

# DATA SCIENCE DASHBOARD



## NETFLIX DATA ANALYSIS

### **Submitted By:**

Vishalakshi  
102017189

### **Submitted To:**

Ms. Kashish Goyal

July 2022 – December 2022

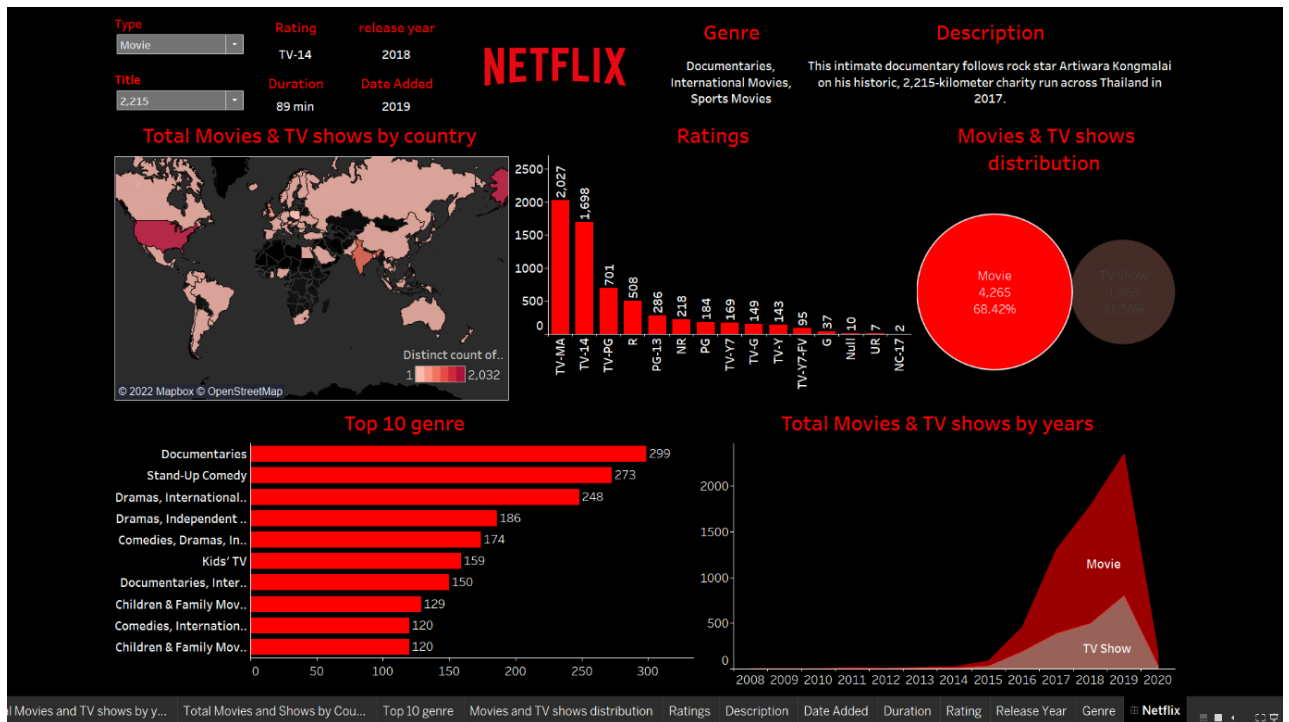
## **OVERVIEW OF PROJECT**

In this project, we worked on data analysis and for that, we looked into the Netflix dataset.

From that dataset we derived various insights that helped us know about the weightage of each feature and how they are interrelated to each other.

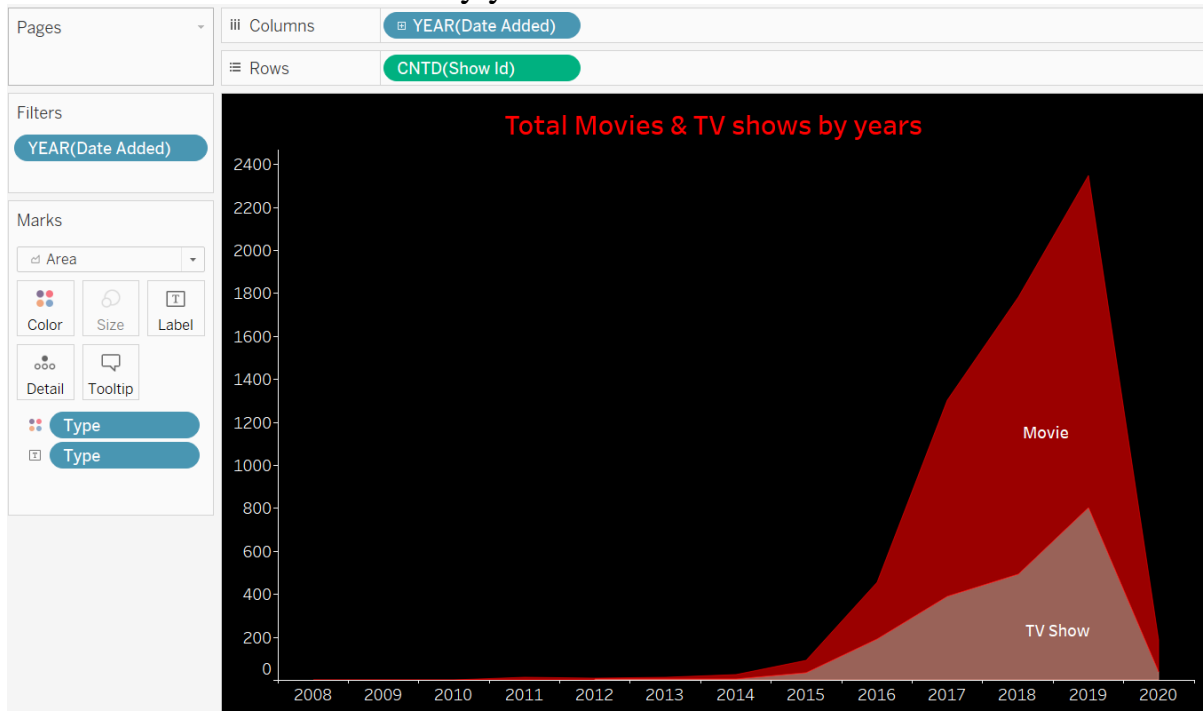
We used R Programming to clean our dataset and plotted graphs using tableau.

# DASHBOARD



## SHEET1

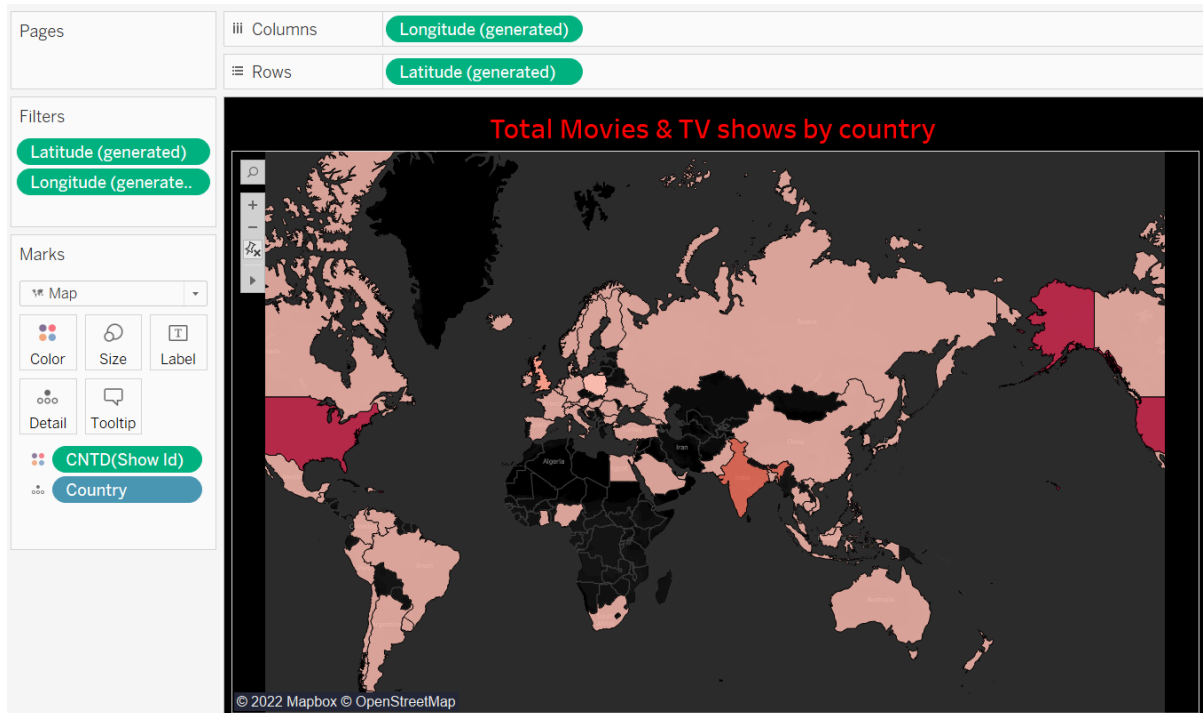
### Total Movies and TV shows by years



Inference: Rate of addition of movies comes out to be approximately twice to that of TV shows.

