

Youtube Music Channels Analysis

Harshad Jadhav

29/01/2022

Problem Statement

In this project, we are going to analyse 6 music channels and observe how they compete with each other, which channel is performing better than others, we can also predict what may happen in year 2022.

NOTE: 1) This analysis is performed at the start of the year 2022. Therefore there is not much data of this year and hence it won't effect the overall results. 2) Also, the results and conclusions are not globally established. These maybe true for only these 6 channels.

Installing and importing required packages

```
#install.packages("tidyverse")
#install.packages("ggplot2")
#install.packages("ggthemes")
#install.packages("cowplot")
#install.packages("lubridate")
#install.packages("forcats")
#install.packages("corrplot")

library(tidyverse)
library(ggthemes)
library(ggplot2)
library(cowplot)
library(lubridate)
library(forcats)
library(corrplot)
```

Importing required datasets

```
channels_details <- read.csv("Channel_Details.csv")
video_stats <- read.csv("Combined_Dataframe.csv")
```

Writing a function to represent the scale in thousands, millions and billions for the plots we are going to plot further in this analysis.

```

scale_units <- function(n) {
  labs <- ifelse(n < 1000, n, # less than thousands
    ifelse(n < 1e6, paste0(round(n/1e3), 'k'), # in thousands
      ifelse(n < 1e9, paste0(round(n/1e6), 'M'), # in millions
        ifelse(n < 1e12, paste0(round(n/1e9), 'B'), # in billions
          ifelse(n < 1e15, paste0(round(n/1e12), 'T'), # in trillions
            'too big!'
          )
        )
      )
    )
  return(labs)
}

```

Plotting various barplots, scatter plots, correlation matrix based on channels_details dataset

Bar graph representing Comparison of Subscribers of each Channel

```

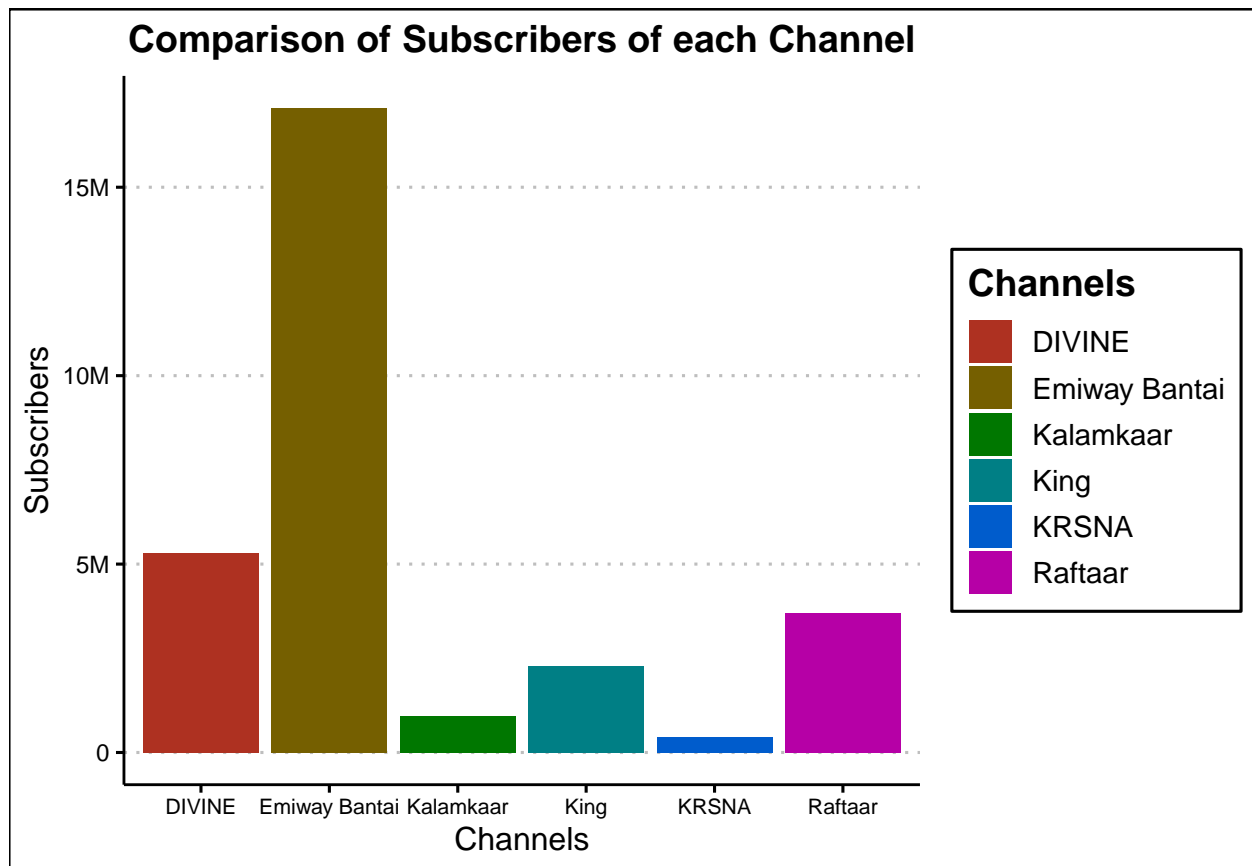
bar_subs <- ggplot(channels_details, aes(x= Channel_title, y= Subscribers_Count, fill =Channel_title))

bar_subs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_y_continuous(labels = scale_units) +

  labs(fill = "Channels",
    title = "Comparison of Subscribers of each Channel",
    x = "Channels",
    y = "Subscribers") +
  theme_clean() +

  theme(plot.title = element_text(size =14,
    face = "bold",
    colour = "black",
    lineheight = 1.2,
    hjust = 0.5
  ),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    axis.text.x = element_text(size = 8),
    axis.text.y = element_text(size = 9),
    legend.title = element_text(size = 14,
      face = "bold",
      colour = "black"))

```



We can see that Emiway Bantai has highest subscriber count with around 17 million subscribers.

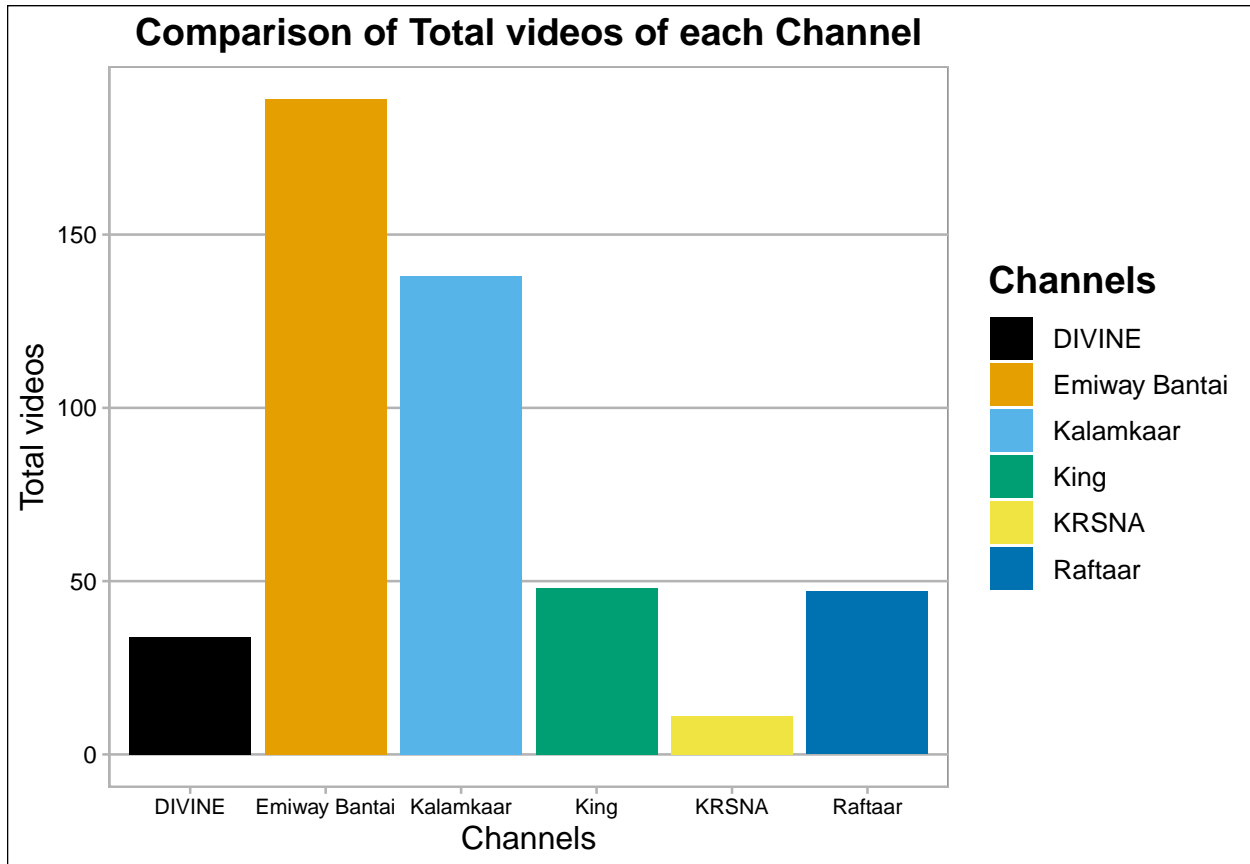
Bar graph representing Comparison of Total videos of each Channel

```
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

bar_vids <- ggplot(channels_details, aes(x= Channel_title, y= Total_Videos, fill =Channel_title))
bar_vids + geom_bar(stat = "identity") +
  scale_fill_manual(values=cbbPalette)+
  scale_y_continuous(labels = scale_units) +
  labs( fill = "Channels",
        title = "Comparison of Total videos of each Channel",
        x = "Channels",
        y = "Total videos") +
  theme_calc() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 8),
```

```
axis.text.y = element_text(size = 9),
legend.title = element_text(size = 14,
                             face = "bold",
                             colour = "black"))
```



The total videos uploaded by Emiway Bantai more as compared to other artists.

