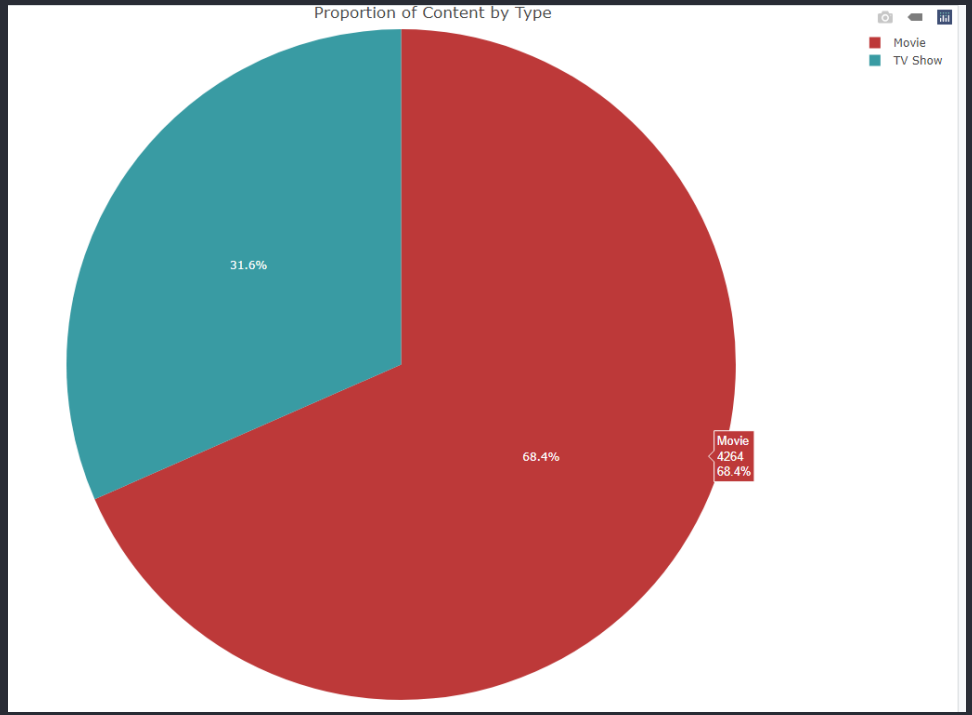
```
> print(b)
       Variable Missing.Values
1       show_id              0
2          type              0
3         title              0
4      director           1968
5          cast            570
6       country            476
7    date_added             11
8  release_year              0
9        rating              0
10     duration              0
11    listed_in              0
12  description              0
```

```
library("dplyr")
Netflix_data= distinct(Netflix_data, title, country, type, release_year, .keep_all =
TRUE)
print(Netflix_data)
```