## SHEET2

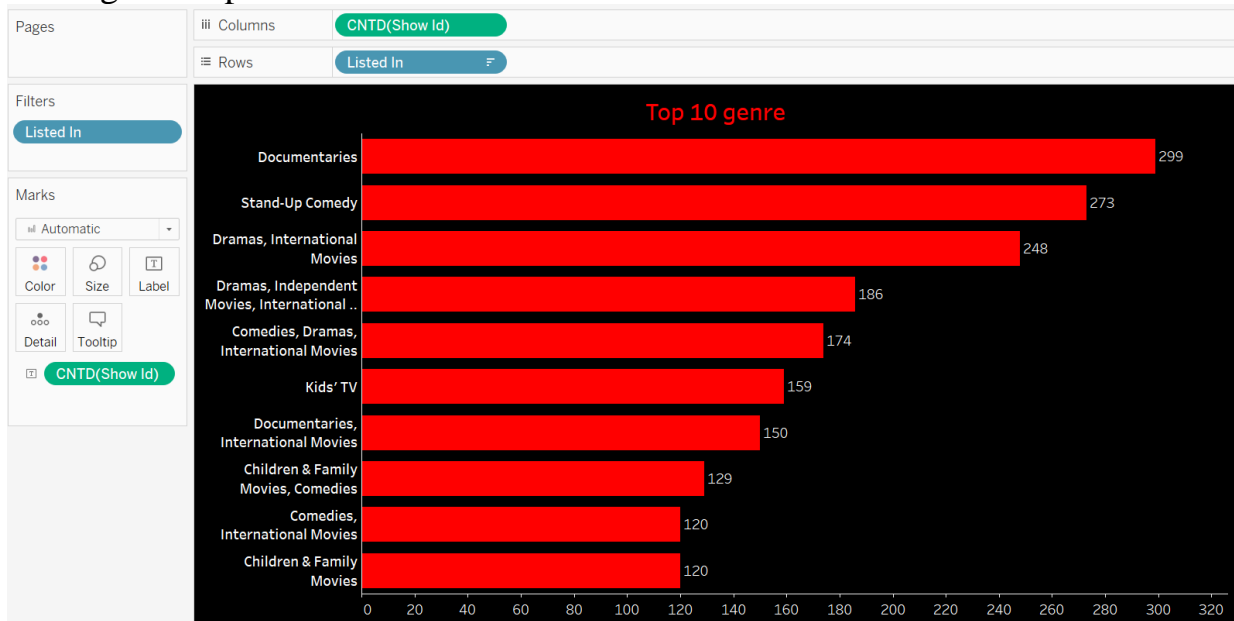
### Distribution of Movies and TV shows across the world



Inference: United States stands at the top with India behind and some countries have no contribution.

## SHEET3

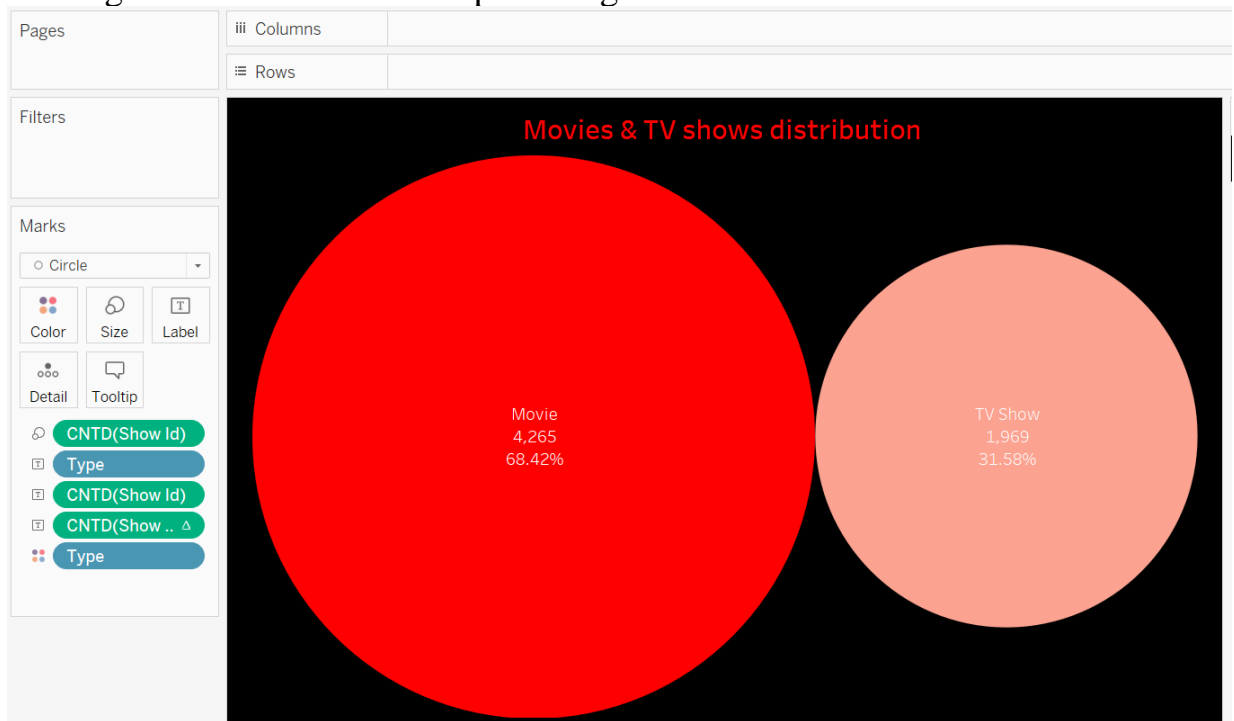
### Getting the Top 10 Genre



Inference: Documentaries and Stand up comedy are the top most genre provided by Netflix.

## SHEET4

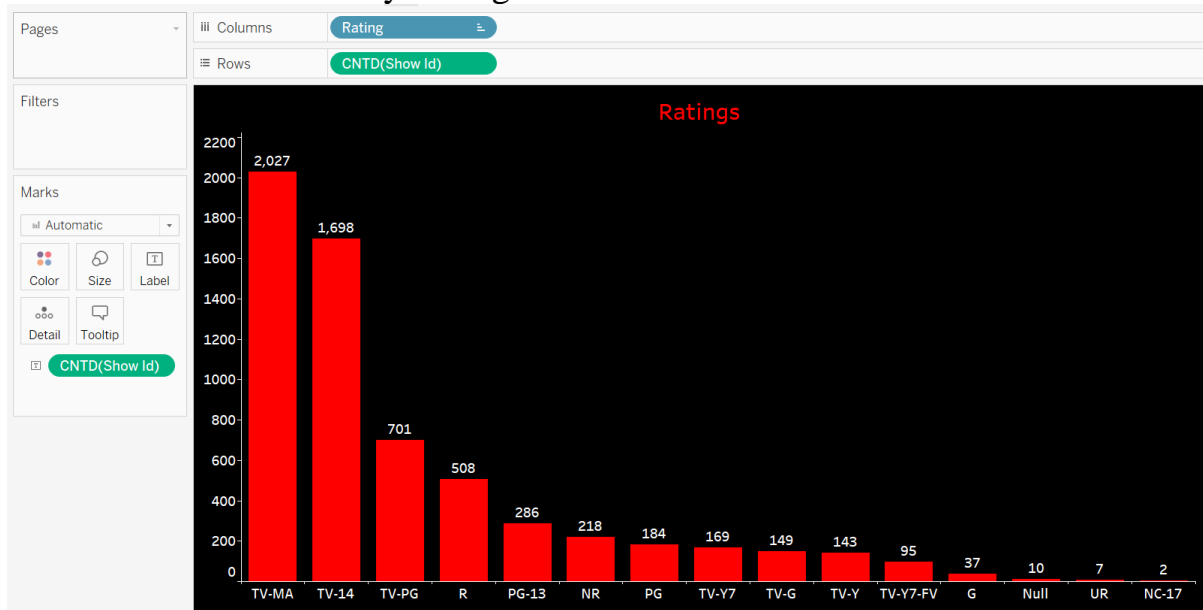
### Finding the total count and the percentage of Movies and TV Shows on Netflix.



Inference: Netflix has a higher proportion of movies as compared to TV Shows .

## SHEET5

### Distribution of Content by Rating

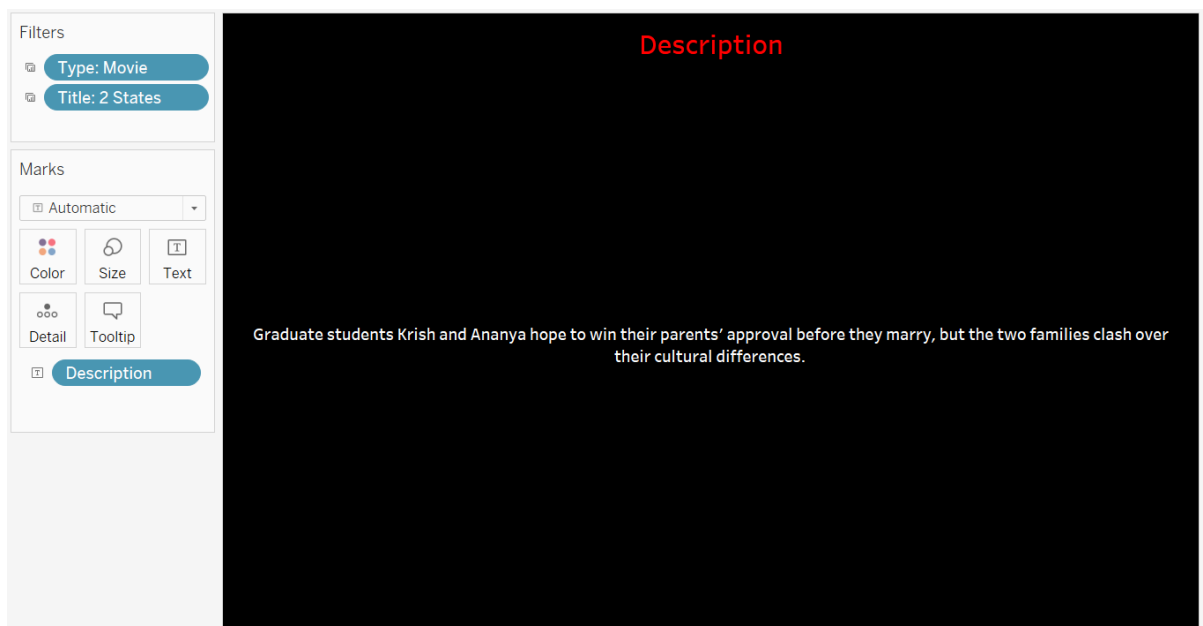


Inference: Most of the content is rated TV-MA that is for Mature Audience.