Bar graph representing Comparison of Total views of each Channel

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

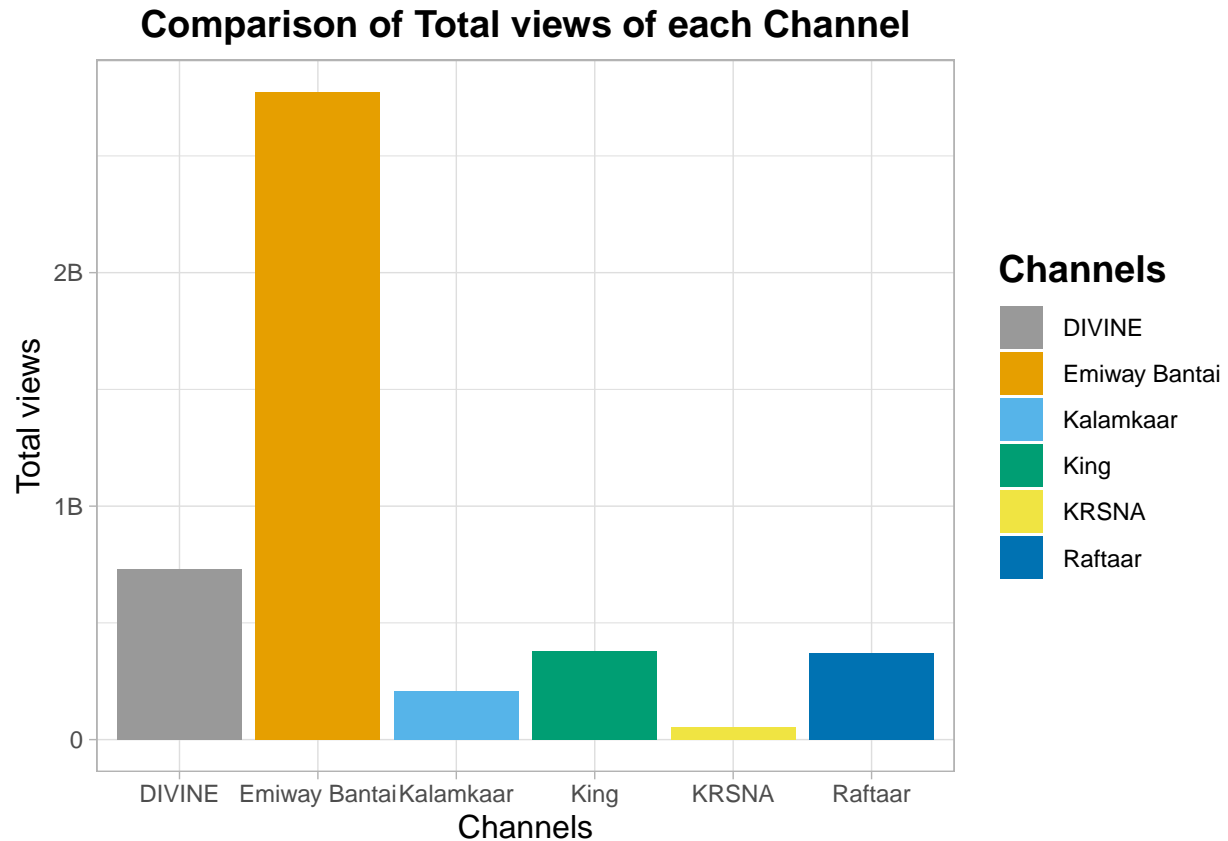
bar_views <- ggplot(channels_details, aes(x= Channel_title, y= Views_Count, fill =Channel_title))
bar_views + geom_bar(stat = "identity") +
  scale_fill_manual(values=cbPalette)+
  scale_y_continuous(labels = scale_units) +
  labs( fill = "Channels",
        title = "Comparison of Total views of each Channel",
        x = "Channels",
        y = "Total views") +
  theme_light() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
```

```

        lineheight = 1.2,
        hjust = 0.5),
axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12),
axis.text.x = element_text(size = 9),
axis.text.y = element_text(size = 9),
legend.title = element_text(size = 14,
                             face = "bold",
                             colour = "black"))

```



We can see that Emiway Bantai has highest view count with around 2 billion views.

Scatter plot of Correlation between Subscribers, Views and total videos uploaded

```

spsv <- ggplot(channels_details, aes(x=Subscribers_Count, y=Views_Count))+
  geom_point(aes(col= Channel_title, size = Total_Videos)) +
  geom_smooth(formula = y~x, method = "lm", col = "firebrick", se= FALSE) +
  scale_x_continuous(labels = scale_units) +
  scale_y_continuous(labels = scale_units) +
  labs( size = "Total Videos",
        col = "Channels",
        title = "Correlation between Subscribers vs Views",
        y = "Views",
        x = "Subscribers") +

```

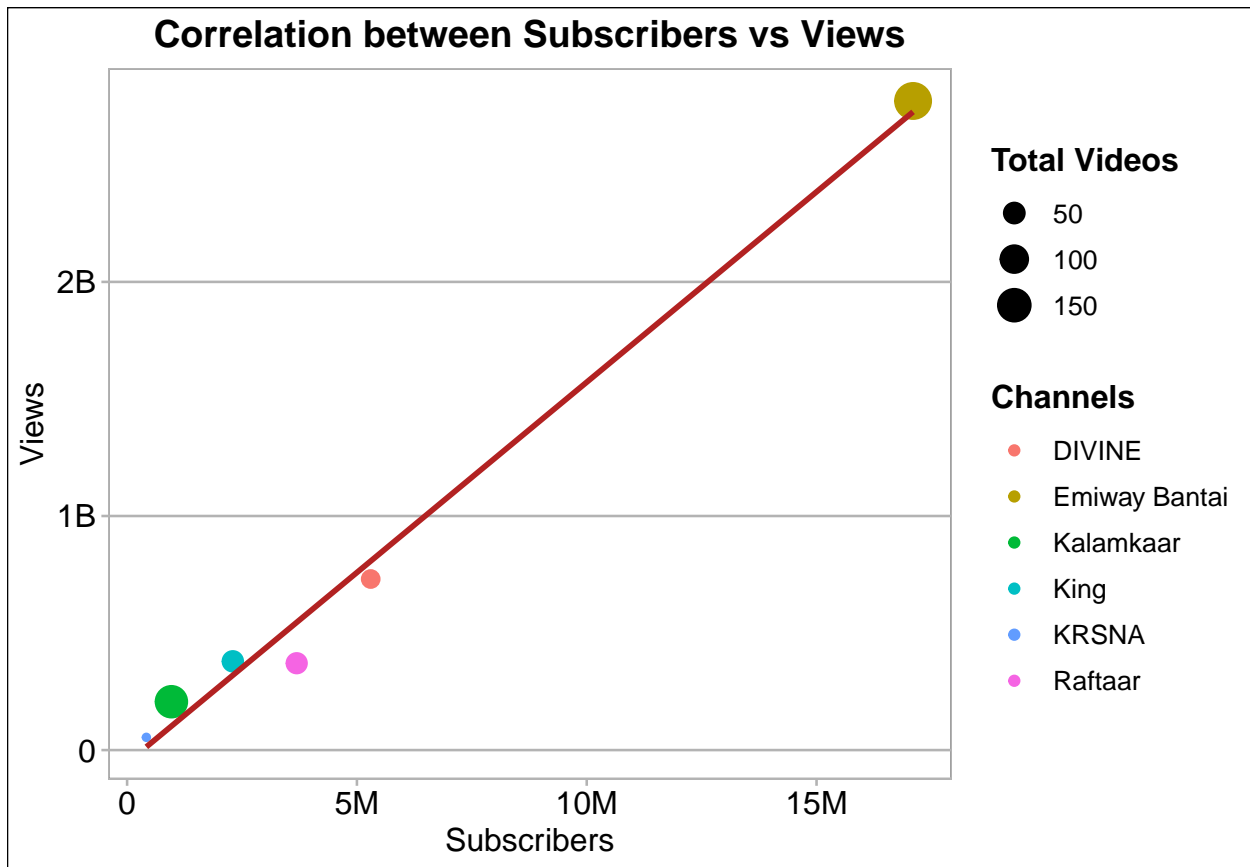
```

theme_calc() +

theme(plot.title = element_text(size = 14,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 12),
      axis.text.y = element_text(size = 12),
      legend.title = element_text(size = 12,
                                   face = "bold",
                                   colour = "black"))

plot(spsv)

```



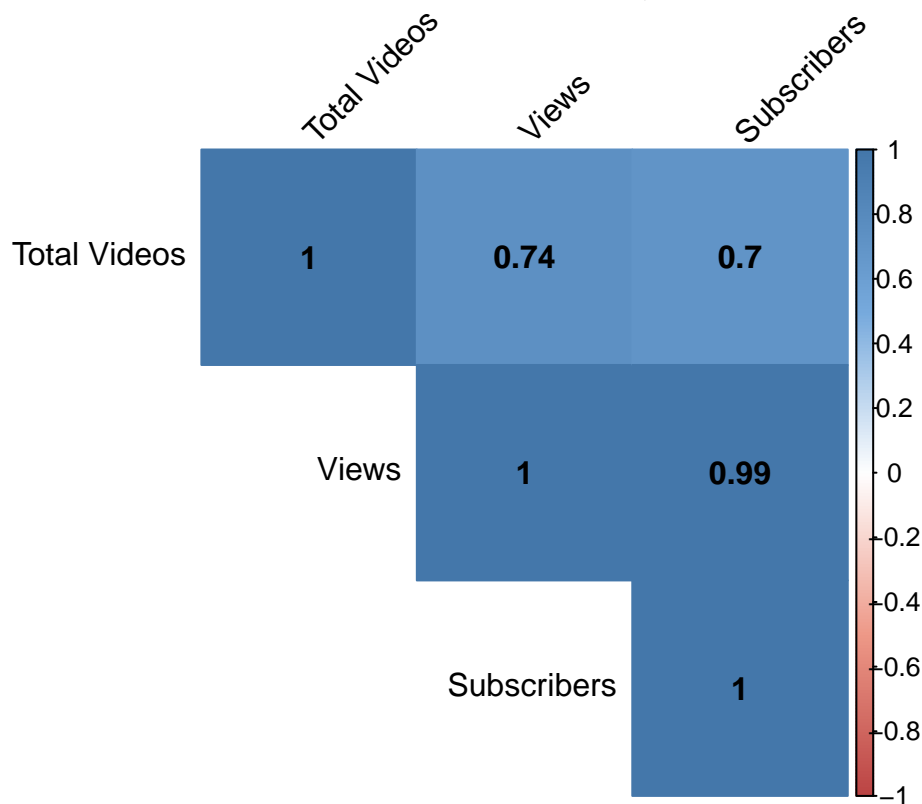
Scatter plot is useful to observe a relationship between variables. Here we have observing a relationship between subscribers, views and total videos uploaded by each artist. It can be inferred from the plot that there is a positive relation between them. However, “Divine” has uploaded less videos as compared to “Kalamkaar” and “King” and still has more views. This may be due to higher subscriber count as compared to other two artists. Overall, more the subscribers, more the views.

Correlation matrix between Subscribers, views, total videos

```
coll <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
channels_details_SVV <- channels_details[,4:6]
corr <- cor(channels_details_SVV)
colnames(corr) <- c("Subscribers", "Views", "Total Videos")
rownames(corr) <- c("Subscribers", "Views", "Total Videos")
corrplot(corr, method = "color", col = coll(200),