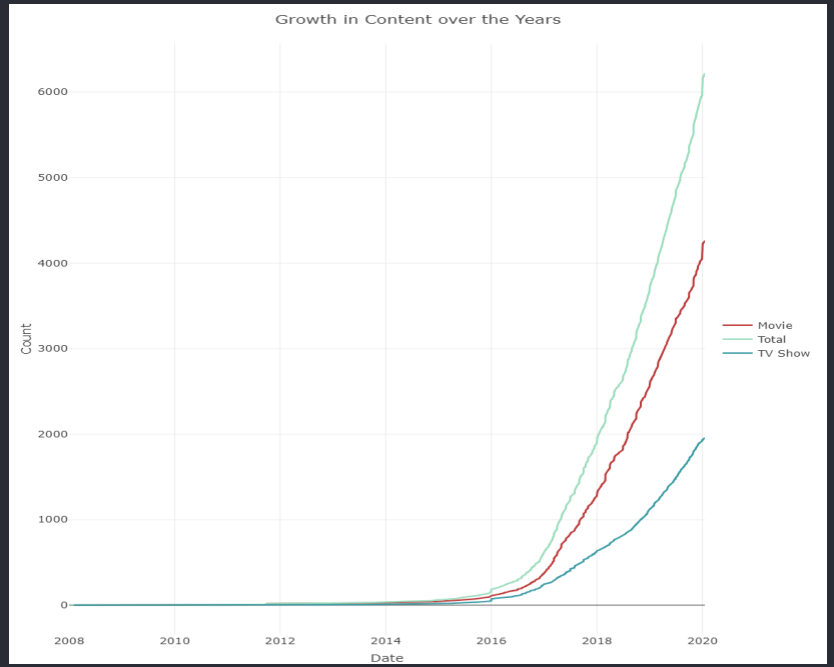
3

DATA VISUALIZATION

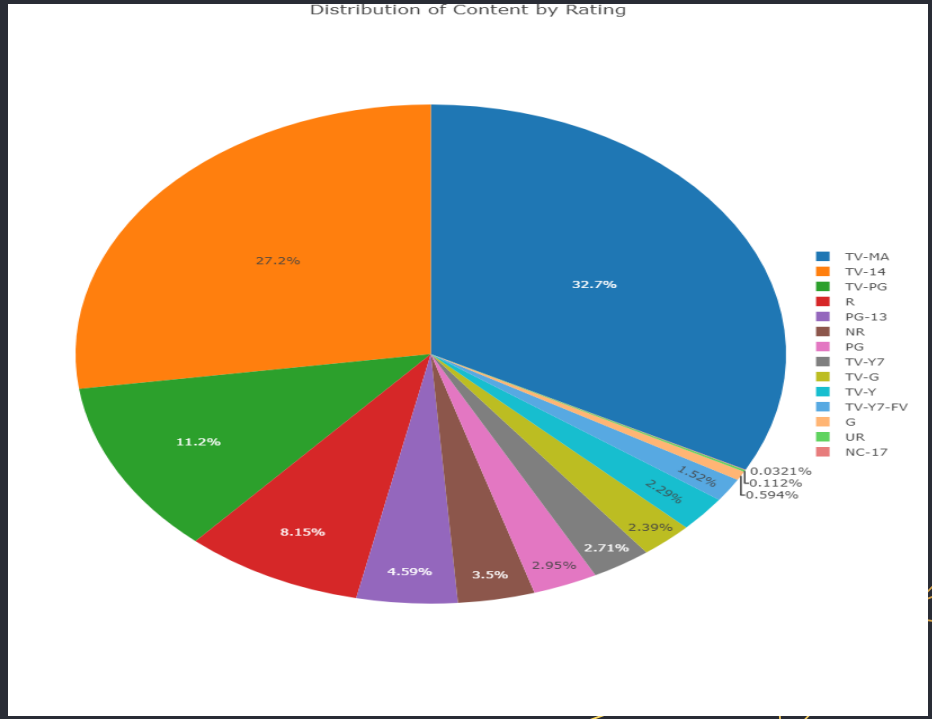
```
#question 1
library("plotly")
library("dplyr")
content_by_type <- Netflix_data %>% group_by(type) %>%
  summarise(count = n())
plot_ly(content_by_type, labels = ~type, values = ~count,
        type = 'pie', marker = list(colors = c("#bd3939", "#399ba3"))) %>%
  layout(title = "Proportion of Content by Type",legend = list(x = 200, y = 1))
```



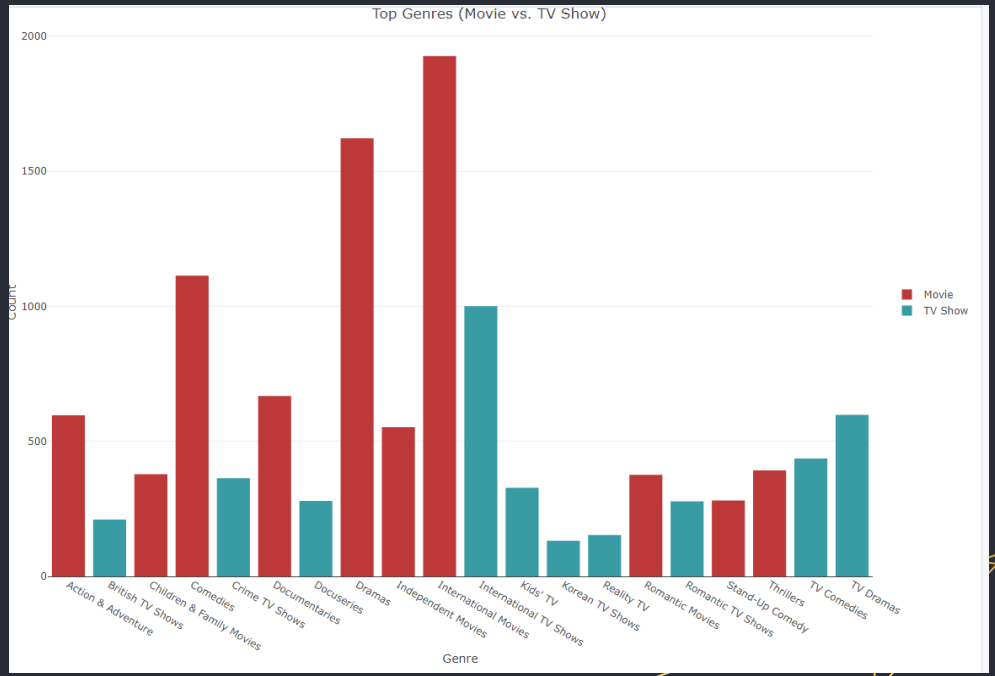
```
#question2
library("plotly")
library("dplyr")
df_by_date <- Netflix_data %>% group_by(date_added, type) %>%
  summarise(added_today = n()) %>%
  group_by(type)
plot_ly(full_data, x = ~date_added, y = ~total_number_of_content,  mode = 'lines', type =
'scatter',  color = ~type, colors = c("#bd3939",  "#9addbd", "#399ba3")) %>% layout(yaxis
= list(title = 'Count'), xaxis = list(title = 'Date'), title = "Growth in Content over the
Years",margin=list(t=54), legend = list(x = 100, y = 0.5))
```