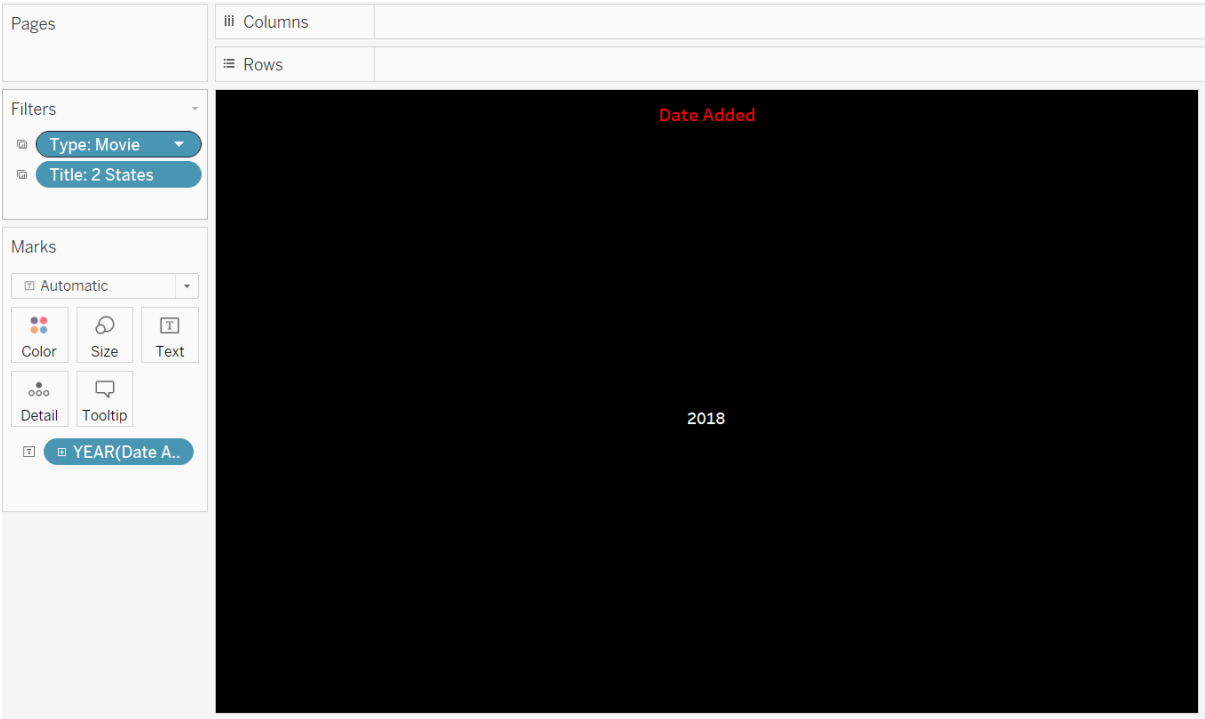
#Sheet 6-11 are used at the top of Dashboard

They show data according to the TYPE and Title specified in the drop down box. Use of filter is done here.