          mar= c(0,0,1.5,0),
          title = "Correlation matrix between Subscriber, Views and Videos Count",
          order = 'AOE', addCoef.col = "black",
          tl.col = "black",
          tl.srt = 45,
          tl.cex = 1,
          type = 'upper'
        )
```

Correlation matrix between Subscriber, Views and Videos Count



As can be seen in the correlation matrix that there is strong relationship between subscriber count and view count. However the data can be skewed towards subscriber count due to the extraordinary subscribers of Emiway Bantai.

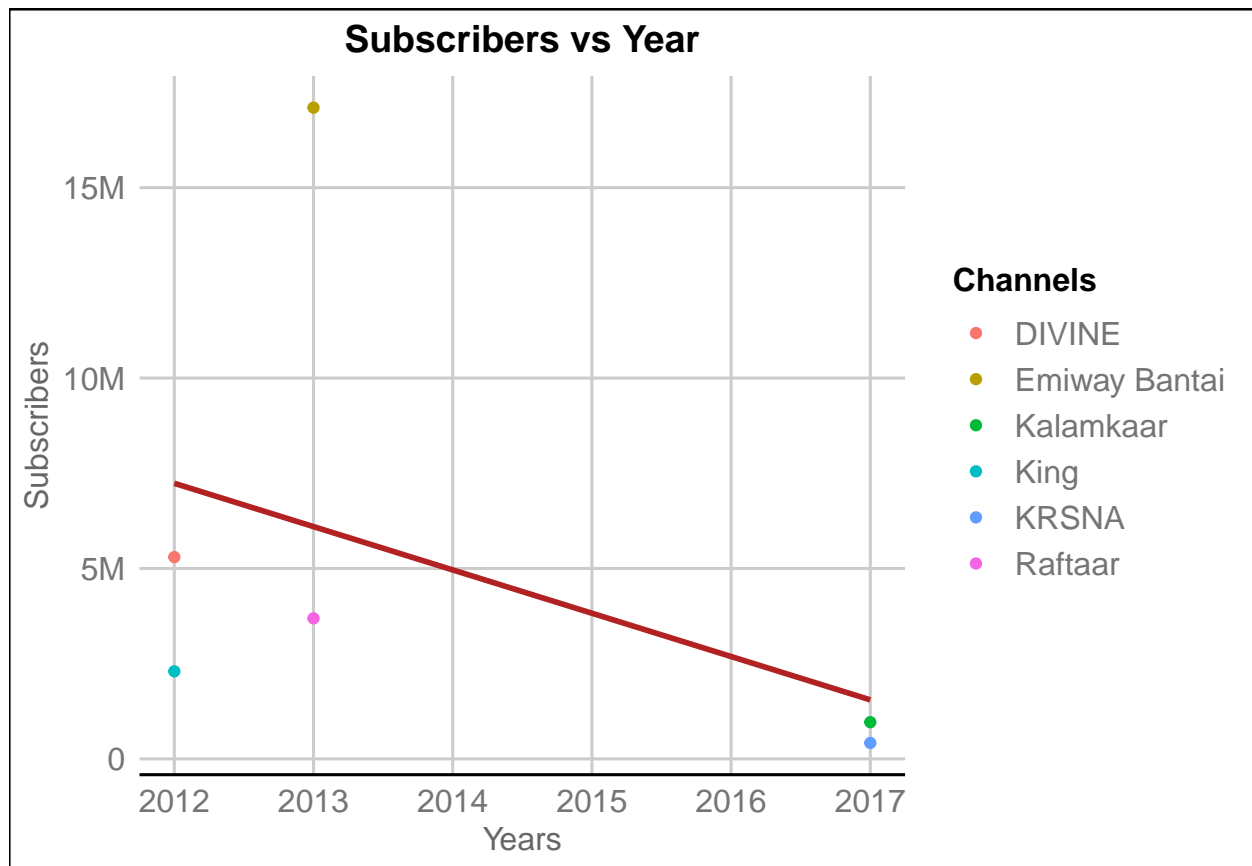
Observing if the year of the start of channel affects on subscriber count, views count

```
class(channels_details$Start_of_Channel)  ## this was character
channels_details$Start_of_Channel <- as.Date(channels_details$Start_of_Channel) ## changed to
##date format
channels_details$Year <- format(channels_details$Start_of_Channel, "%Y")  ## formed a new column
##that only extracts year from the start of channel
channels_details$Year <- as.Date(as.character(channels_details$Year), format = "%Y") ## converting
## the year column to date
channels_details$Year <- year(channels_details$Year) ## only keeping the years
class(channels_details$Year) ## checking class of the year column...it is numeric
```

Observing whether Subscriber count depend on how early the channel was started

```
yr_subs <- ggplot(channels_details, aes(x=Year, y=Subscribers_Count))+
  geom_point(aes(col= Channel_title))+
  geom_smooth(formula = y~x, method = "lm", col = "firebrick", se= FALSE)+
  scale_y_continuous(labels = scale_units)+
  labs(col = "Channels",
       title = "Subscribers vs Year",
       y = "Subscribers",
       x = "Years")+
  theme_gdocs()+
  theme(plot.title = element_text(size = 14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        legend.title = element_text(size = 12,
                                     face = "bold",
                                     colour = "black"))

plot(yr_subs)
```

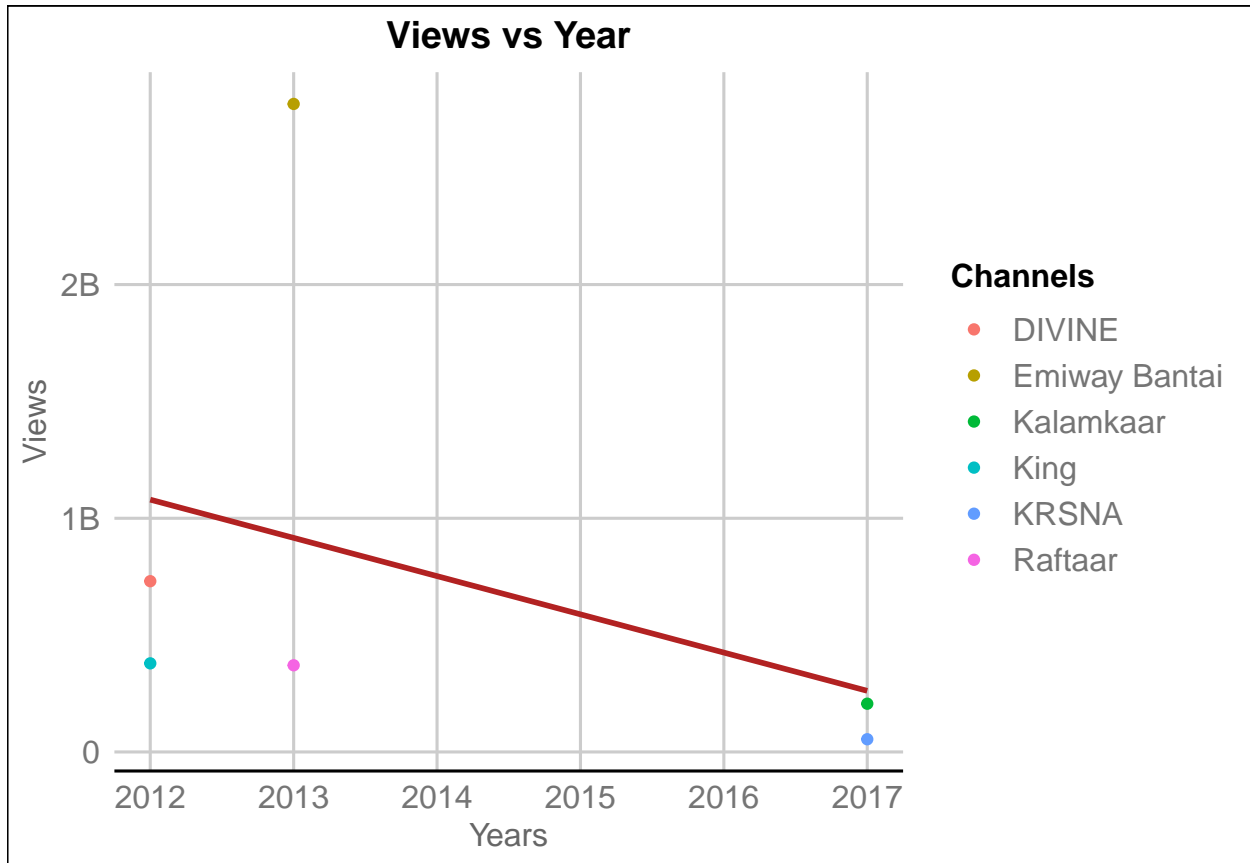
Overall if the channel was started early, the subscribers gained by the channel is also more. “Emiway Bantai” and “Raftaar” had started their channel in 2013 but still has more subscribers than “Divine” and “King” respectively. But the channels started in 2017 have less subscribers compared to all other channels.

Observing whether Views count depend on how early the channel was started

```
yr_views <- ggplot(channels_details, aes(x=Year, y=Views_Count))+

  geom_smooth(formula = y~x, method = "lm", col = "firebrick", se= FALSE)+
  geom_point(aes(col= Channel_title))+
  scale_y_continuous(labels = scale_units)+
  labs(col = "Channels",
       title = "Views vs Year",
       y = "Views",
       x = "Years")+
  theme_gdocs()+
  theme(plot.title = element_text(size = 14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 12),
```

```
axis.text.y = element_text(size = 12),
legend.title = element_text(size = 12,
                             face = "bold",
                             colour = "black"))
plot(yr_views)
```



Here also there is positive relation between views and the year in which channel was started. Channels that were started early have accumulated more views. “Emiway Bantai” had started his channel a year late as compared to “Divine” and “King” but still has more views.

Frequency of Songs uploaded by an artist per year

```
video_stats <- as_tibble(video_stats) ## converting dataframe into tibble
```

Emiway Bantai

```
## selecting and filtering rows and columns required
emiway <- video_stats %>% select(Channel_title, Published_date) %>%
  filter(Channel_title == "Emiway Bantai") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Channel_title) %>%
```

```

count(Year) %>%
  rename(Songs_per_Year = n)

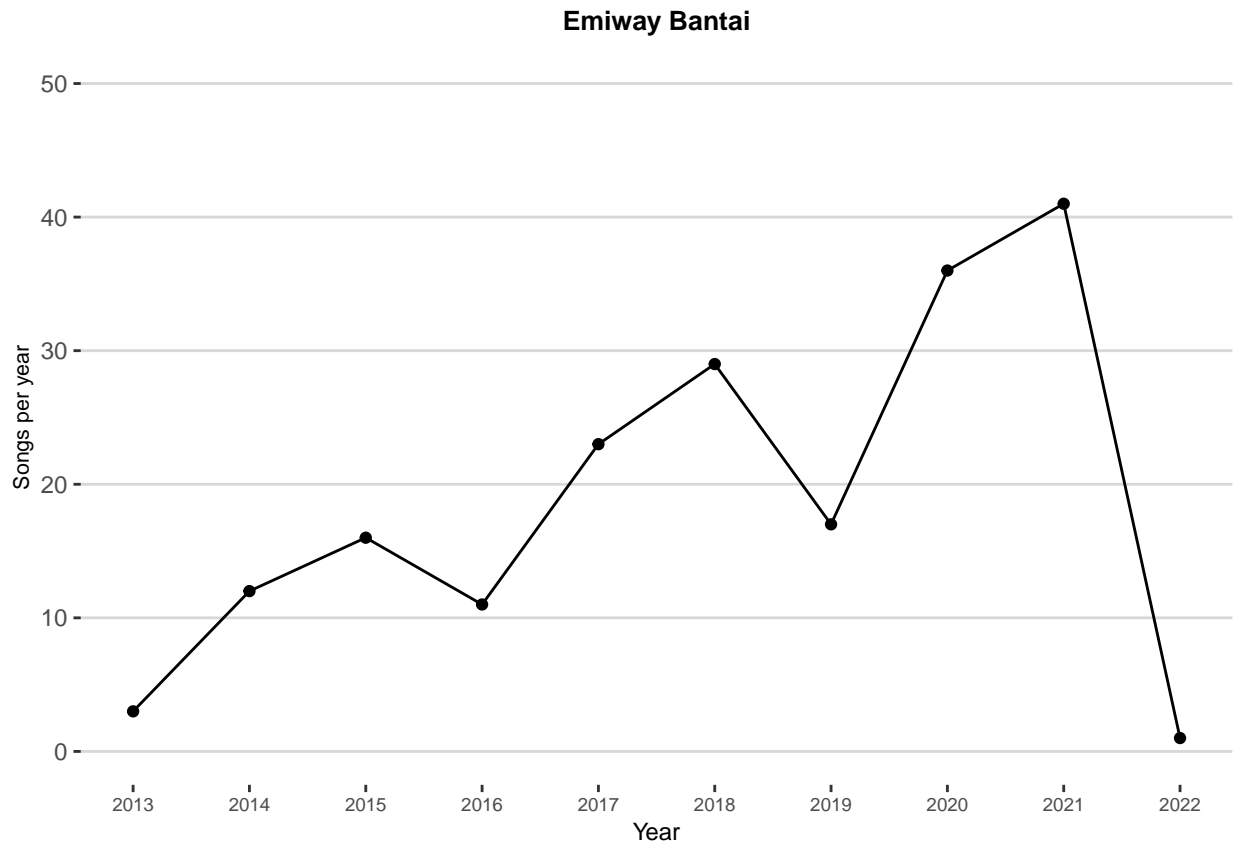
## Plotting graph that indicates number of songs released per year