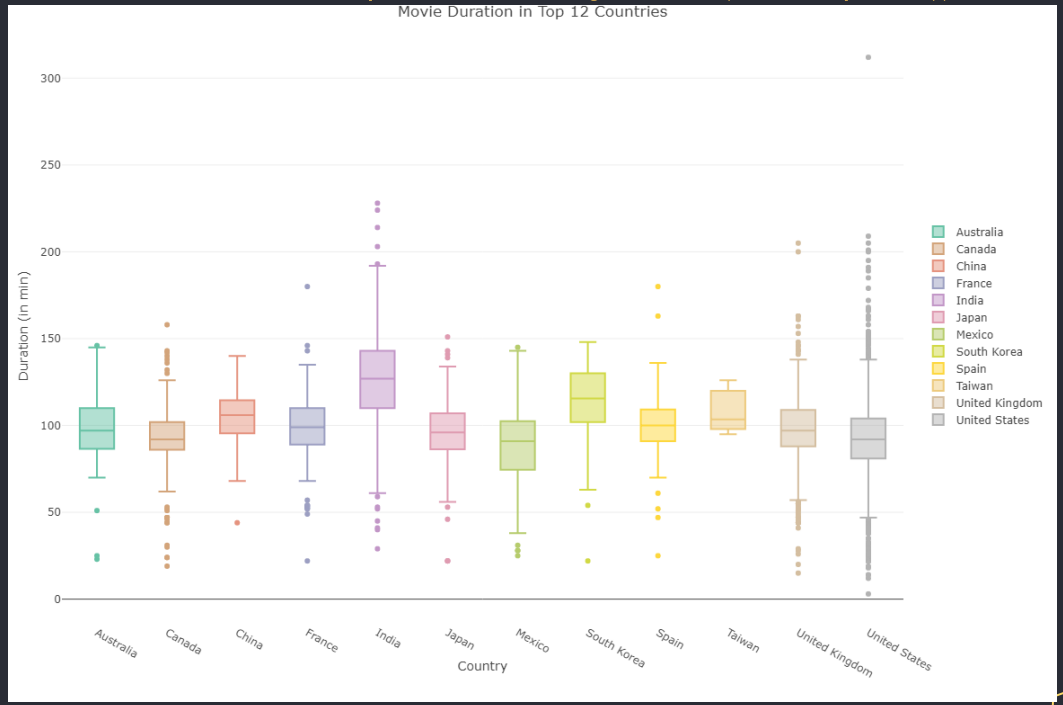
```
#question3
library("plotly")
library("dplyr")
df_by_rating <- Netflix_data %>% group_by(rating) %>%
  summarise(count = n())
plot_ly(df_by_rating, type = 'pie',
        labels = ~rating, values = ~count) %>%
  layout(title = "Distribution of Content by Rating",
         legend = list(x = 100, y = 0.5))
```



```
#question4
library("plotly")
library("dplyr")
df_by_listed_in <-  group_by(type, listed_in) %>%
 summarise(count = n())
plot_ly(df_by_listed_in, x = ~listed_in, y = ~count, type = 'bar', color = ~type,
      colors = c("#bd3939", "#399ba3")) %>% layout(xaxis = list(title = 'Genre'),
       yaxis = list(title = 'Count'),  title = "Top Genres (Movie vs. TV Show)",
       legend = list(x = 100, y = 0.5))
```



```
#question5
library("plotly")
library("dplyr")
duration_full_subset <- duration_full[duration_full$country %in%
c("United States", "India", "United Kingdom", "Canada", "France", "Japan", "Spain", "South
Korea", "Mexico", "Australia", "China", "Taiwan"),]
plot_ly(duration_full_subset, y = ~duration, color = ~country, type = "box") %>%
  layout(xaxis = list(title = "Country"), yaxis = list(title = 'Duration (in min)'),
         title = "Movie Duration in Top 12 Countries",legend = list(x = 100, y = 0.5))
```



```
#question 6
#Access the title of first 20 Movie/ TV Shows of data set.
head_movieshows<- head(Netflix_data$title,20)
print("Title of first 20 Movies and Tv shows: ")
print(head_movieshows)
```

```
> head_movieshows<- head(Netflix_data$title,20)
> print("Title of first 20 Movies and Tv shows: ")
[1] "Title of first 20 Movies and Tv shows: "
> print(head_movieshows)
 [1] "Norm of the North: King Sized Adventure"           "Jandino: Whatever it Takes"
 [3] "Transformers Prime"                                "Transformers: Robots in Disguise"
 [5] "#realityhigh"                                      "Apaches"
 [7] "Automata"                                          "Fabrizio Copano: Solo pienso en mi"
 [9] "Fire Chasers"                                      "Good People"
[11] "JoaquÃn Reyes: Una y no mÃ¡s"                      "Kidnapping Mr. Heineken"
[13] "Krish Trish and Baltiboy"                          "Krish Trish and Baltiboy: Battle of Wits"
[15] "Krish Trish and Baltiboy: Best Friends Forever"    "Krish Trish and Baltiboy: Comics of India"
[17] "Krish Trish and Baltiboy: Oversmartness Never Pays" "Krish Trish and Baltiboy: Part II"
[19] "Krish Trish and Baltiboy: The Greatest Trick"      "Love"
```

```
#question 7
#Find the total number of movies and TV shows in the data set.
a1<- aggregate(Netflix_data$show_id ~ Netflix_data$type, Netflix_data, length)
colnames(a1)= c("type","length")
print(a1)
```

```
> #Find the total number of movies and TV shows in the data set.
> a1<- aggregate(Netflix_data$show_id ~ Netflix_data$type, Netflix_data, length)
> colnames(a1)= c("type","length")
> print(a1)
     type length
1   Movie   4264
2 TV Show   1968
>
```