## SHEET6



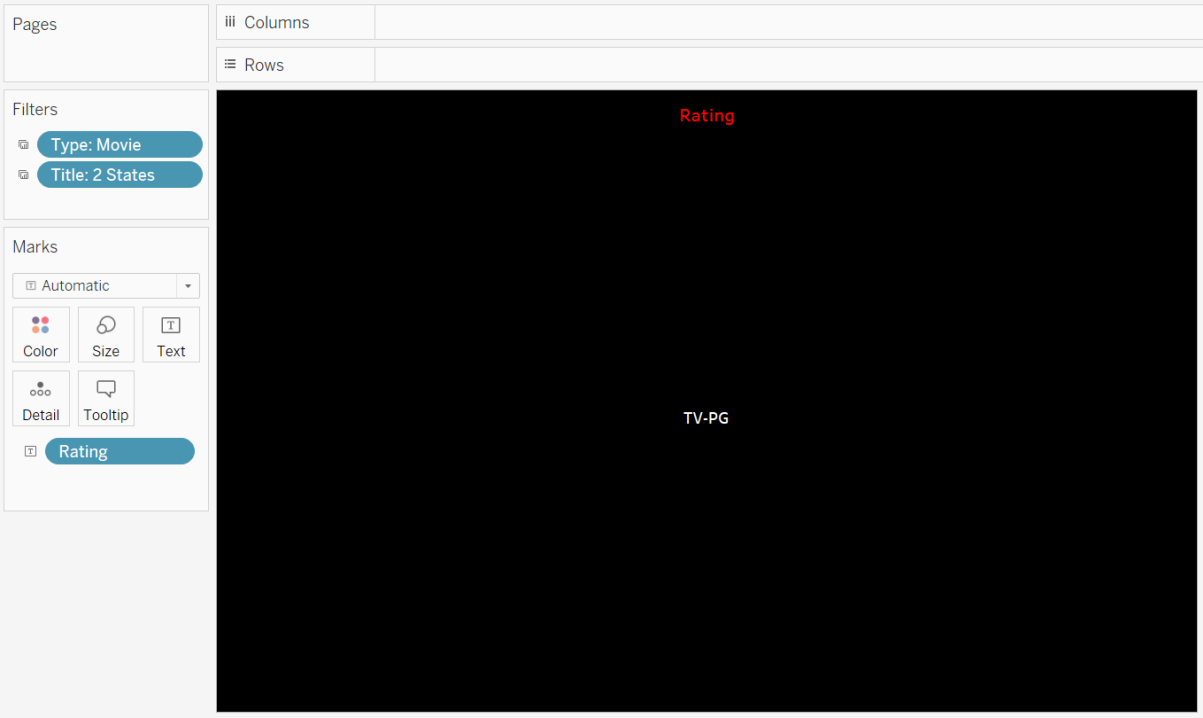
# SHEET7



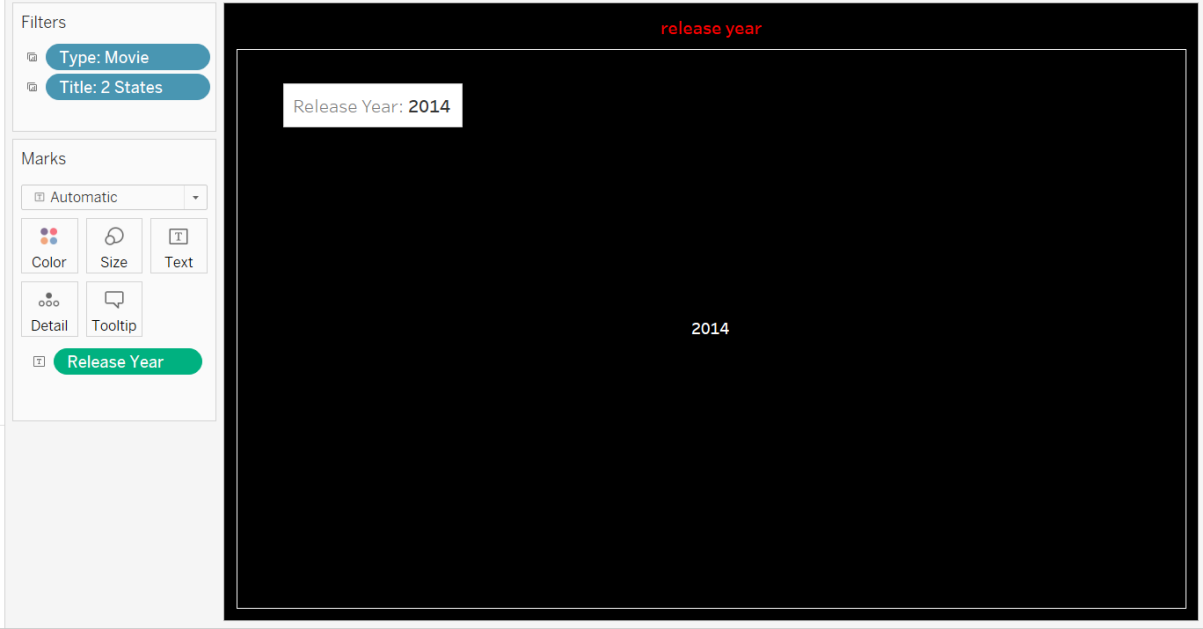
# SHEET8



# SHEET9



# SHEET10



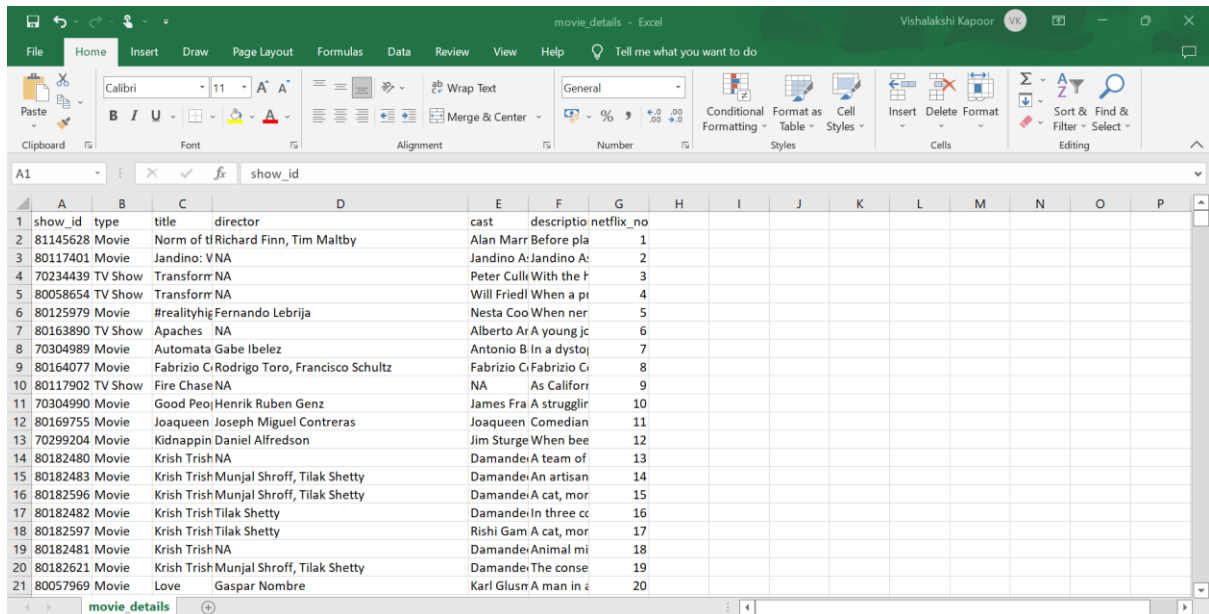


# SHEET11



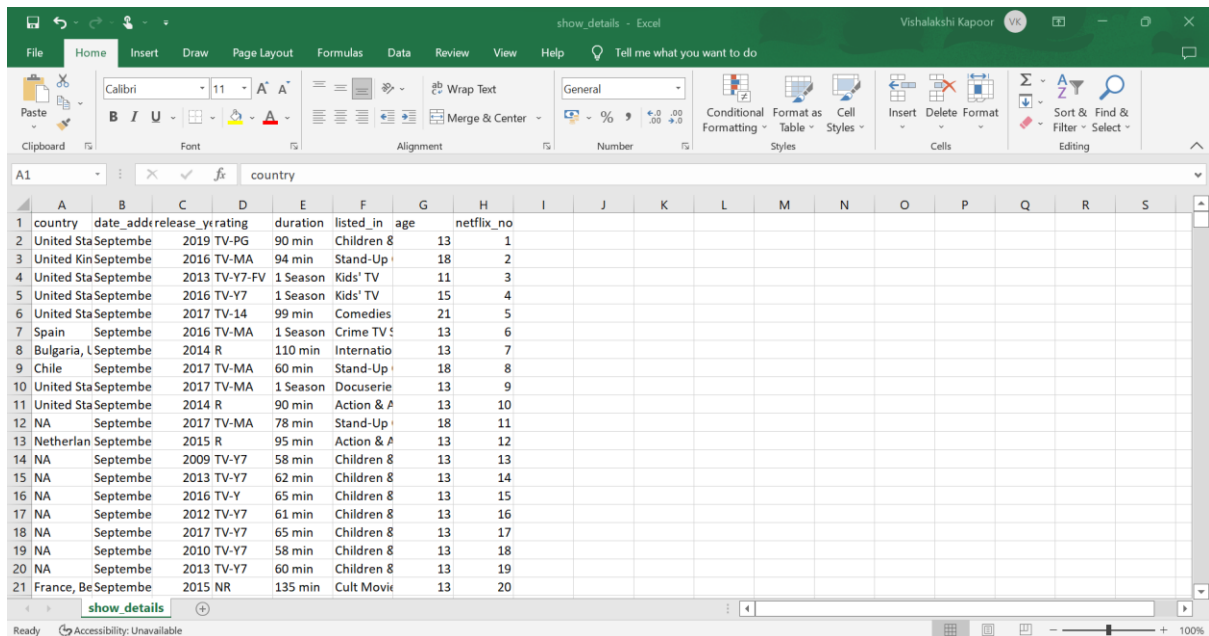
# DATA BEFORE PRE-PROCESSING

## DATA SET-1 Movie\_details



show_id	type	title	director	cast	description	netflix_no
81145628	Movie	Norm of the North	Richard Finn, Tim Maltby	Alan Marr	Before the	1
80117401	Movie	Jandino V	NA	Jandino A	Jandino A	2
70234439	TV Show	Transformers	NA	Peter Cullen	With the	3
80058654	TV Show	Transformers	NA	Will Friedle	When a	4
80125979	Movie	#realityhigh	Fernando Lebrija	Nesta Coo	When ner	5
80163890	TV Show	Apaches	NA	Alberto Ar	A young	6
70304989	Movie	Automata	Gabe Ibelez	Antonio B	In a dysto	7
80164077	Movie	Fabrizio C	Rodrigo Toro, Francisco Schultz	Fabrizio C	Fabrizio C	8
80117902	TV Show	Fire Chase	NA	As Califor		9
70304990	Movie	Good People	Henrik Ruben Genz	James Fra	A strugglir	10
80169755	Movie	Joaqueline	Joseph Miguel Contreras	Joaqueline	Comedian	11
70299204	Movie	Kidnapping	Daniel Alfrekson	Jim Sturge	When bee	12
80182480	Movie	Krish Trish	NA	Damande	A team of	13
80182483	Movie	Krish Trish	Munjal Shroff, Tilak Shetty	Damande	An artisan	14
80182596	Movie	Krish Trish	Munjal Shroff, Tilak Shetty	Damande	A cat, mor	15
80182482	Movie	Krish Trish	Tilak Shetty	Damande	In three cc	16
80182597	Movie	Krish Trish	Tilak Shetty	Rishi Gam	A cat, mor	17
80182481	Movie	Krish Trish	NA	Damande	Animal mi	18
80182621	Movie	Krish Trish	Munjal Shroff, Tilak Shetty	Damande	The conse	19
80057969	Movie	Love	Gaspar Nombre	Karl Glusn	A man in	20

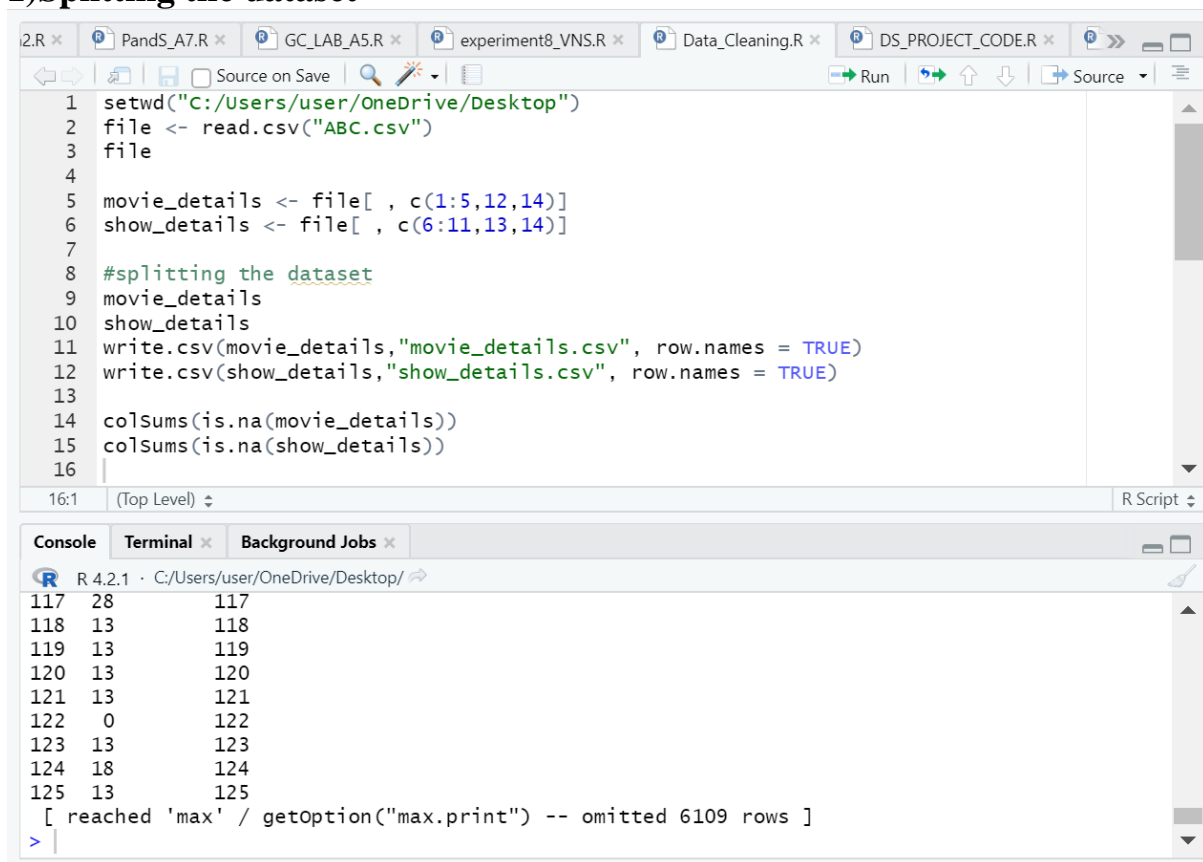
## DATA SET-2 Show details



country	date	address	release	year	rating	duration	listed in	age	netflix_no
United States	September		2019	TV-PG	90 min	Children &		13	1
United Kingdom	September		2016	TV-MA	94 min	Stand-Up		18	2
United States	September		2013	TV-Y7-FV	1 Season	Kids' TV		11	3
United States	September		2016	TV-Y7	1 Season	Kids' TV		15	4
United States	September		2017	TV-14	99 min	Comedies		21	5
Spain	September		2016	TV-MA	1 Season	Crime TV		13	6
Bulgaria, L	September		2014	R	110 min	Internatio		13	7
Chile	September		2017	TV-MA	60 min	Stand-Up		18	8
United States	September		2017	TV-MA	1 Season	Docuserie		13	9
United States	September		2014	R	90 min	Action & A		13	10
NA	September		2017	TV-MA	78 min	Stand-Up		18	11
Netherlands	September		2015	R	95 min	Action & A		13	12
NA	September		2009	TV-Y7	58 min	Children &		13	13
NA	September		2013	TV-Y7	62 min	Children &		13	14
NA	September		2016	TV-Y	65 min	Children &		13	15
NA	September		2012	TV-Y7	61 min	Children &		13	16
NA	September		2017	TV-Y7	65 min	Children &		13	17
NA	September		2010	TV-Y7	58 min	Children &		13	18
NA	September		2013	TV-Y7	60 min	Children &		13	19
France, Be	September		2015	NR	135 min	Cult Movie		13	20

# DATA CLEANING

## 1) Splitting the dataset



The screenshot displays the RStudio environment with several open scripts. The active script, 'Data\_Cleaning.R', contains the following R code:

```
1 setwd("C:/Users/user/OneDrive/Desktop")
2 file <- read.csv("ABC.csv")
3 file
4
5 movie_details <- file[, c(1:5,12,14)]
6 show_details <- file[, c(6:11,13,14)]
7
8 #splitting the dataset
9 movie_details
10 show_details
11 write.csv(movie_details,"movie_details.csv", row.names = TRUE)
12 write.csv(show_details,"show_details.csv", row.names = TRUE)
13
14 colSums(is.na(movie_details))
15 colSums(is.na(show_details))
16
```

The console output shows the result of the column sums for missing values:

```
117      28      117
118     13      118
119     13      119
120     13      120
121     13      121
122      0      122
123     13      123
124     18      124
125     13      125
[ reached 'max' / getOption("max.print") -- omitted 6109 rows ]
>
```

## 2)finding NULL Values

```
12.R x PandS_A7.R x GC_LAB_A5.R x experiment8_VNS.R x Data_Cleaning.R x DS_PROJECT_CODER x
Source on Save Run
10 show_details
11 write.csv(movie_details,"movie_details.csv", row.names = TRUE)
12 write.csv(show_details,"show_details.csv", row.names = TRUE)
13
14 colSums(is.na(movie_details))
15 colSums(is.na(show_details))
16
17
18 # Replacing null age values with mean age values
19 age = complete.cases(show_details$age)
20 mean_age = mean(age)
21 show_details$age[is.na(show_details$age)] <- mean_age
22
23
14:1 (Top Level) R Script
```