emiway_spy <- ggplot(emiway, aes(x =Year, y = Songs_per_Year))+
  geom_line() + geom_point() +
  scale_x_continuous(breaks = 2013:2022)+
  ylim(0,50) +
  labs(title = "Emiway Bantai",
        y = "Songs per year",
        x= "Year") +
  theme_hc()+

  theme(plot.title = element_text(size = 10,
                                    face = "bold",
                                    colour = "black",
                                    lineheight = 1.2,
                                    hjust = 0.5),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 8),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 9))

plot(emiway_spy)

```



Divine

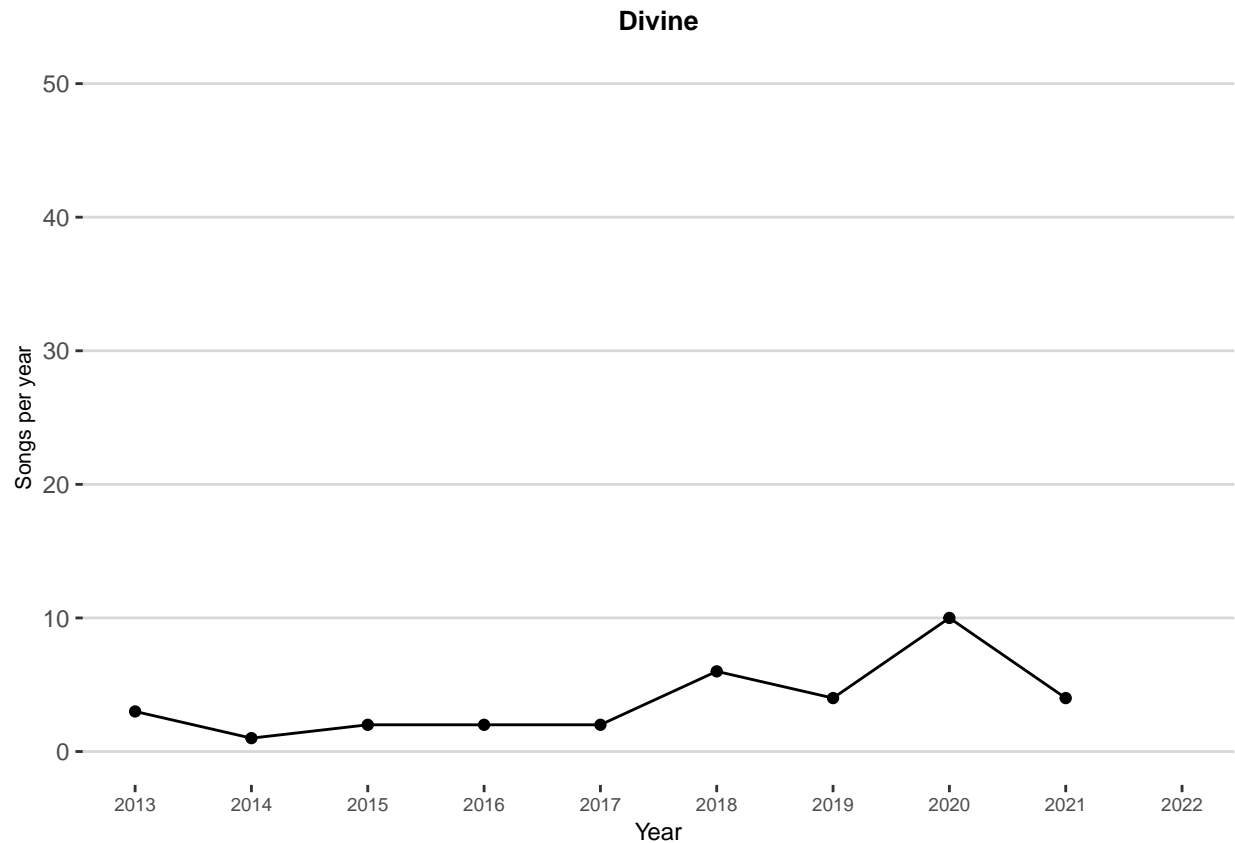
```
## selecting and filtering rows and columns required

divine <- video_stats %>% select(Channel_title, Published_date) %>%
  filter(Channel_title == "DIVINE") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Channel_title) %>%
  count(Year) %>%
  rename(Songs_per_Year = n)

## Plotting graph that indicates number of songs released per year

divine_spy <- ggplot(divine, aes(x =Year, y = Songs_per_Year))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1))+
  ylim(0,50) +
  labs(title = "Divine",
       y = "Songs per year",
       x= "Year") +
  theme_hc()+

  theme(plot.title = element_text(size = 10,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 8),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 9))
plot(divine_spy)
```



Raftaar

```
## selecting and filtering rows and columns required

raftaar <- video_stats %>% select(Channel_title, Published_date) %>%
  filter(Channel_title == "Raftaar") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Channel_title) %>%
  count(Year) %>%
  rename(Songs_per_Year = n)

## Plotting graph that indicates number of songs released per year

raftaar_spy <- ggplot(raftaar, aes(x =Year, y = Songs_per_Year))+
  geom_line() + geom_point() +
  scale_x_continuous(breaks = 2013:2022)+
  ylim(0,50) +
  labs(title = "Raftaar",
       y = "Songs per year",
       x= "Year") +
  theme_hc()+

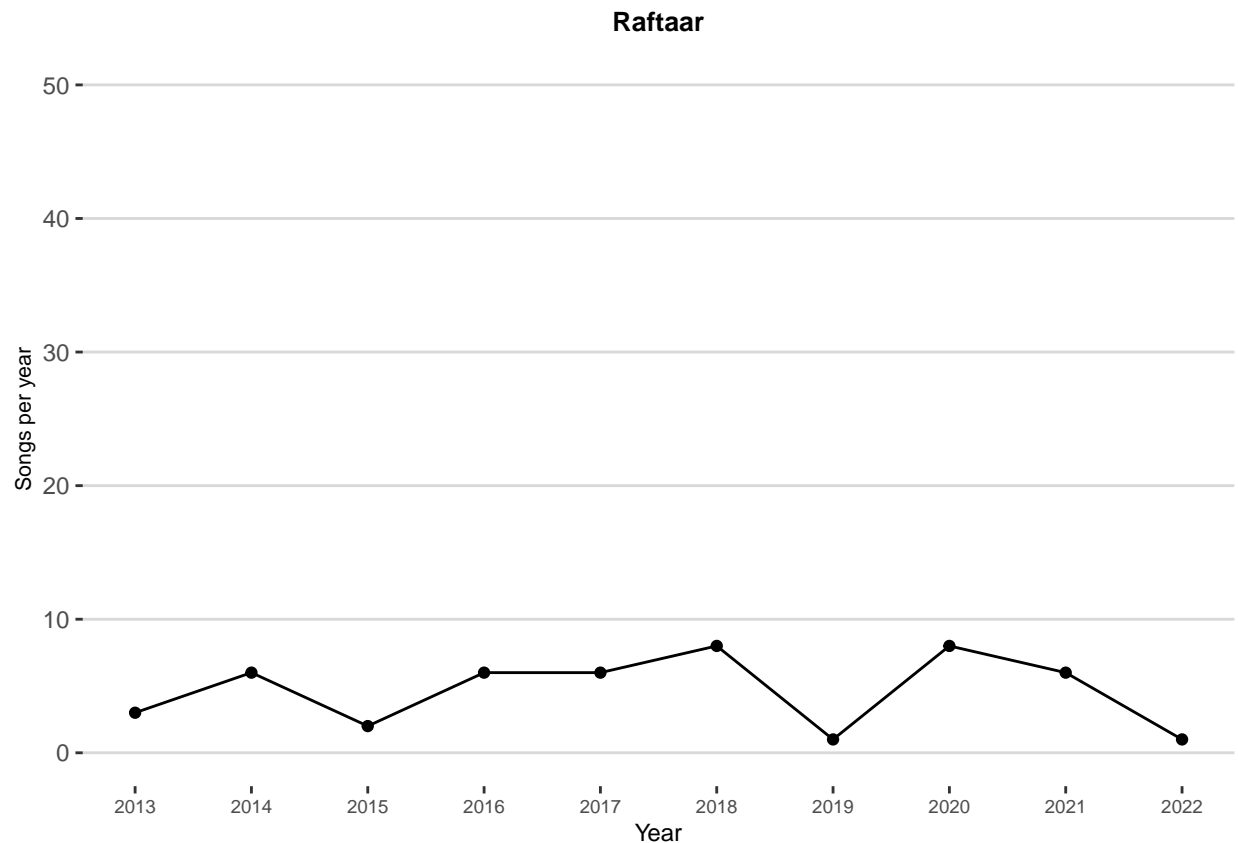
  theme(plot.title = element_text(size = 10,
                                   face = "bold",
```

```

        colour = "black",
        lineheight = 1.2,
        hjust = 0.5),
    axis.title.x = element_text(size = 9),
    axis.title.y = element_text(size = 8),
    axis.text.x = element_text(size = 7),
    axis.text.y = element_text(size = 9))

plot(raftaar_spy)

```



Kalamkaar

```

## selecting and filtering rows and columns required

kalamkaar <- video_stats %>% select(Channel_title, Published_date) %>%
  filter(Channel_title == "Kalamkaar") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Channel_title) %>%
  count(Year) %>%
  rename(Songs_per_Year = n)