#question 8
#Display the number of TV Shows Produced by each country.
e<- subset(Netflix_data, type=="TV Show")
f<- aggregate(e$show_id~e$country, e, length)
colnames(f)= c("Country","Tv_Shows_Produced")
print(f)

```
> #Display the number of TV Shows Produced by each country.
>  e<- subset(Netflix_data, type=="TV Show")
>  f<- aggregate(e$show_id~e$country, e, length)
>  colnames(f)= c("Country","Tv_Shows_Produced")
>  print(f)
```

	Country	Tv_Shows_Produced
1	Argentina	12
2	Argentina, Spain	1
3	Argentina, United States, Mexico	1
4	Australia	39
5	Australia, Canada	1
6	Australia, New Zealand	1
7	Australia, New Zealand, United States	1
8	Australia, United States	2
9	Belgium	5
10	Belgium, Netherlands	1
11	Brazil	15
12	Canada	53
13	Canada, Australia	2
14	Canada, Brazil	1
15	Canada, France	1
16	Canada, Germany, South Africa	1
17	Canada, Japan, Netherlands	1
18	Canada, United Kingdom	2
19	Canada, United Kingdom, United States	2
20	Canada, United States	12
21	Canada, United States, France	1
22	Chile	3
23	China	31
24	China, Hong Kong	1
25	China, United Kingdom	1
26	Colombia	16
27	Colombia, Mexico, United States	2
28	Colombia, United States	1
29	Croatia	1
30	Cyprus, Austria, Thailand	1
31	Czech Republic, United States	1
32	Denmark	7
33	Denmark, Singapore, Canada, United States	1

```
#question 8
#Display the number of TV Shows Produced by each country.
e<- subset(Netflix_data, type=="TV Show")
f<- aggregate(e$show_id~e$country, e, length)
colnames(f)= c("Country","Tv_Shows_Produced")
print(f)
```

	Country	Tv_Shows_Produced
33	Denmark, Singapore, Canada, United States	1
34	Denmark, United States	1
35	Egypt	8
36	Finland	1
37	Finland, France	1
38	Finland, Germany	1
39	France	40
40	France, Belgium	2
41	France, Canada	1
42	France, Germany	1
43	France, South Korea, Japan	1
44	France, United Kingdom	1
45	France, United Kingdom, United States	1
46	France, United States	2
47	France, United States, Canada	1
48	Germany	7
49	Germany, Australia	1
50	Germany, France, Russia	1
51	Germany, United States	1
52	Germany, United States, Italy	1
53	Hong Kong	2
54	Iceland	1
55	India	53
56	India, Germany, France	1
57	India, United States	1
58	Indonesia	1
59	Ireland	4
60	Ireland, Canada, United States, United Kingdom	1
61	Ireland, United Kingdom, United States	2
62	Israel	8
63	Italy	5
64	Italy, Canada, France	1
65	Italy, Germany	1
66	Italy, South Africa, West Germany, Australia, United States	1
67	Italy, United Kingdom, France	1
68	Italy, United States	1
69	Japan	128

```r
#question 8
#Display the number of TV Shows Produced by each country.
e<- subset(Netflix_data, type=="TV Show")
f<- aggregate(e$show_id~e$country, e, length)
colnames(f)= c("Country","Tv_Shows_Produced")
print(f)
```