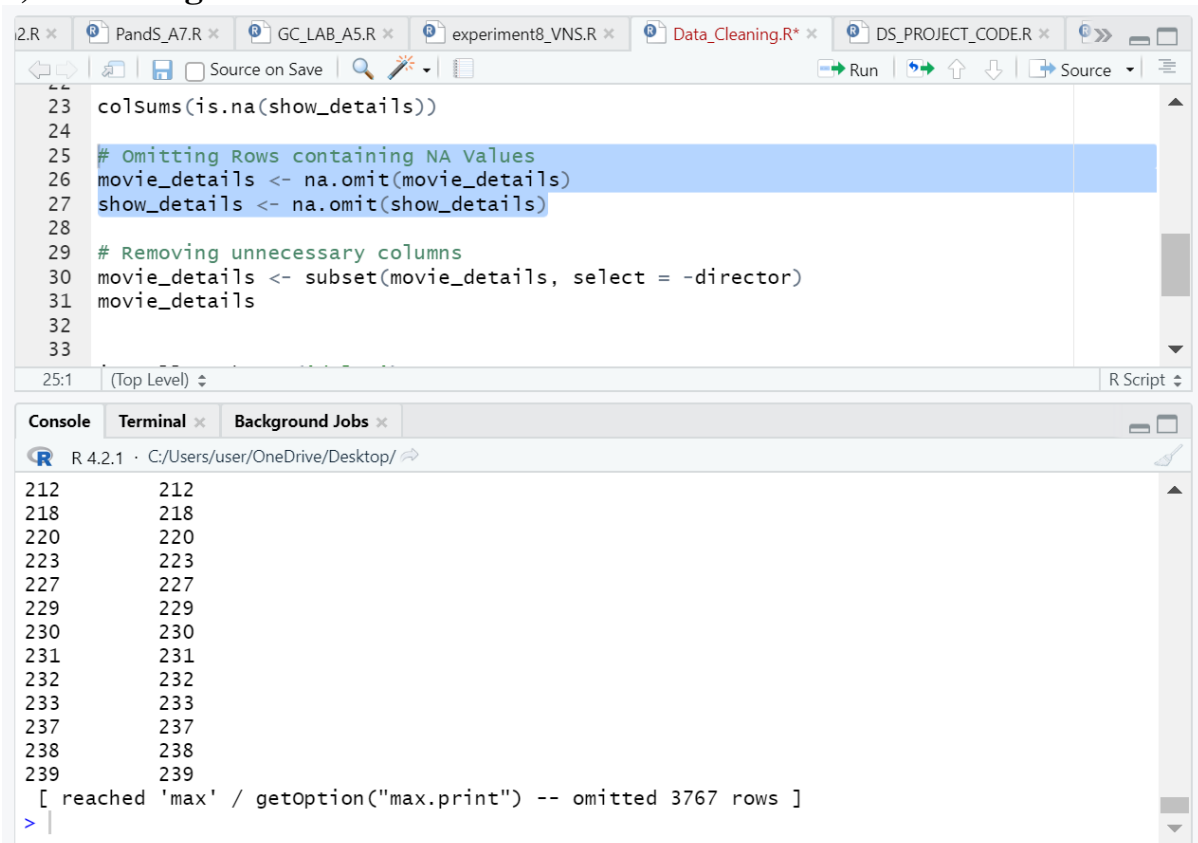
```
Console Terminal Background Jobs
R 4.2.1 C:/Users/user/OneDrive/Desktop/
123 13 123
124 18 124
125 13 125
[ reached 'max' / getOption("max.print") -- omitted 6109 rows ]
> colSums(is.na(movie_details))
  show_id      type      title      director      cast description  netflix_no
      0         0         0         1969         570         0         0
> colSums(is.na(show_details))
  country  date_added release_year      rating      duration  listed_in
    476         11         0         10         0         0
  age  netflix_no
    265         0
> |
```

## 3)Replacing NULL values with mean in age column of show details dataset

```
12.R x PandS_A7.R x GC_LAB_A5.R x experiment8_VNS.R x Data_Cleaning.R* x DS_PROJECT_CODER x
Source on Save Run
13
14 colSums(is.na(movie_details))
15 colSums(is.na(show_details))
16
17
18 # Replacing null age values with mean age values
19 age = complete.cases(show_details$age)
20 mean_age = mean(age)
21 show_details$age[is.na(show_details$age)] <- mean_age
22
23 colSums(is.na(show_details))
24
25:1 (Top Level) R Script
```

```
Console Terminal Background Jobs
R 4.2.1 C:/Users/user/OneDrive/Desktop/
> colSums(is.na(show_details))
  country  date_added release_year      rating      duration  listed_in
    476         11         0         10         0         0
  age  netflix_no
    265         0
> # Replacing null age values with mean age values
> age = complete.cases(show_details$age)
> mean_age = mean(age)
> show_details$age[is.na(show_details$age)] <- mean_age
> colSums(is.na(show_details))
  country  date_added release_year      rating      duration  listed_in
    476         11         0         10         0         0
  age  netflix_no
    0         0
> |
```

#### 4) Removing rows with NULL values



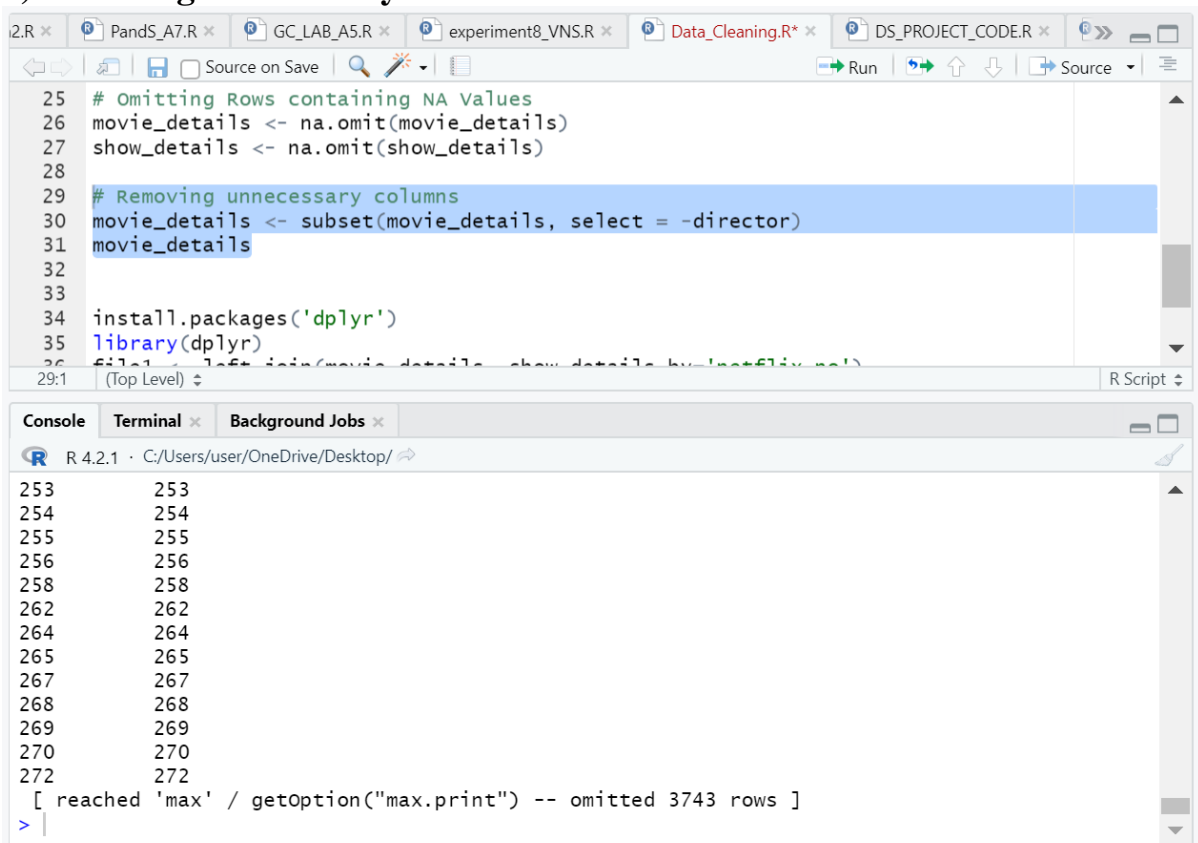
The screenshot shows the RStudio interface with the following code in the script editor:

```
23 colSums(is.na(show_details))
24
25 # Omitting Rows containing NA Values
26 movie_details <- na.omit(movie_details)
27 show_details <- na.omit(show_details)
28
29 # Removing unnecessary columns
30 movie_details <- subset(movie_details, select = -director)
31 movie_details
32
33
```

The console output shows the execution of the code, with the following visible text:

```
212      212
218      218
220      220
223      223
227      227
229      229
230      230
231      231
232      232
233      233
237      237
238      238
239      239
[ reached 'max' / getOption("max.print") -- omitted 3767 rows ]
>
```