## Plotting graph that indicates number of songs released per year

```

```

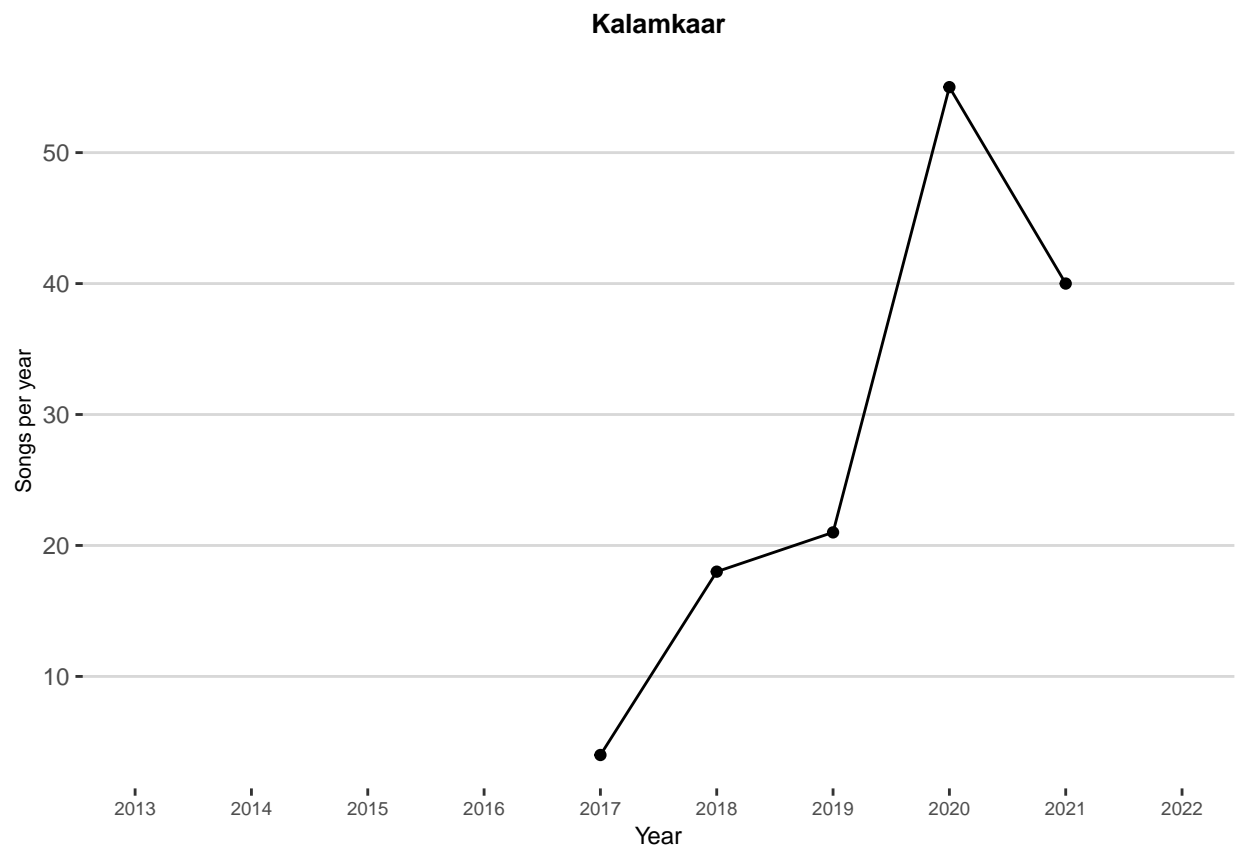
kalamkaar_spy <- ggplot(kalamkaar, aes(x =Year, y = Songs_per_Year))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1))+

  labs(title = "Kalamkaar",
        y = "Songs per year",
        x= "Year") +
  theme_hc()+

  theme(plot.title = element_text(size = 10,
                                    face = "bold",
                                    colour = "black",
                                    lineheight = 1.2,
                                    hjust = 0.5),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 8),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 9))

plot(kalamkaar_spy)

```



Krnsa

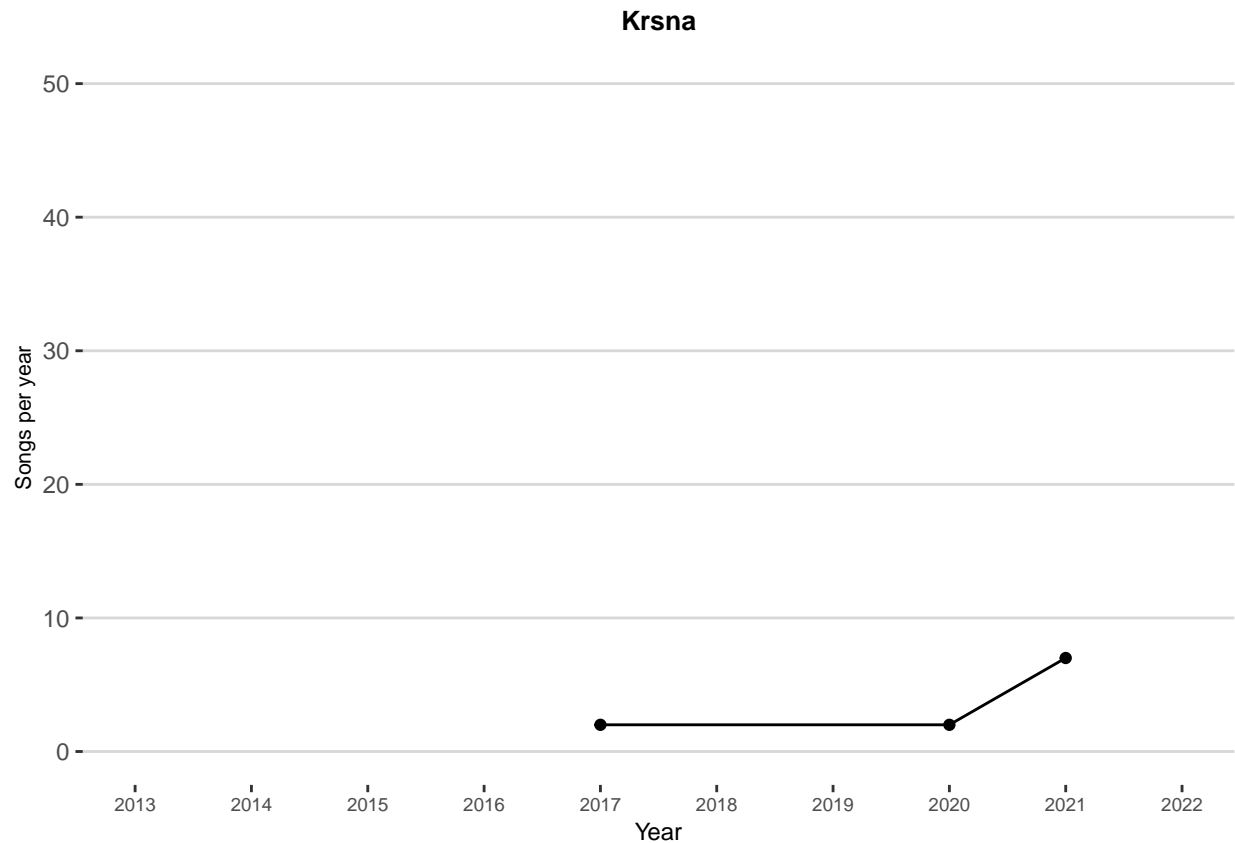
```
## selecting and filtering rows and columns required

krsna <- video_stats %>% select(Channel_title, Published_date) %>%
  filter(Channel_title == "KRSNA") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Channel_title) %>%
  count(Year) %>%
  rename(Songs_per_Year = n)

## Plotting graph that indicates number of songs released per year

krsna_spy <- ggplot(krsna, aes(x =Year, y = Songs_per_Year))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1))+
  ylim(0,50) +
  labs(title = "Krsna",
       y = "Songs per year",
       x= "Year") +
  theme_hc()+

  theme(plot.title = element_text(size = 10,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 8),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 9))
plot(krsna_spy)
```

King

```
## selecting and filtering rows and columns required

king <- video_stats %>% select(Channel_title, Published_date) %>%
  filter(Channel_title == "King") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Channel_title) %>%
  count(Year) %>%
  rename(Songs_per_Year = n)

## Plotting graph that indicates number of songs released per year

king_spy <- ggplot(king, aes(x =Year, y = Songs_per_Year))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1))+
  ylim(0,50) +
  labs(title = "King",
       y = "Songs per year",
       x= "Year") +
  theme_hc()+

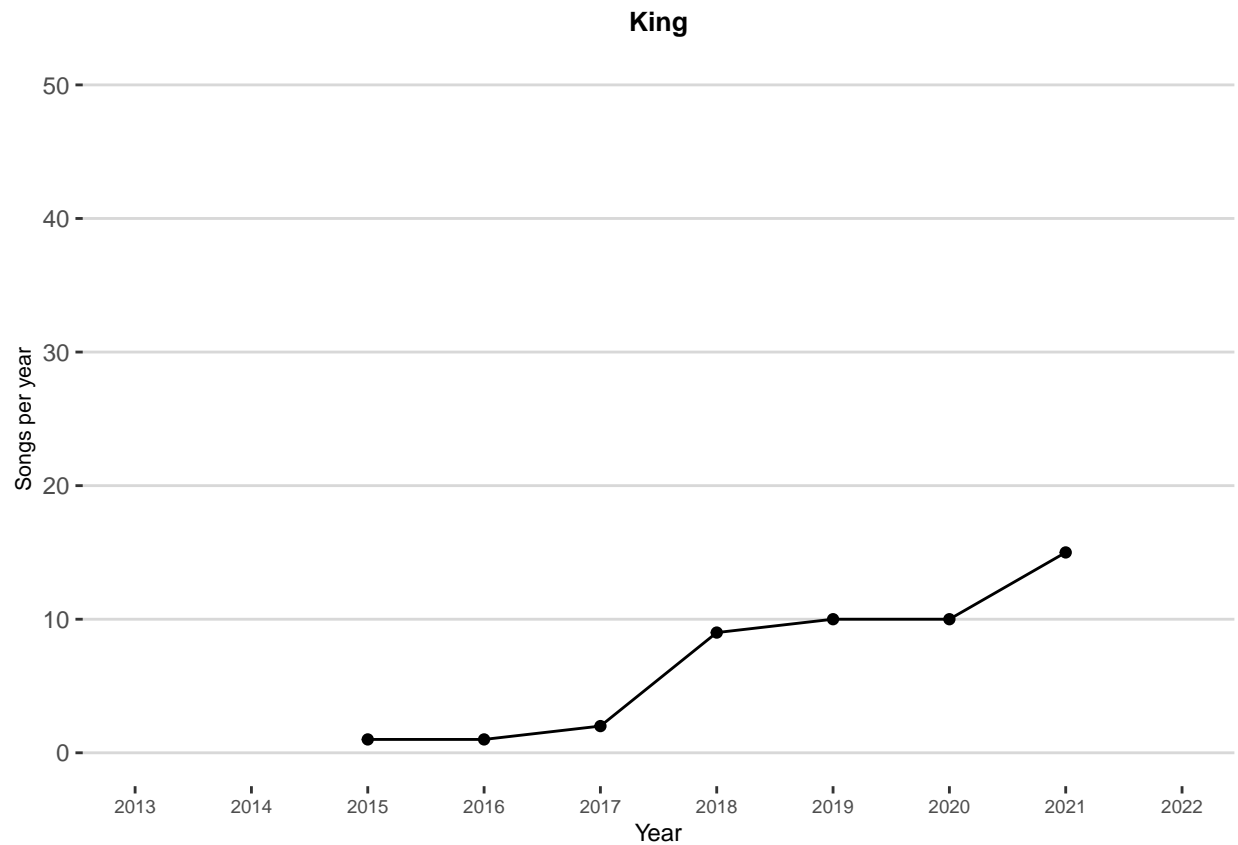
  theme(plot.title = element_text(size = 10,
                                   face = "bold",
```

```

        colour = "black",
        lineheight = 1.2,
        hjust = 0.5),
axis.title.x = element_text(size = 9),
axis.title.y = element_text(size = 8),
axis.text.x = element_text(size = 7),
axis.text.y = element_text(size = 9))

plot(king_spy)

```



Representing all the plots of the above section in one graph