	Country	Tv_Shows_Produced
70	Japan, Canada	1
71	Japan, Canada, South Korea	1
72	Japan, United States	1
73	Lebanon	5
74	Lebanon, Jordan	1
75	Malaysia	6
76	Malaysia, Singapore, Hong Kong	1
77	Mauritius, South Africa	1
78	Mexico	29
79	Mexico, United States	4
80	Netherlands	2
81	Netherlands, Germany, Italy, Canada	1
82	New Zealand	2
83	Norway	5
84	Norway, Germany, Sweden	1
85	Norway, Sweden	1
86	Norway, United States	1
87	Pakistan	4
88	Philippines	1
89	Poland	3
90	Poland, United States	2
91	Russia	12
92	Saudi Arabia, Syria, Egypt, Lebanon, Kuwait	1
93	Singapore	9
94	Singapore, United States	1
95	South Korea	104
96	South Korea, Canada, United States, China	1
97	South Korea, France	1
98	South Korea, United States	3
99	Spain	37
100	Spain, Cuba	1
101	Spain, United Kingdom	1
102	Sweden	5
103	Switzerland	1
104	Taiwan	65
105	Thailand	17
106	Turkey	24
107	Turkey, Azerbaijan	1
108	Ukraine	2

#question 8

#Display the number of TV Shows Produced by each country.

```
e<- subset(Netflix_data, type=="TV Show")
f<- aggregate(e$show_id~e$country, e, length)
colnames(f)= c("Country","Tv_Shows_Produced")
print(f)
```

117	United Kingdom, Singapore	1
118	United Kingdom, South Africa, Australia, United States	1
119	United Kingdom, United States	10
120	United Kingdom, United States, Czech Republic	1
121	United Kingdom, United States, Greece, Italy, Germany	1
122	United Kingdom, United States, Spain, Germany, Greece, Canada	2
123	United Kingdom, West Germany	1
124	United States	550
125	United States, Brazil, South Korea, Mexico, Japan, Germany	2
126	United States, Canada	12
127	United States, Chile	1
128	United States, Colombia	1
129	United States, Colombia, Mexico	1
130	United States, Czech Republic	3
131	United States, France	1
132	United States, France, Canada	1
133	United States, France, Japan	6
134	United States, France, South Korea, Indonesia	1
135	United States, Hungary, Ireland, Canada	1
136	United States, Ireland	1
137	United States, Ireland, United Kingdom	1
138	United States, Israel, Italy, South Africa	1
139	United States, Italy	1
140	United States, Japan	10
141	United States, Japan, Canada	2
142	United States, Mexico	3
143	United States, Mexico, Colombia	1
144	United States, Mexico, Spain, Malta	1
145	United States, Netherlands, Japan, France	1
146	United States, New Zealand	3
147	United States, New Zealand, Japan	1
148	United States, Russia	1
149	United States, South Korea, China	2
150	United States, Spain, Colombia, Mexico	1
151	United States, Sweden	1
152	United States, United Kingdom	7
153	United States, United Kingdom, Australia	1
154	United States, United Kingdom, Canada	1

```
#question 9
#Display the director name along with number of movies they have
produced.
 num_movies<- subset(Netflix_data,type="Movie")
 agg_movies<-
aggregate(num_movies$show_id~num_movies$director,
num_movies, length)
 colnames(agg_movies)= c("Director","No_Of_Movies")
 print(agg_movies)
```

```
A.R. Murugadoss
Ã“skar ThÃ³r Axelsson
Ã€lex Pastor, David Pastor
Ã‡agan Irmak
Ã\u0081lex de la Iglesia
Ã\u0081lvaro Brechner
Ã\u0081lvaro Longoria, Gerardo Olivares
Ã\u008dsold UggadÃ³ttir
Aadish Keluskar
Aamir Bashir
Aamir Khan
Aanand Rai
Aaron Burns
Aaron Hancox, Michael McNamara
Aaron Hann, Mario Miscione
Aaron Nee, Adam Nee
Aatmaram Dharne
Abba T. Makama
Abbas Alibhai Burmawalla, Mastan Alibhai Burmawalla
Abbas Tyrewala
Abby Epstein
Abdellatif Kechiche
Abdul Aziz Hashad
Abel Ferrara
Abhay Chopra
Abhijit Kokate, Srivinay Salian
Abhinay Deo
Abhishek Chaubey
Abhishek Kapoor
Abhishek Saxena
Abhishek Sharma
Abhishek Varman
Abu Bakr Shawky
Adam Alleca
Adam Bhala Lough
Adam Collins, Luke Radford
Adam Davis, Jerry Kolber, Trey Nelson, Erich Sturm
Adam Del Giudice
Adam Deyoe
```

Adam Marino
Adam McKay
Adam Nimoy
Adam Randall
Adam Shankman
Adam SjÃ¶berg
Adam Smith
Adam Wingard
Adam Wood
Adarsh Eshwarappa
Adele K. Thomas, Richard Bailey
Adisorn Tresirikasem
Aditya Kripalani
Aditya Sarpotdar
Aditya Vikram Sengupta
Adrian Murray
Adrian Teh
Advait Chandan
Adze Ugah
Afia Nathaniel
Afonso Poyart
Agasyah Karim, Khalid Kashogi
Agnidev Chatterjee
AgustÃ Villaronga
Ah Loong
Ahishor Solomon
Ahmad El-Badri
Ahmad Samir Farag
Ahmed Al-Badry
Ahmed El Gendy
Ahmed Khaled Moussa
Ahmed Zain
Ahmet KatÄ±ksÄ±z
Ahsan Rahim
Aijaz Khan
Aitor Arregi, Jon GaraÃ±o
Ajay Bahl
Ajay Bhuyan, Kunal Kohli