#### 5) Removing unnecessary Column



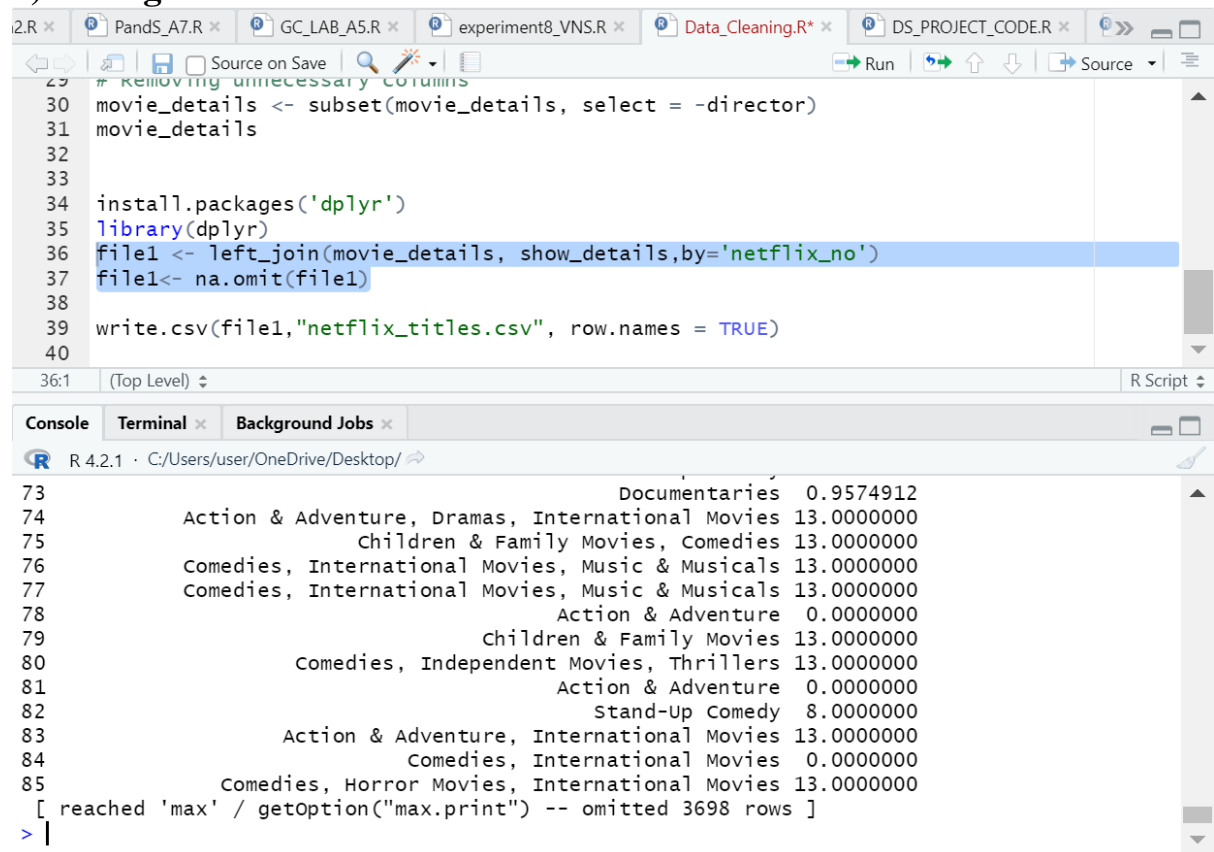
The screenshot shows the RStudio interface with the following code in the script editor:

```
25 # Omitting Rows containing NA Values
26 movie_details <- na.omit(movie_details)
27 show_details <- na.omit(show_details)
28
29 # Removing unnecessary columns
30 movie_details <- subset(movie_details, select = -director)
31 movie_details
32
33
34 install.packages('dplyr')
35 library(dplyr)
36 # Left join(movie_details, show_details, by = 'movie_id')
```

The console output shows the execution of the code, with the following visible text:

```
253      253
254      254
255      255
256      256
258      258
262      262
264      264
265      265
267      267
268      268
269      269
270      270
272      272
[ reached 'max' / getOption("max.print") -- omitted 3743 rows ]
>
```

## 6)Joining the 2 tables



The screenshot shows the RStudio environment with several open scripts. The active script is `Data_Cleaning.R`, which contains the following R code:

```
29 # Removing unnecessary columns
30 movie_details <- subset(movie_details, select = -director)
31 movie_details
32
33
34 install.packages('dplyr')
35 library(dplyr)
36 file1 <- left_join(movie_details, show_details, by='netflix_no')
37 file1 <- na.omit(file1)
38
39 write.csv(file1, "netflix_titles.csv", row.names = TRUE)
40
```

The console output shows the result of the `write.csv` function, displaying the first 10 rows of the joined data:

```
73 Documentaries 0.9574912
74 Action & Adventure, Dramas, International Movies 13.0000000
75 Children & Family Movies, Comedies 13.0000000
76 Comedies, International Movies, Music & Musicals 13.0000000
77 Comedies, International Movies, Music & Musicals 13.0000000
78 Action & Adventure 0.0000000
79 Children & Family Movies 13.0000000
80 Comedies, Independent Movies, Thrillers 13.0000000
81 Action & Adventure 0.0000000
82 Stand-Up Comedy 8.0000000
83 Action & Adventure, International Movies 13.0000000
84 Comedies, International Movies 0.0000000
85 Comedies, Horror Movies, International Movies 13.0000000
[ reached 'max' / getOption("max.print") -- omitted 3698 rows ]
> |
```

## DATA AFTER PRE-PROCESSING

O8														
	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	show_id	type	title	cast	description	netflix_no	country	date_added	release_year	rating	duration	listed_in	age	
2	81145628	Movie	Norm of the North	Alan Marr	Before playing	1	United States	September 1, 2019	2019	TV-PG	90 min	Children & Family	8	13
3	80125979	Movie	#realityhigh	Nesta Cooper	When nerds	5	United States	September 1, 2017	2017	TV-14	99 min	Comedies		21
4	70304989	Movie	Automata	Antonio B. DiNo	In a dystopian	7	Bulgaria, United States	September 1, 2014	2014	R	110 min	International		13
5	80164077	Movie	Fabrizio Craxi	Fabrizio Craxi	Fabrizio Craxi	8	Chile	September 1, 2017	2017	TV-MA	60 min	Stand-Up		18
6	70304990	Movie	Good People	James Franco	A struggling	10	United States	September 1, 2014	2014	R	90 min	Action & Adventure		13
7	70299204	Movie	Kidnapping	Jim Sturgis	When bees	12	Netherlands	September 1, 2015	2015	R	95 min	Action & Adventure		13
8	80057969	Movie	Love	Karl Glusman	A man in a	20	France, Belgium	September 1, 2015	2015	NR	135 min	Cult Movies		13
9	80060297	Movie	Manhattan	Tom O'Brien	A filmmaker	21	United States	September 1, 2014	2014	TV-14	98 min	Comedies		13
10	80046728	Movie	Moonwalk	Ron Perlman	A brain-acc	22	France, Belgium	September 1, 2015	2015	R	96 min	Action & Adventure		13
11	70304988	Movie	Stonehearst	Kate Beckinsale	In 1899, a	24	United States	September 1, 2014	2014	PG-13	113 min	Horror Movies		3
12	80057700	Movie	The Runner	Nicolas Cage	A New Orleans	25	United States	September 1, 2015	2015	R	90 min	Dramas, International		11
13	80045922	Movie	6 Years	Taissa Farinha	As a volatile	26	United States	September 1, 2015	2015	NR	80 min	Dramas, International		13
14	70241607	Movie	Laddaland	Saharath Sarath	When a fa	30	Thailand	September 1, 2011	2011	TV-MA	112 min	Horror Movies		13
15	80988892	Movie	Next Gen	John Krasinski	When lon	31	China, Canada	September 1, 2018	2018	TV-PG	106 min	Children & Family		13
16	80239639	Movie	Sierra Burgess	Shannon Fessler	A wrong-n	32	United States	September 1, 2018	2018	PG-13	106 min	Comedies		32
17	80159586	Movie	The Most	Anna Mouglalis	In 1930s P	33	Belgium, United States	September 1, 2018	2018	TV-MA	102 min	Dramas, International		13
18	80152447	Movie	Cathezar	Guillaume Canet	This histor	34	Belgium, France	September 1, 2016	2016	R	114 min	Dramas, International		13
19	81154455	Movie	Article 15	Ayushmar	The grim r	36	India	September 1, 2019	2019	TV-MA	125 min	Dramas, International		13
20	81052275	Movie	Ee Nagara	Vishwak	In Goa and	38	India	September 1, 2018	2018	TV-14	133 min	Comedies		50
21	80058026	Movie	Hell and B	Nick Swain	When bes	41	United States	September 1, 2015	2015	R	86 min	Action & Adventure		13