```

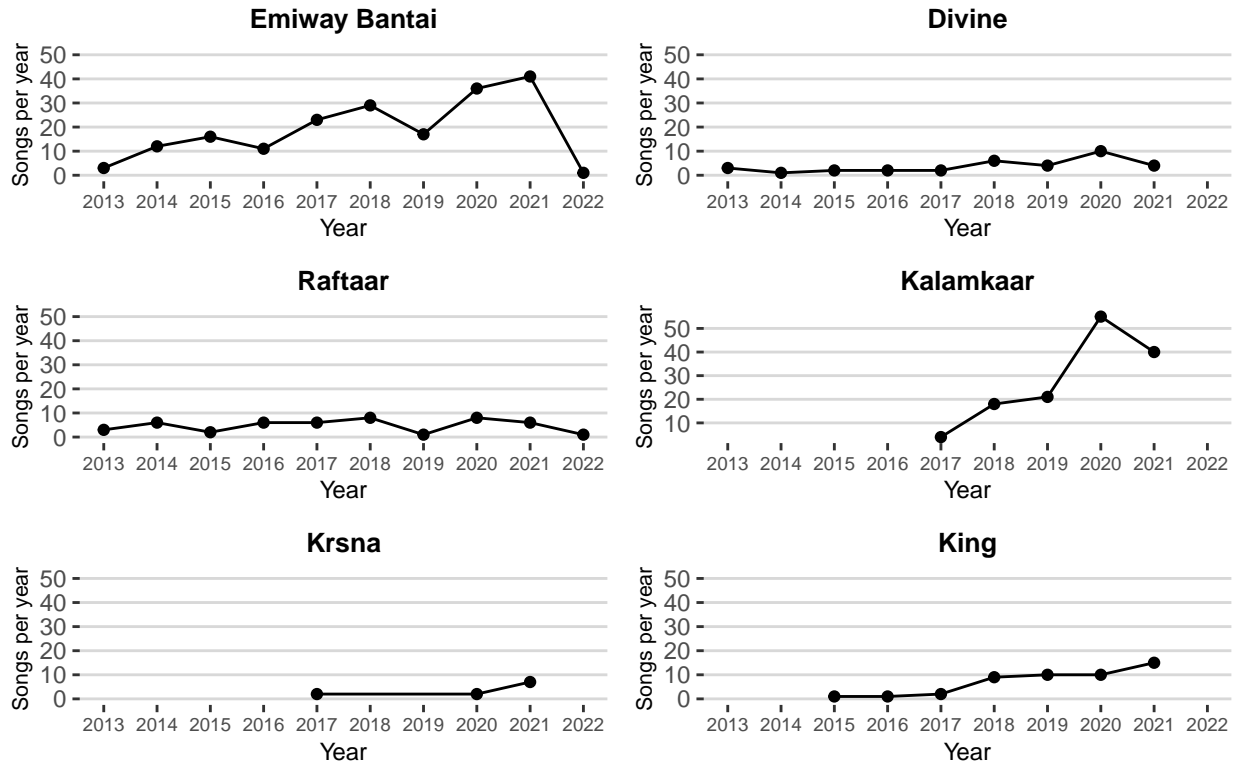
plots <- plot_grid(emiway_spy, divine_spy, raftaar_spy, kalamkaar_spy, krsna_spy, king_spy,
  nrow = 3)

title <- ggdraw() + draw_label("Songs uploaded per year by Artists", fontface='bold')

plot_grid(title, plots, ncol=1, rel_heights=c(0.1, 1))

```

Songs uploaded per year by Artists



- 1) Emiway Bantai: He has uploaded more than 40 videos in 2021 which is his best. Also, you can observe a pattern here. In 2015 he had uploaded more videos as compared to 2013 and 2014 and then in 2016 there was drop and then in 2018 more videos were uploaded as compared to 2016 and 2017 and then again after 2 years in 2019 there was a drop. So maybe at the end of 2022, we will be observing a drop in the total videos uploaded this year.
- 2) Divine: The number tracks uploaded from 2015 to 2017 is constant and then there is rise and fall every alternate year. We may see a rise in 2022.
- 3) Raftaar: Almost a similar pattern as Divine. Rise and fall every alternate year. Only for 2016 ,2017 and 2018 there was a constant rise. We may see a rise in 2022.
- 4) Kalamkaar: This channel was always on the rise till 2020 and 2020 was the year they uploaded highest number of songs per year. But then in 2021, there was decline.
- 5) Krsna: Very few uploads per year. Almost constant for few years. His best was in 2021.
- 6) King: Ever since his channel started, the number of uploads per year have always increased. SO good news for his Fans !! He may upload more videos this year than 2021.

Total views per year by channels

Emiway Bantai

```

## selecting and filtering rows and columns required

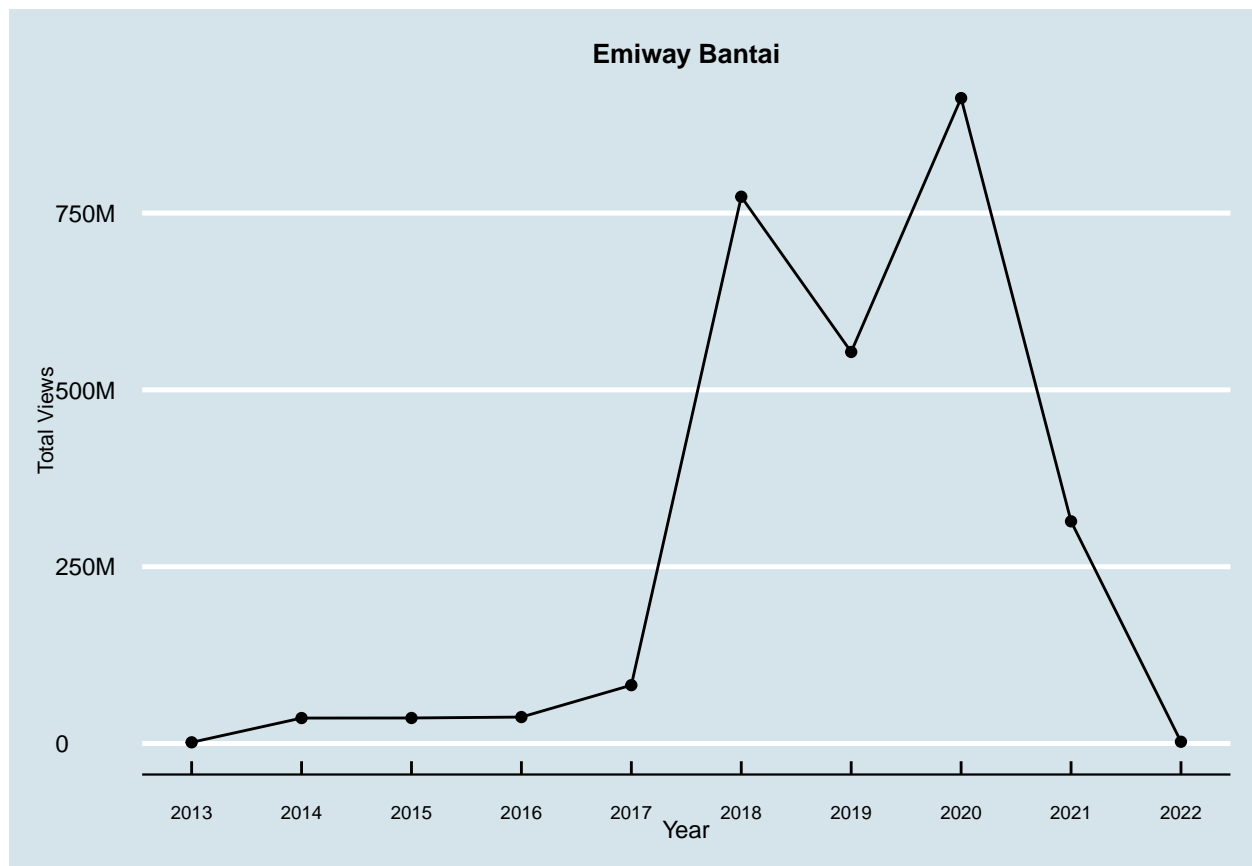
emiway_views <- video_stats %>% select(Channel_title, Published_date, Views) %>%
  filter(Channel_title == "Emiway Bantai") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Year) %>%
  summarise(sum_views =sum(Views))

## Plotting graph that indicated total views per year

emiway_lg <- ggplot( emiway_views, aes(x =Year, y = sum_views))+
  geom_line() + geom_point() +
  scale_x_continuous(breaks = 2013:2022) +
  scale_y_continuous(labels = scale_units) +

  labs(title = "Emiway Bantai",
        y = "Total Views",
        x= "Year") +
  theme_economist()+
  theme(plot.title = element_text(size = 10,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 8),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 9))
plot(emiway_lg)

```



Divine

```
## selecting and filtering rows and columns required

divine_views <- video_stats %>% select(Channel_title, Published_date, Views) %>%
  filter(Channel_title == "DIVINE") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Year) %>%
  summarise(sum_views = sum(Views))

## Plotting graph that indicated total views per year

divine_lg <- ggplot( divine_views, aes(x =Year, y = sum_views))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1)) +
  scale_y_continuous(labels = scale_units) +

  labs(title = "Divine",
        y = "Total Views",
        x= "Year") +
  theme_economist()+

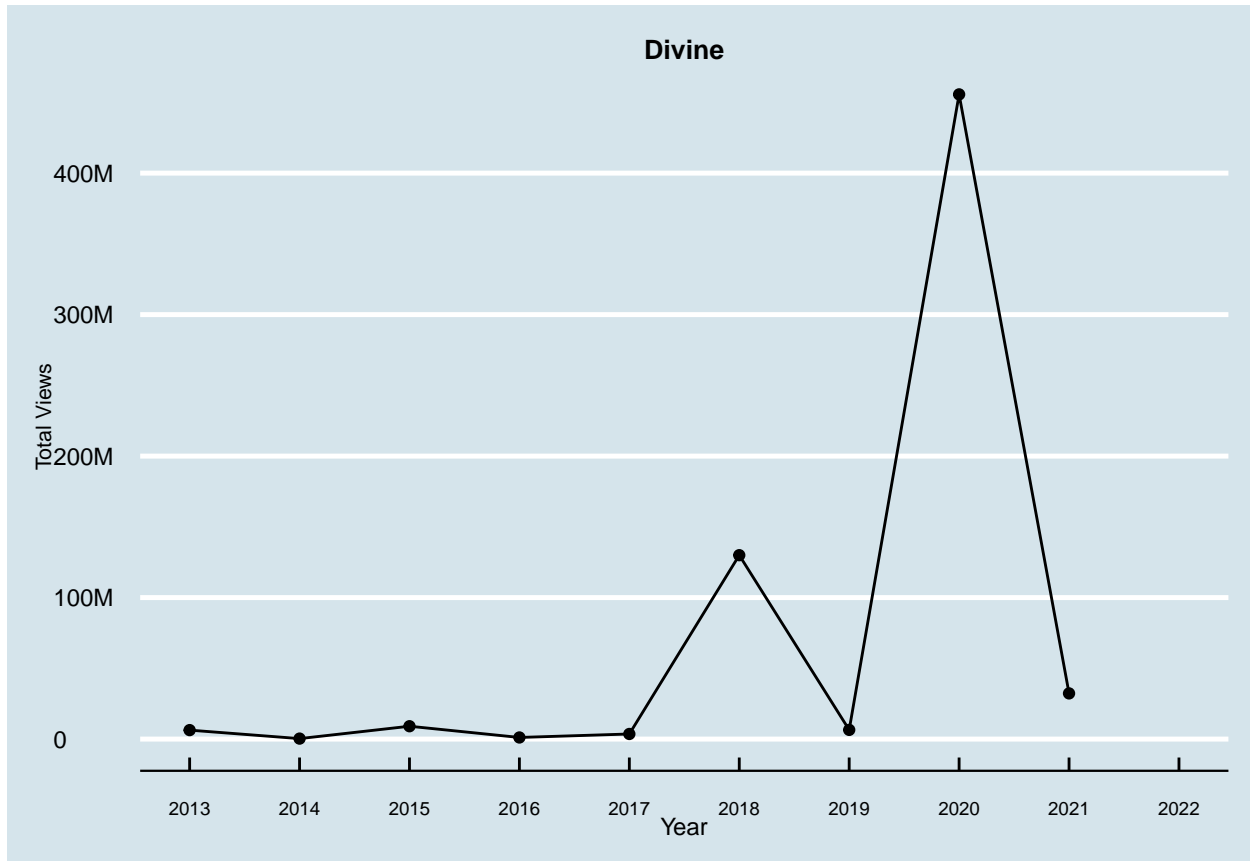
  theme(plot.title = element_text(size = 10,
                                   face = "bold",
```