Alastair Fothergill
Albert Hughes, Allen Hughes
Albert Sharpe
Alberto Arnaut Estrada
Alberto RodrÃguez
Alejandra MÃ¡rquez Abella
Alejandro Agresti
Alejandro Doria
Alejandro FernÃ¡ndez Almendras
Alejandro G. IÃ±Ã¡rritu
Alejandro Lozano
Alejandro Montiel
Aleksandr Chernyaev, Fedor Lyass
Aleksey German
Alessandro Angulo
Alessandro Pepe
Alessio Cremonini
Alethea Jones
Alex Burunova
Alex Coletti
Alex DÃaz, Marcos Bucay
Alex Garland
Alex Holmes
Alex Infascelli
Alex Israel
Alex Law
Alex Lehmann
Alex Merkin
Alex Parkinson, Richard da Costa
Alex Proyas
Alex Richanbach
Alex Smith, Andrew J. Smith
Alex Stapleton
Alex Timbers
Alex Winter
Alex Zamm
Alexander Nevsky
Alexandra Dean
Alexandre Aja

```
465                                    Brian Smith
466                                     Brian Smrz
467                               Brian Volk-Weiss
468                                    Brie Larson
469                                   Brit McAdams
470                              Brittany Andrews
471                             Brodje Wemboendja
472                                Bronwen Hughes
473                               Bruce Beresford
474                                  Bruce Gowers
475                                   Bruce Leddy
476              Bruce MacDonald, Gabriel Sabloff
477                               Bruce McCulloch
478                                Bruce McDonald
479                                Bruce Robinson
480                                Bruce W. Smith
481                                 Bruno Garotti
482                                 Bryan Bertino
483                                 Bryan Buckley
484                                   Bryan Fogel
485                                  Bryan Singer
486                                 Bryce Wagoner
487                             Buddhadev Dasgupta
488                                         Bumpy
489                                Bunmi Ajakaiye
490                                   Burak Aksak
491                                Burhan Qurbani
492                                    Buta Singh
493                                   Buz Wallick
494                                    Buzz Kulik
495                   Byron Howard, Chris Williams
496                                     Byun Hyuk
497                                       C. Fitz
498                                   C.J. Wallis
499   CÃ©lia Catunda, Kiko Mistrorigo, Rodrigo Eba
500                                    Caio Cobra
    No_Of_Movies
1              2
```

```
500
    No_Of_Movies
1              2
2              2
3              1
4              2
5              1
6              1
7              1
8              2
9              1
10             1
11             1
12             1
13             1
14             1
15             2
16             1
17             1
18             1
19             1
20             1
21             1
22             4
23             1
24             1
25             1
26             1
27             1
28             1
29             1
30             3
31             1
32             3
```

```
466            1
467            1
468            1
469            1
470            1
471            1
472            1
473            1
474            1
475            1
476            1
477            1
478            1
479            1
480            1
481            1
482            1
483            1
484            1
485            1
486            1
487            1
488            1
489            1
490            2
491            1
492            1
493            1
494            1
495            1
496            1
497            1
498            1
499            1
500            1
 [ reached 'max' / getOption("max.print") -- omitted 2801 rows ]
>
```

```
#question 10
#Display title and date_added for TV Shows that are listed in Reality TV.
 Tv_title<- subset(Netflix_data,Netflix_data$type=="TV Show"&
Netflix_data$listed_in=="Reality TV")
 print(Tv_title[,c(3,7)])
```

```
> #Display title and date_added for TV Shows that are listed in Reality TV.
>
>  Tv_title<- subset(Netflix_data,Netflix_data$type=="TV Show"& Netflix_data$listed_in=="Reality TV")
>  print(Tv_title[,c(3,7)])
                                 title          date_added
219                          Pawn Stars  September 15, 2019
224                       The Rap Game  September 15, 2019
260      Car Masters: Rust to Riches  September 14, 2018
423                       Rhythm + Flow     October 9, 2019
447                       Dancing Queen     October 5, 2018
592                       Border Patrol     October 2, 2019
597                    Ghost Town Gold     October 2, 2019
600      License to Drill: Louisiana     October 2, 2019
907                             Strong     October 1, 2018
1009                          Westside    November 9, 2018
1063                   Death by Magic   November 30, 2018
1084             Sugar Rush Christmas   November 29, 2019
1203                 The Final Table   November 20, 2018
1248                Shot in the Dark   November 17, 2017
1270 I'm with the Band: Nasty Cherry   November 15, 2019
1627                            Flinch         May 3, 2019
1804                          Jailbirds        May 10, 2019
1833             The Kindness Diaries         May 1, 2019
2095                   Selling Sunset      March 22, 2019
2567                  Cooking on High       June 22, 2018
2659  Awake: The Million Dollar Game       June 14, 2019
3013                     MegaTruckers       July 12, 2019
3573                       The Circle     January 1, 2020
3635     Tidying Up with Marie Kondo     January 1, 2019
4249                       For the Win   December 31, 2018
4252                       Inst@famous   December 31, 2018
4259                    Slobby's World   December 31, 2018
4512                         Emogenius   December 15, 2018
4516                  Minute to Win It   December 15, 2018
4522          Skin Wars: Fresh Paint   December 15, 2018
4525                         Winsanity   December 15, 2018
```