netflix\_titles

+

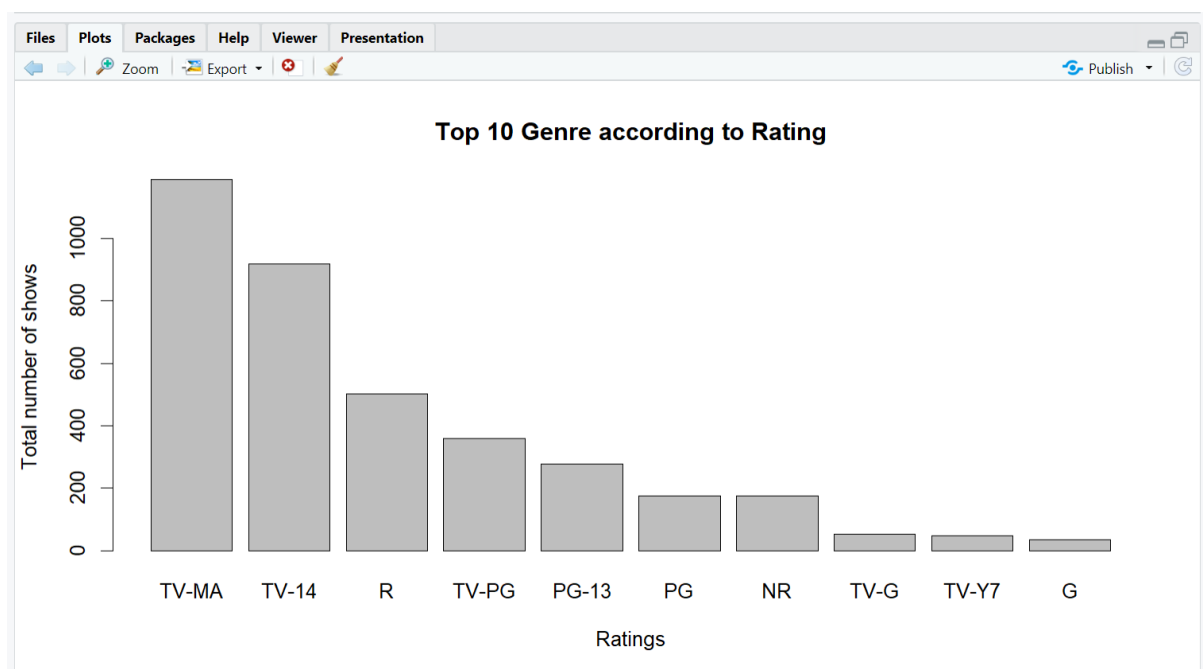
Ready Accessibility: Unavailable

# QUERIES

- Plotting count of Movies and Shows by Rating

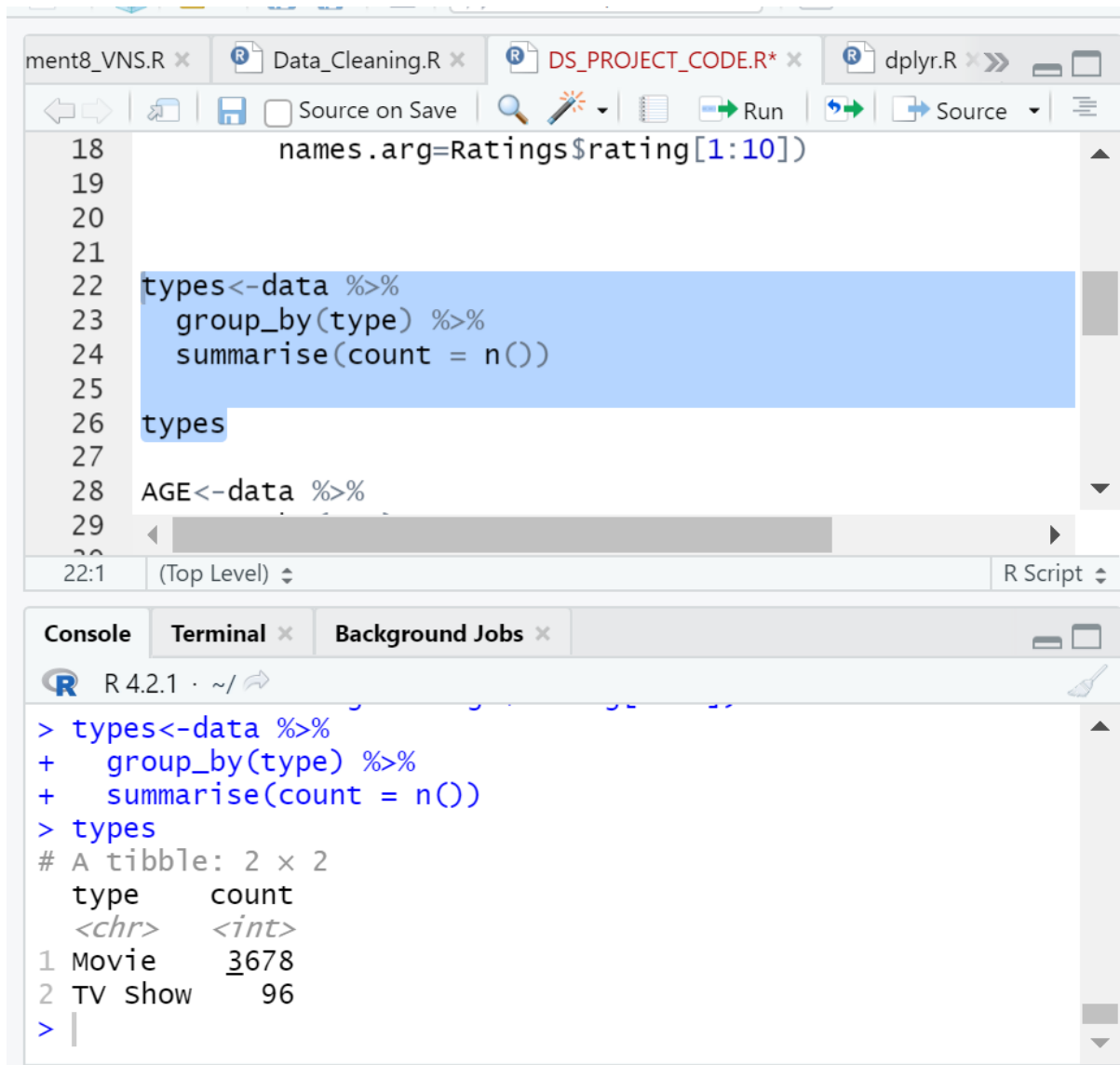
```
1 data<-read.csv("C:/Users/user/OneDrive/Desktop/netflix_titles.csv")
2
3 summary(data)
4
5 library(dplyr)
6 library(tibble)
7
8
9 Ratings<-data %>%
10   group_by(rating) %>%
11   summarise(count = n())
12
13 Ratings <- Ratings[order(-Ratings$count),]
14 barplot(Ratings$count[1:10],
15         main = "Top 10 Genre according to Rating",
16         xlab = "Ratings",
17         ylab = "Total number of shows",
18         names.arg=Ratings$rating[1:10])
19
```

```
> Ratings <- Ratings[order(-Ratings$count),]
> barplot(Ratings$count[1:10],
+         main = "Top 10 Genre according to Rating",
+         xlab = "Ratings",
+         ylab = "Total number of shows",
+         names.arg=Ratings$rating[1:10])
>
```





- Counting total number of Movies and Shows



The screenshot shows the RStudio environment. The top pane contains R code for data manipulation. The bottom pane shows the console output of the executed code.

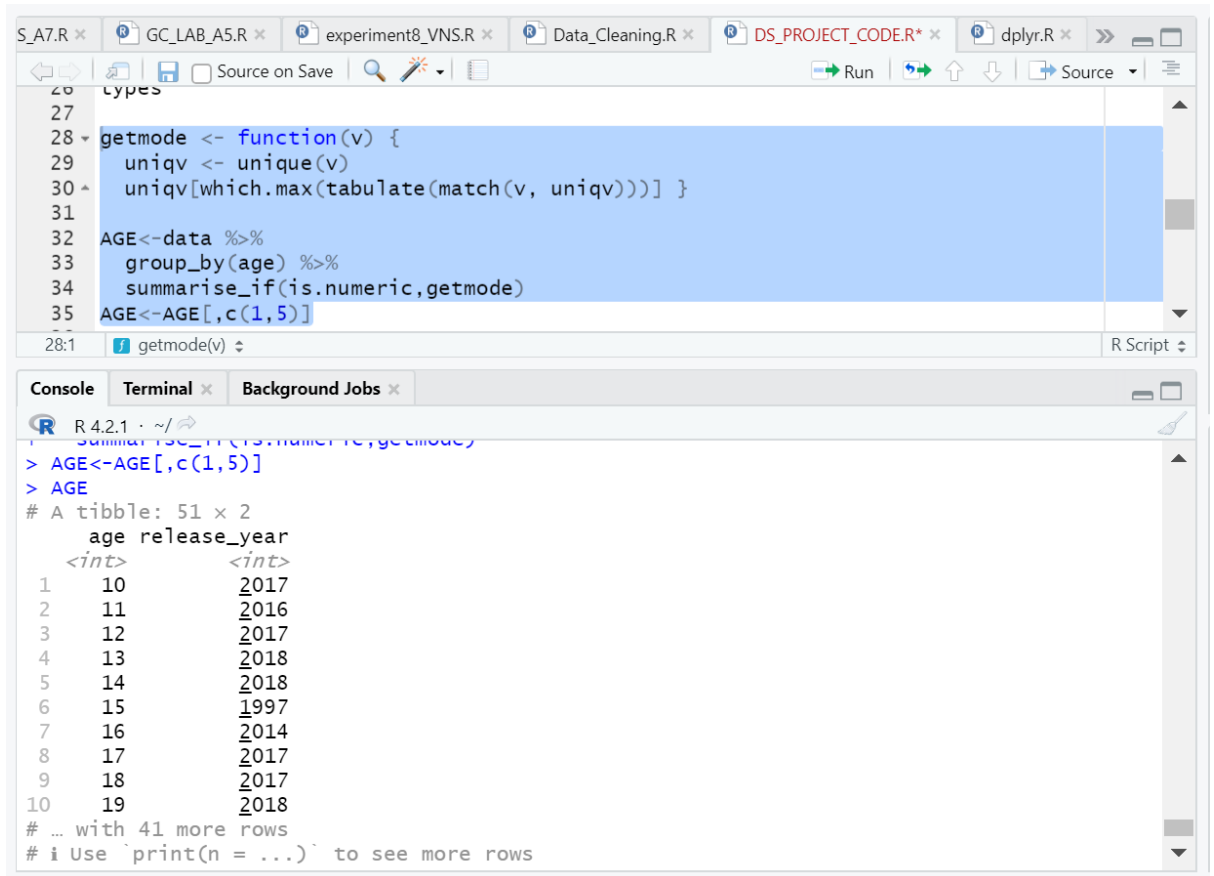
**Editor Code:**

```
18 names.arg=ratings$rating[1:10])
19
20
21
22 types<-data %>%
23   group_by(type) %>%
24   summarise(count = n())
25
26 types
27
28 AGE<-data %>%
29
```

**Console Output:**

```
> types<-data %>%
+   group_by(type) %>%
+   summarise(count = n())
> types
# A tibble: 2 x 2
  type      count
  <chr>    <int>
1 Movie     3678
2 TV show     96
> |
```