```

        colour = "black",
        lineheight = 1.2,
        hjust = 0.5),
    axis.title.x = element_text(size = 9),
    axis.title.y = element_text(size = 8),
    axis.text.x = element_text(size = 7),
    axis.text.y = element_text(size = 9))

plot(divine_lg)

```



Raftaar

```

## selecting and filtering rows and columns required

raftaar_views <- video_stats %>% select(Channel_title, Published_date, Views) %>%
  filter(Channel_title == "Raftaar") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Year) %>%
  summarise(sum_views = sum(Views))

## Plotting graph that indicated total views per year

raftaar_lg <- ggplot( raftaar_views, aes(x =Year, y = sum_views))+

```

```

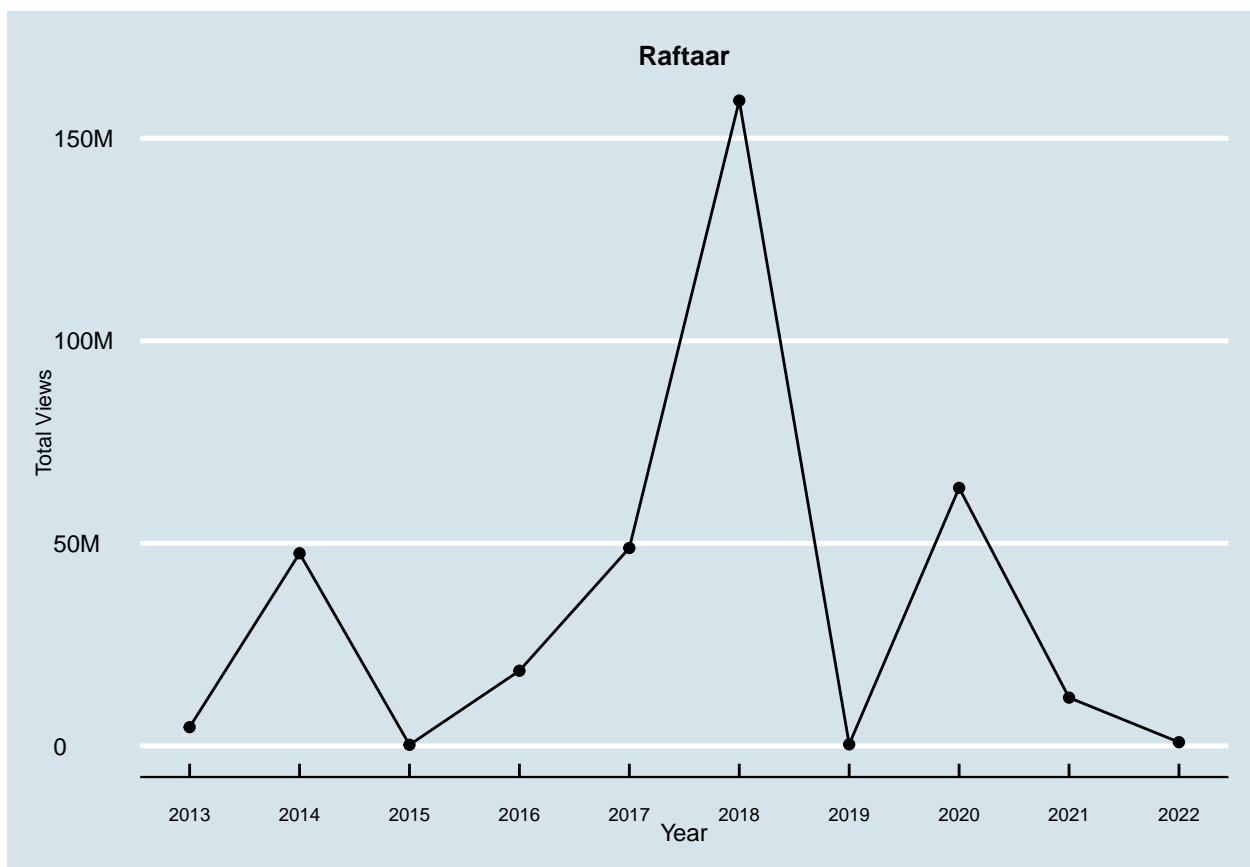
geom_line() + geom_point() +
scale_x_continuous(breaks = 2013:2022) +
scale_y_continuous(labels = scale_units) +

labs(title = "Raftaar",
      y = "Total Views",
      x = "Year") +
theme_economist()+

theme(plot.title = element_text(size = 10,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5),
      axis.title.x = element_text(size = 9),
      axis.title.y = element_text(size = 8),
      axis.text.x = element_text(size = 7),
      axis.text.y = element_text(size = 9))

plot(raftaar_lg)

```



Kalamkaar

```
## selecting and filtering rows and columns required

kalamkaar_views <- video_stats %>% select(Channel_title, Published_date, Views) %>%
  filter(Channel_title == "Kalamkaar") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Year) %>%
  summarise(sum_views = sum(Views))

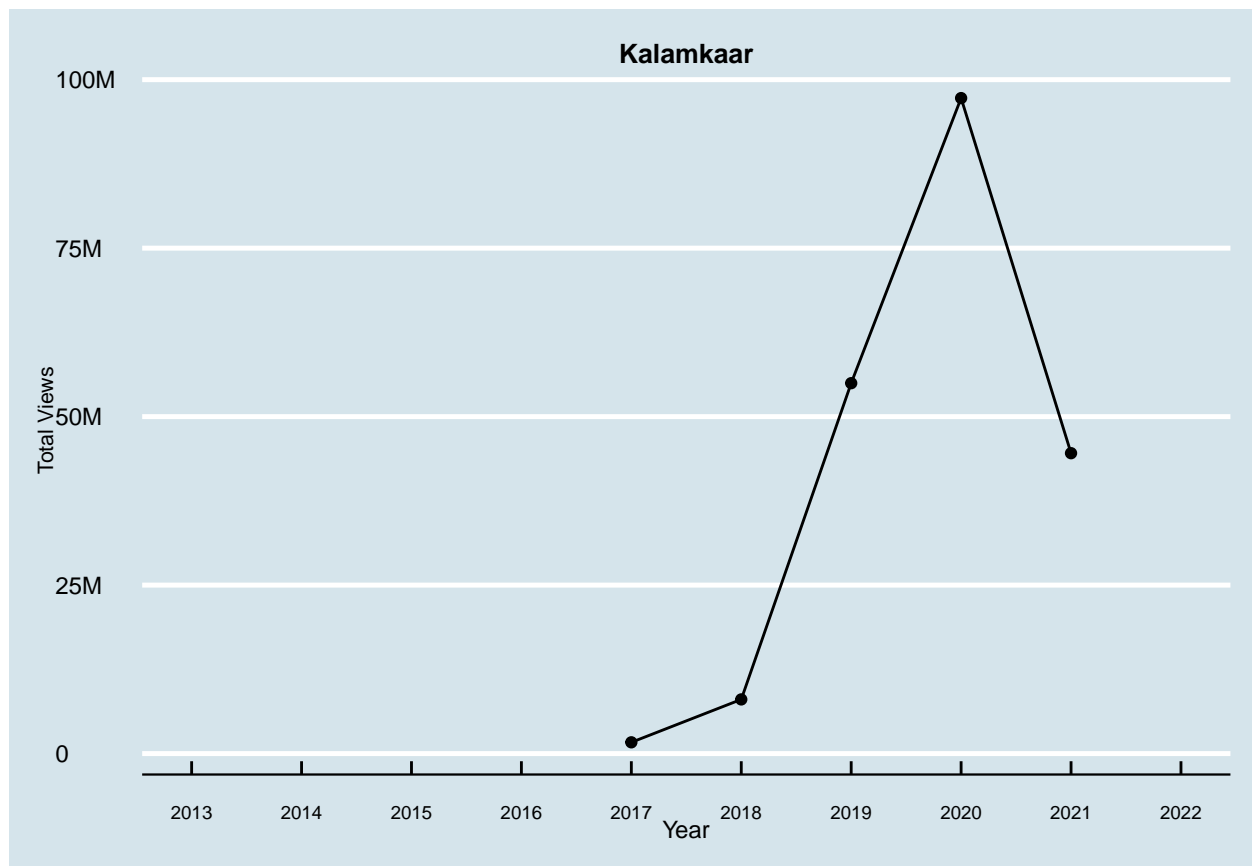
## Plotting graph that indicated total views per year

kalamkaar_lg <- ggplot( kalamkaar_views, aes(x =Year, y = sum_views))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1)) +
  scale_y_continuous(labels = scale_units) +

  labs(title = "Kalamkaar",
        y = "Total Views",
        x= "Year") +
  theme_economist()+

  theme(plot.title = element_text(size = 10,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 8),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 9))

plot(kalamkaar_lg)
```

Krsna

```
krsna_views <- video_stats %>% select(Channel_title, Published_date, Views) %>%
  filter(Channel_title == "KRSNA") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Year) %>%
  summarise(sum_views = sum(Views))

## Plotting graph that indicated total views per year

krsna_lg <- ggplot( krsna_views, aes(x =Year, y = sum_views))+
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1)) +
  scale_y_continuous(labels = scale_units) +