#question 10

#Display title and date_added for TV Shows that are listed in Reality TV.

```
Tv_title<- subset(Netflix_data,Netflix_data$type=="TV Show"&
Netflix_data$listed_in=="Reality TV")
print(Tv_title[,c(3,7)])
```

```
1248                         Shot in the Dark   November 17, 2017
1270 I'm with the Band: Nasty Cherry   November 15, 2019
1627                                   Flinch        May 3, 2019
1804                                Jailbirds       May 10, 2019
1833                    The Kindness Diaries        May 1, 2019
2095                           Selling Sunset     March 22, 2019
2567                          Cooking on High      June 22, 2018
2659      Awake: The Million Dollar Game      June 14, 2019
3013                             MegaTruckers      July 12, 2019
3573                               The Circle    January 1, 2020
3635             Tidying Up with Marie Kondo    January 1, 2019
4249                              For the Win  December 31, 2018
4252                             Inst@famous  December 31, 2018
4259                           Slobby's World  December 31, 2018
4512                                Emogenius  December 15, 2018
4516                          Minute to Win It  December 15, 2018
4522                   Skin Wars: Fresh Paint  December 15, 2018
4525                                Winsanity  December 15, 2018
4729                        Tiny House Nation     August 9, 2019
4805                        Styling Hollywood    August 30, 2019
4882                               Hyperdrive    August 21, 2019
4942                          Save Our Shelter    August 18, 2018
4949                                Stay Here    August 17, 2018
5612                              Fastest Car September 20, 2019
5657                                Jeopardy!   October 28, 2019
5707                                Skin Wars    October 1, 2016
5735                      Nailed It! Holiday! November 22, 2019
5760                             Bondi Rescue   November 1, 2019
5766             The Boulet Brothers Dragula   November 1, 2019
5811                                Nailed It       May 17, 2019
5857                         Outback Truckers     March 17, 2017
5942                               Sugar Rush      July 26, 2019
5947                       Fix It and Finish It      July 20, 2018
5955                                Queer Eye      July 19, 2019
6074                                Top Grier  December 31, 2018
6077                        Highway Thru Hell   December 3, 2019
6138                     Ultimate Beastmaster    August 31, 2018
6145              Droppin' Cash: Los Angeles    August 28, 2019
>
```

```
#question 11
#Display the show_id and the title of Movie/TV Show released in the year 2020.
 b1<- Netflix_data[Netflix_data$release_year==2020,c("show_id","title")]
 print(b1)
```

```
> #Display the show_id and the title of Movie/TV Show released in the year 2021.
>  b1<- Netflix_data[Netflix_data$release_year==2020,c("show_id","title")]
>  print(b1)
      show_id                                                 title
1316 81034946                                  Maradona in Mexico
3180 81214114                                       Bulletproof 2
3189 81039393                                               Cheer
3195 80233408                                Live Twice, Love Once
3220 80997687                                             Dracula
3221 80237347                                Go! Go! Cory Carson
3249 81006825                        All the Freckles in the World
3325 81160763                                       Sex, Explained
3352 81127902                                 A Fall from Grace
3353 80995039                                               Ares
3354 81062580                                  Nailed It! Germany
3363 80996973                                   Handsome Siblings
3379 81062828          Killer Inside: The Mind of Aaron Hernandez
3426 80221553                     Kipo and the Age of Wonderbeasts
3427 81060049                           Leslie Jones: Time Machine
3436 80239306                         The Healing Powers of Dude
3464 80237329                                  AJ and the Queen
3466 81183491                       Jamtara - Sabka Number Ayega
3467 81011449                                  Medical Police
3472 81074060                                      Until Dawn
3518 81088083                                   Ghost Stories
3541 80117557                                          Messiah
3546 80197991 Nisman: The Prosecutor, the President, and the Spy
3562 80201590                                      Spinning Out
3573 81044551                                        The Circle
>
```