- Calculating year in which most movies were released for every age group



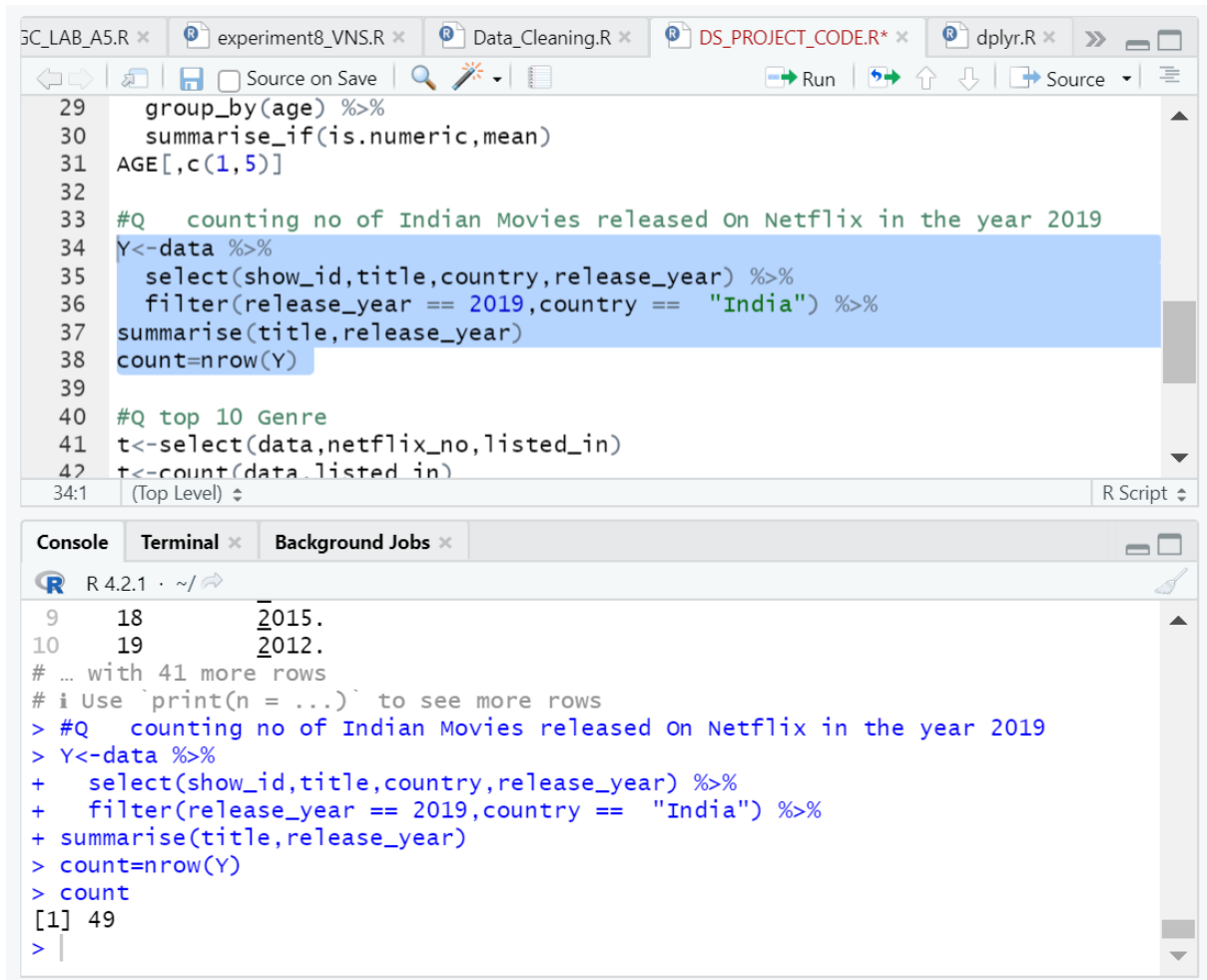
The screenshot shows the RStudio environment with several open scripts. The active script, `DS_PROJECT_CODE.R`, contains the following R code:

```
26 types
27
28 getmode <- function(v) {
29   uniqv <- unique(v)
30   uniqv[which.max(tabulate(match(v, uniqv)))] }
31
32 AGE<-data %>%
33   group_by(age) %>%
34   summarise_if(is.numeric,getmode)
35 AGE<-AGE[,c(1,5)]
```

The console output shows the execution of the code, resulting in a tibble with 51 rows and 2 columns: `age` and `release_year`.

```
R 4.2.1 ~ /
> summarise_if(is.numeric,getmode)
> AGE<-AGE[,c(1,5)]
> AGE
# A tibble: 51 x 2
   age release_year
<int>         <int>
1    10          2017
2    11          2016
3    12          2017
4    13          2018
5    14          2018
6    15          1997
7    16          2014
8    17          2017
9    18          2017
10   19          2018
# ... with 41 more rows
# i Use `print(n = ...)` to see more rows
```

- Counting number of Indian movies released On Netflix in the year 2019



The screenshot displays the R Studio environment. The top pane shows the script editor with R code. The bottom pane shows the console output.

**Script Editor Code:**

```
29 group_by(age) %>%
30 summarise_if(is.numeric, mean)
31 AGE[,c(1,5)]
32
33 #Q counting no of Indian Movies released On Netflix in the year 2019
34 Y<-data %>%
35   select(show_id,title,country,release_year) %>%
36   filter(release_year == 2019,country == "India") %>%
37   summarise(title,release_year)
38   count=nrow(Y)
39
40 #Q top 10 Genre
41 t<-select(data,netflix_no,listed_in)
42 t<-count(data_listed_in)
```

**Console Output:**

```
R 4.2.1 · ~/
9 18 2015.
10 19 2012.
# ... with 41 more rows
# i Use `print(n = ...)` to see more rows
> #Q counting no of Indian Movies released On Netflix in the year 2019
> Y<-data %>%
+   select(show_id,title,country,release_year) %>%
+   filter(release_year == 2019,country == "India") %>%
+   summarise(title,release_year)
> count=nrow(Y)
> count
[1] 49
>
```

- Top 10 Genre in Netflix

The screenshot shows an RStudio window with several tabs open: 'indS\_A7.R', 'GC\_LAB\_A5.R', 'experiment8\_VNS.R', 'Data\_Cleaning.R', 'DS\_PROJECT\_CODE.R', and 'dplyr.R'. The active script is 'DS\_PROJECT\_CODE.R', which contains the following R code:

```

38 count=nrow(Y)
39
40 #Q top 10 Genre
41 t<-select(data,netflix_no,listed_in)
42 t1<-count(data,listed_in)
43 s<-arrange(t1[1:10,],desc(n),listed_in)
44
45
46 #total movies and shows for kids(less than 18 years)
43:40 (Top Level)

```

The console output shows the execution of the code, resulting in a table of the top 10 genres and their counts:

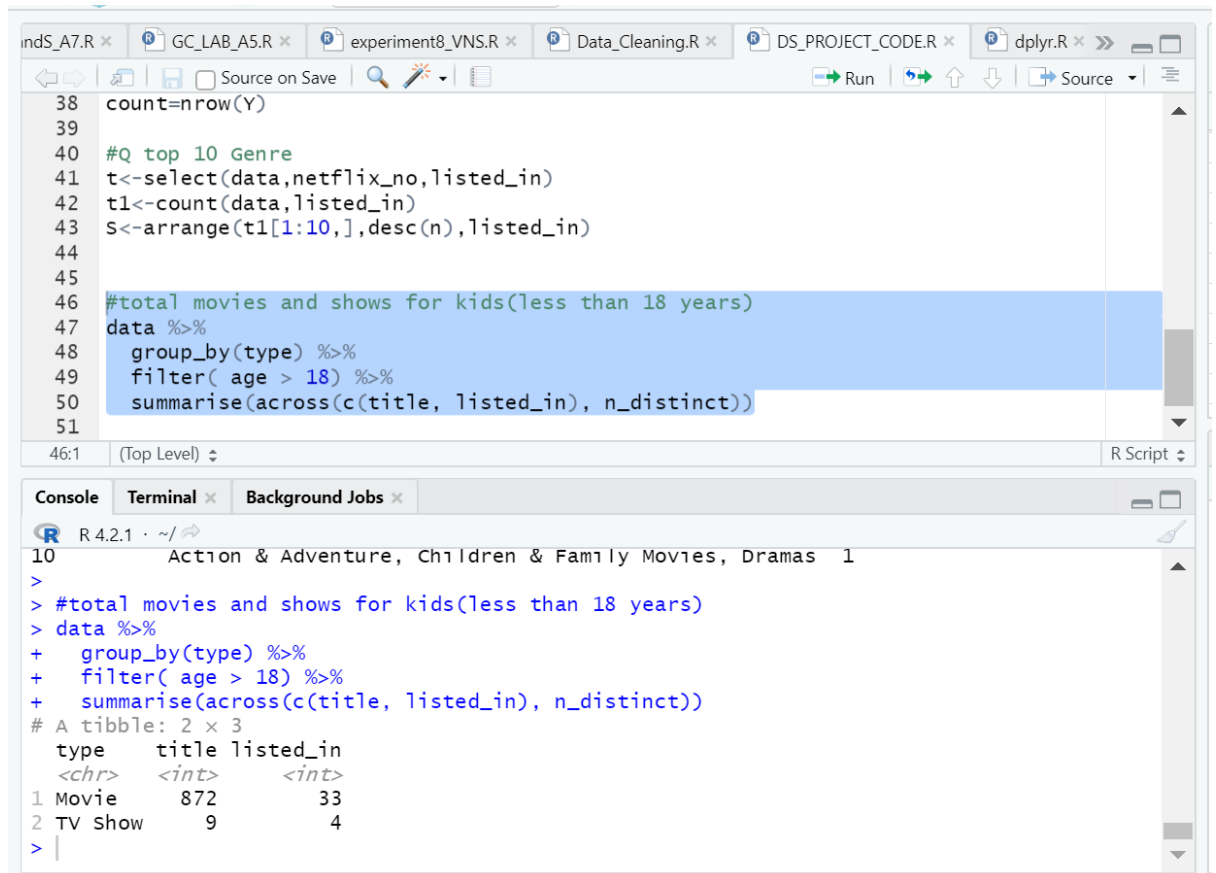
```

> #Q top 10 Genre
> t<-select(data,netflix_no,listed_in)
> t1<-count(data,listed_in)
> s<-arrange(t1[1:10,],desc(n),listed_in)
> s

```

	listed_in	n
1	Action & Adventure	68
2	Action & Adventure, Anime Features, International Movies	25
3	Action & Adventure, Anime Features, Sci-Fi & Fantasy	5
4	Action & Adventure, Children & Family Movies, Classic Movies	3
5	Action & Adventure, Children & Family Movies, Comedies	2
6	Action & Adventure, Anime Features, Children & Family Movies	1
7	Action & Adventure, Anime Features, Classic Movies	1
8	Action & Adventure, Anime Features, Horror Movies	1
9	Action & Adventure, Children & Family Movies	1
10	Action & Adventure, Children & Family Movies, Dramas	1

- Total Shows and Movies for kids



The screenshot shows the RStudio environment with several open R scripts in the top pane. The active script is `DS_PROJECT_CODE.R`, which contains the following R code:

```
38 count=nrow(Y)
39
40 #Q top 10 Genre
41 t<-select(data,netflix_no,listed_in)
42 t1<-count(data,listed_in)
43 s<-arrange(t1[1:10,],desc(n),listed_in)
44
45
46 #total movies and shows for kids(less than 18 years)
47 data %>%
48   group_by(type) %>%
49   filter( age > 18) %>%
50   summarise(across(c(title, listed_in), n_distinct))
51
```

The bottom pane shows the R console output for the code executed above. The output indicates that the data is filtered for ages greater than 18, resulting in a tibble with 2 rows and 3 columns: `type`, `title`, and `listed_in`.

```
R 4.2.1 ~/>
10 Action & Adventure, Children & Family Movies, Dramas 1
>
> #total movies and shows for kids(less than 18 years)
> data %>%
+   group_by(type) %>%
+   filter( age > 18) %>%
+   summarise(across(c(title, listed_in), n_distinct))
# A tibble: 2 x 3
  type    title listed_in
<chr> <int> <int>
1 Movie    872      33
2 TV Show     9       4
> |
```