4

SOURCE CODE

```
Netflix_data = read.csv("netflix_titles.csv", na.strings =
c("","NA"), stringsAsFactors = FALSE)
print(class(Netflix_data))

#print(Netflix_data)

print(str(Netflix_data))

print(summary((Netflix_data)))
print(dim(Netflix_data))
```

```
library("plotly")
Netflix_data$rating<-
as.factor(Netflix_data$rating)

#printing missing values by creating a new data
frame.
a<-
data.frame("Variable"=c(colnames(Netflix_data)),
"Missing Values"=sapply(Netflix_data, function(x)
sum(is.na(x))), row.names = NULL)
print(a)
mode<- function(v){
uniqv<- unique(v)
uniqv[which.max(tabulate(match(v,uniqv)))]
}
```

```
Netflix_data$rating[is.na(Netflix_data$rating)] =
mode(Netflix_data$rating)
b<-
data.frame("Variable"=c(colnames(Netflix_data)),
"Missing Values"=sapply(Netflix_data, function(x)
sum(is.na(x))), row.names = NULL)
print(b)

library("dplyr")
Netflix_data= distinct(Netflix_data, title, country,
type, release_year, .keep_all = TRUE)
View(Netflix_data)
```

```
#question 1
library("plotly")
library("dplyr")
content_by_type <- Netflix_data %>%
group_by(type) %>%
summarise(count = n())

plot_ly(content_by_type, labels = ~type, values =
~count,
type = 'pie', marker = list(colors =
c("#bd3939", "#399ba3"))) %>%
layout(title = "Proportion of Content by
Type",legend = list(x = 200, y = 1))
```

```
#question2
library("plotly")
library("dplyr")
df_by_date <- Netflix_data %>% group_by(date_added,
type) %>%
summarise(added_today = n()) %>%
group_by(type)
plot_ly(full_data, x = ~date_added, y =
~total_number_of_content,
mode = 'lines', type = 'scatter',
color = ~type, colors = c("#bd3939",  "#9addbd",
"#399ba3")) %>%
layout(yaxis = list(title = 'Count'),
xaxis = list(title = 'Date'),
title = "Growth in Content over the Years", margin
= list(t = 54),
legend = list(x = 100, y = 0.5))
```

```
#question3
library("plotly")
library("dplyr")
df_by_rating <- Netflix_data %>%
group_by(rating) %>%
summarise(count = n())

plot_ly(df_by_rating, type = 'pie',
labels = ~rating, values = ~count) %>%
layout(title = "Distribution of Content by Rating",
legend = list(x = 100, y = 0.5))
```

```
#question4
library("plotly")
library("dplyr")
s_genres <- strsplit(Netflix_data$listed_in, split
= ", ")
df_by_listed_in <-  group_by(type, listed_in) %>%
summarise(count = n())

plot_ly(df_by_listed_in, x = ~listed_in, y =
~count,
type = 'bar', color = ~type,
colors = c("#bd3939", "#399ba3")) %>%
layout(xaxis = list(title = 'Genre'),
yaxis = list(title = 'Count'),
title = "Top Genres (Movie vs. TV Show)",
legend = list(x = 100, y = 0.5))
```

```
#question5
library("plotly")
library("dplyr")
duration_full_subset <-
duration_full[duration_full$country %in%
                c("United States", "India",
"United Kingdom",
                "Canada", "France", "Japan",
"Spain", "South Korea",
                "Mexico", "Australia", "China",
"Taiwan"),]
plot_ly(duration_full_subset, y = ~duration, color =
~country, type = "box") %>%
layout(xaxis = list(title = "Country"),
yaxis = list(title = 'Duration (in min)'),
title = "Movie Duration in Top 12 Countries",
legend = list(x = 100, y = 0.5))
```

```
#Access the title of first 20 Movie/ TV Shows of data
set.
head_movieshows<- head(Netflix_data$title,20)
print("Title of first 20 Movies and Tv shows: ")
print(head_movieshows)
#Find the total number of movies and TV shows in the
data set.
a1<- aggregate(Netflix_data$show_id ~
Netflix_data$type, Netflix_data, length)
colnames(a1)= c("type","length")
print(a1)
#Display the show_id and the title of Movie/TV Show
released in the year 2020.
b1<-
Netflix_data[Netflix_data$release_year==2020,c("sho
w_id","title")]
print(b1)
```

```
#Display the number of TV Shows Produced by each country.
e<- subset(Netflix_data, type=="TV Show")
f<- aggregate(e$show_id~e$country, e, length)
colnames(f)= c("Country","Tv_Shows_Produced")
print(f)
#Display the director name along with number of movies
they have produced.
num_movies<- subset(Netflix_data,type="Movie")
agg_movies<-
aggregate(num_movies$show_id~num_movies$director,
num_movies, length)
colnames(agg_movies)= c("Director","No_Of_Movies")
print(agg_movies)
#Display title and date_added for TV Shows that are listed
in Reality TV.
Tv_title<- subset(Netflix_data,Netflix_data$type=="TV
Show"& Netflix_data$listed_in=="Reality TV")
print(Tv_title[,c(3,7)])
```