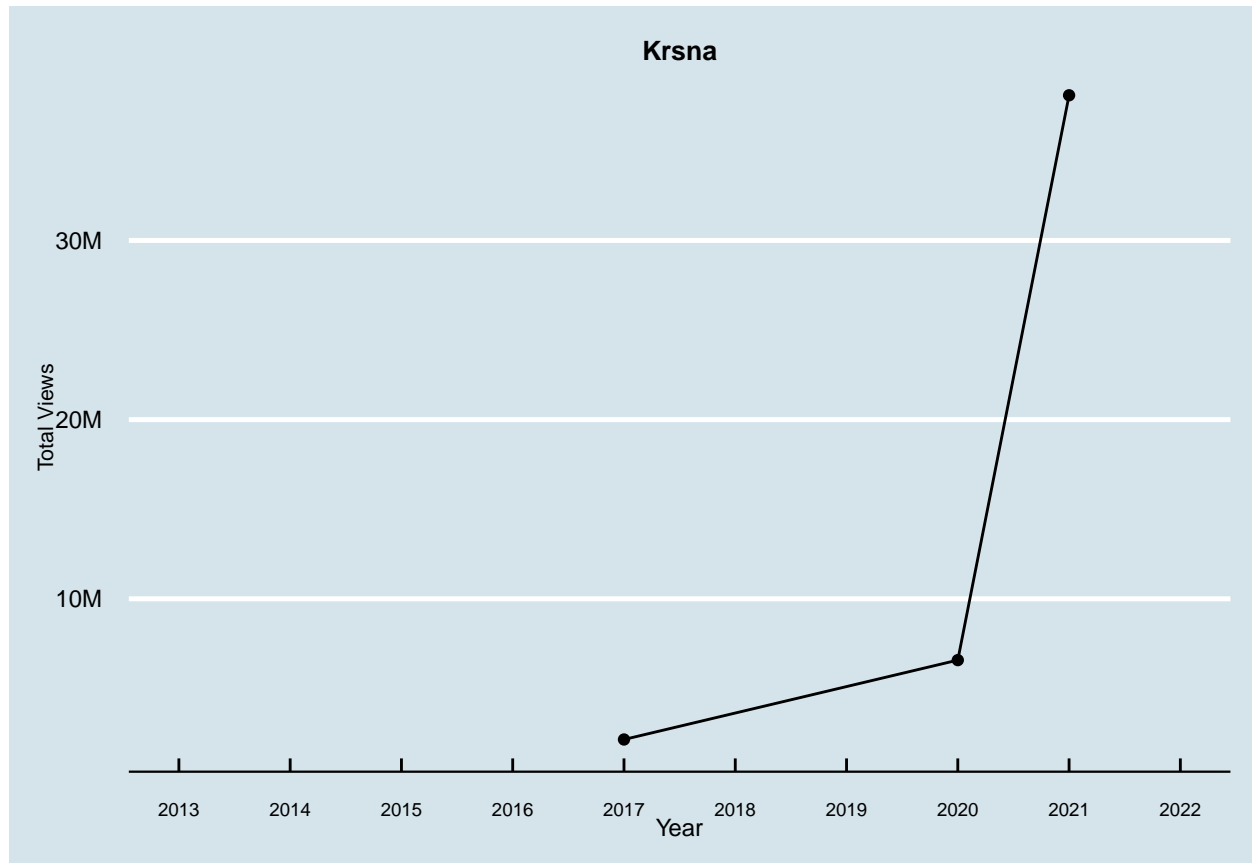
  labs(title = "Krsna",
        y = "Total Views",
        x= "Year") +
  theme_economist()+

  theme(plot.title = element_text(size = 10,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
```

```

      hjust = 0.5),
    axis.title.x = element_text(size = 9),
    axis.title.y = element_text(size = 8),
    axis.text.x = element_text(size = 7),
    axis.text.y = element_text(size = 9))
plot(krsna_lg)

```



King

```

## selecting and filtering rows and columns required

king_views <- video_stats %>% select(Channel_title, Published_date, Views) %>%
  filter(Channel_title == "King") %>%
  mutate(Year = as.numeric(format(as.Date(Published_date), "%Y"))) %>%
  group_by(Year) %>%
  summarise(sum_views = sum(Views))

## Plotting graph that indicated total views per year

king_lg <- ggplot(king_views, aes(x = Year, y = sum_views)) +
  geom_line() + geom_point() +
  scale_x_continuous(limits = c(2013, 2022), breaks = seq(2013, 2022, by = 1)) +
  scale_y_continuous(labels = scale_units()) +

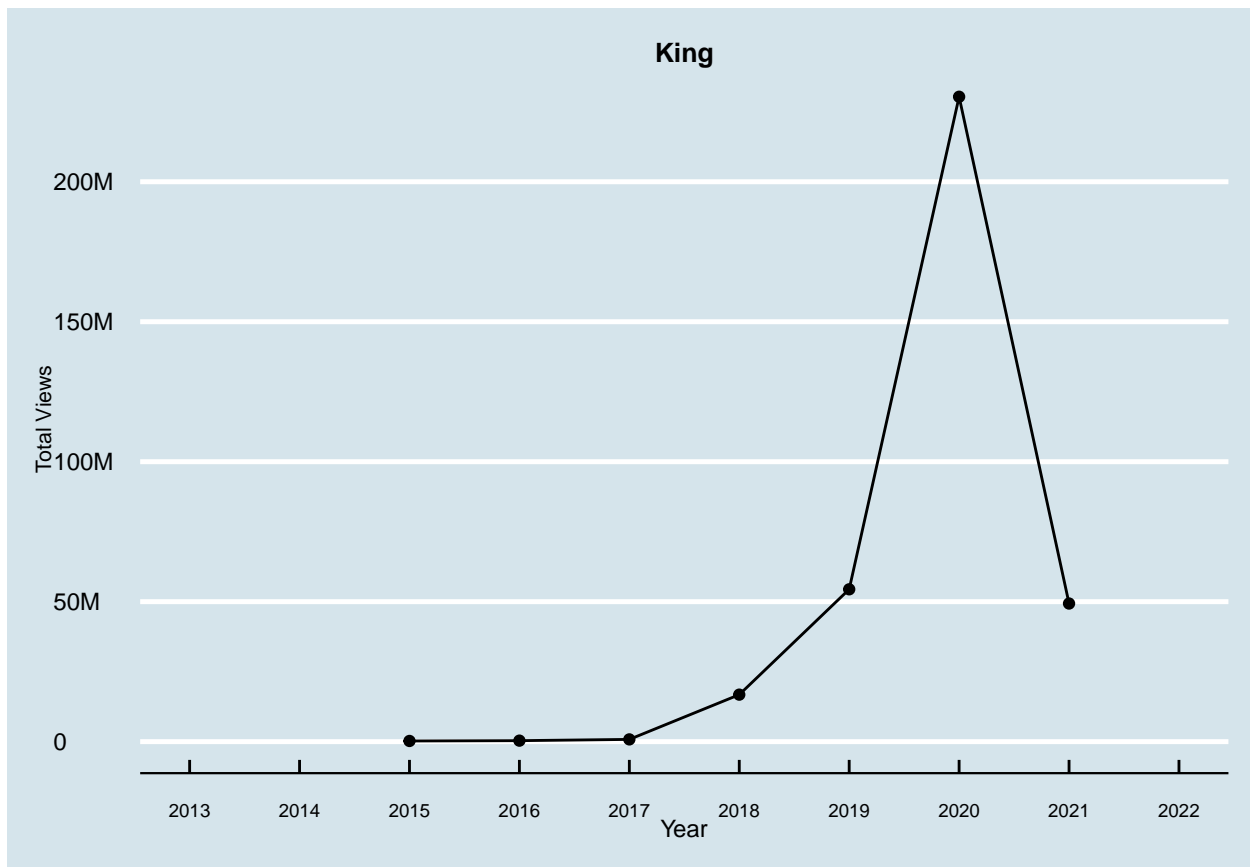
```

```

labs(title = "King",
     y = "Total Views",
     x = "Year") +
theme_economist() +

theme(plot.title = element_text(size = 10,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5),
      axis.title.x = element_text(size = 9),
      axis.title.y = element_text(size = 8),
      axis.text.x = element_text(size = 7),
      axis.text.y = element_text(size = 9))
plot(king_lg)

```



Representing all line graphs in the above section in one graph

```

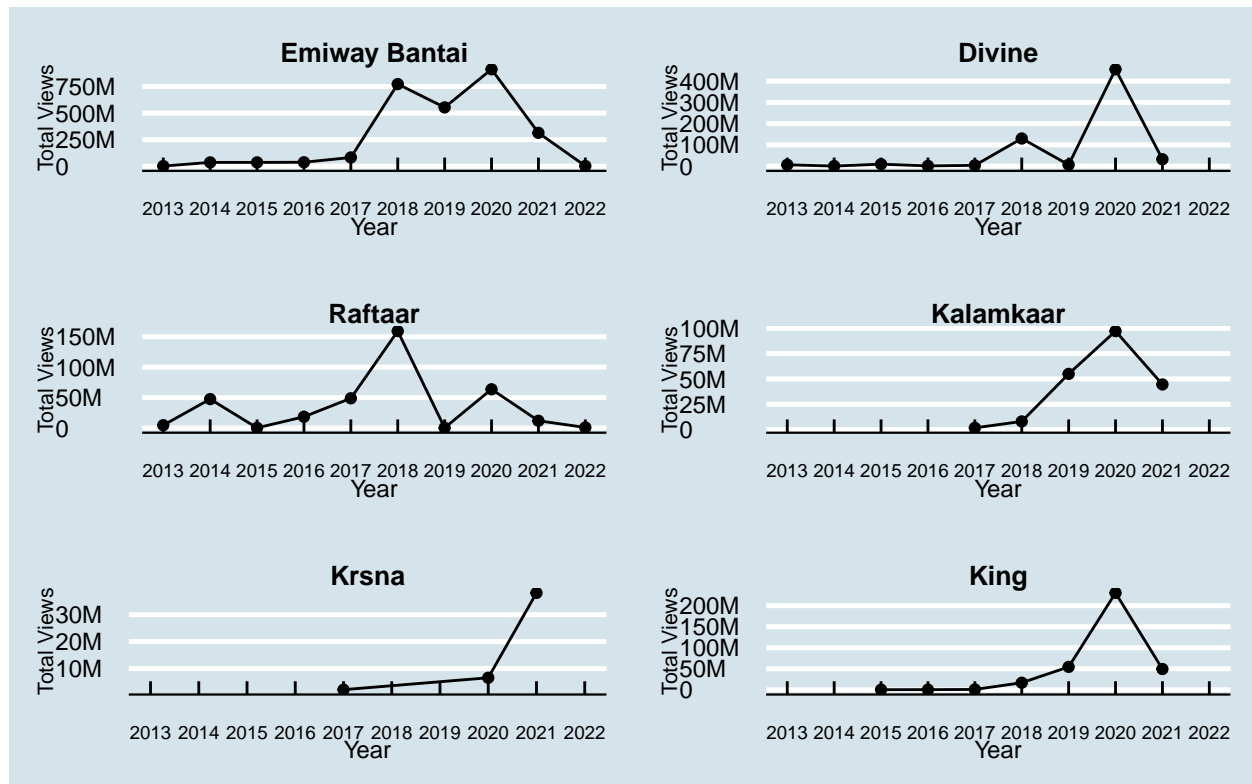
plots_two <- plot_grid(emiway_lg, divine_lg, raftaar_lg, kalamkaar_lg, krsna_lg, king_lg, nrow = 3)

title <- ggdraw() + draw_label("Views per year by Artists", fontface='bold')

plot_grid(title, plots_two, ncol=1, rel_heights=c(0.1, 1))

```

Views per year by Artists



- 1) Emiway Bantai: Since the start of the channel, the views on this channels have increased until 2018. After 2018, there was an alternate decline and growth in following years. In 2022, maybe we will see an upsurge.
- 2) Divine: Since the dawn of the channel, the view counts remained constant and similar to “Emiway Bantai’s” channel, there was an an alternate drop and hike in following years. In 2020, there was a sudden escalation but in the next year the views plummeted sharply.
- 3) Raftaar: High fluctuations were observed. 2018 was the best in terms of views count.
- 4) Kalamkaar; The channel is performing good in terms of views count. The views have always increased since the channel started except it declined in the year 2021.
- 5) Krsna: This channel hasn’t seen a downfall yet. The views kept on increasing through these years and it is performing very well when we take only views into consideration.
- 6) King: Same as Kalamkaar, the views kept on improving through all the years except for 2021, where there wasa sudden drop.

Top 10 songs by channels based on views

Emiway Bantai

```
## selecting and filtering rows and columns required
```

```

emiway_top_ten <- video_stats %>%
  select(Channel_title, Song_Name, Views) %>%
  filter(Channel_title == "Emiway Bantai") %>%
  arrange(desc(Views)) %>%
  slice(1:10) %>%
  mutate(Song_Name = fct_reorder(Song_Name, Views, .desc = FALSE))

## Plotting bar plots indicating top ten songs

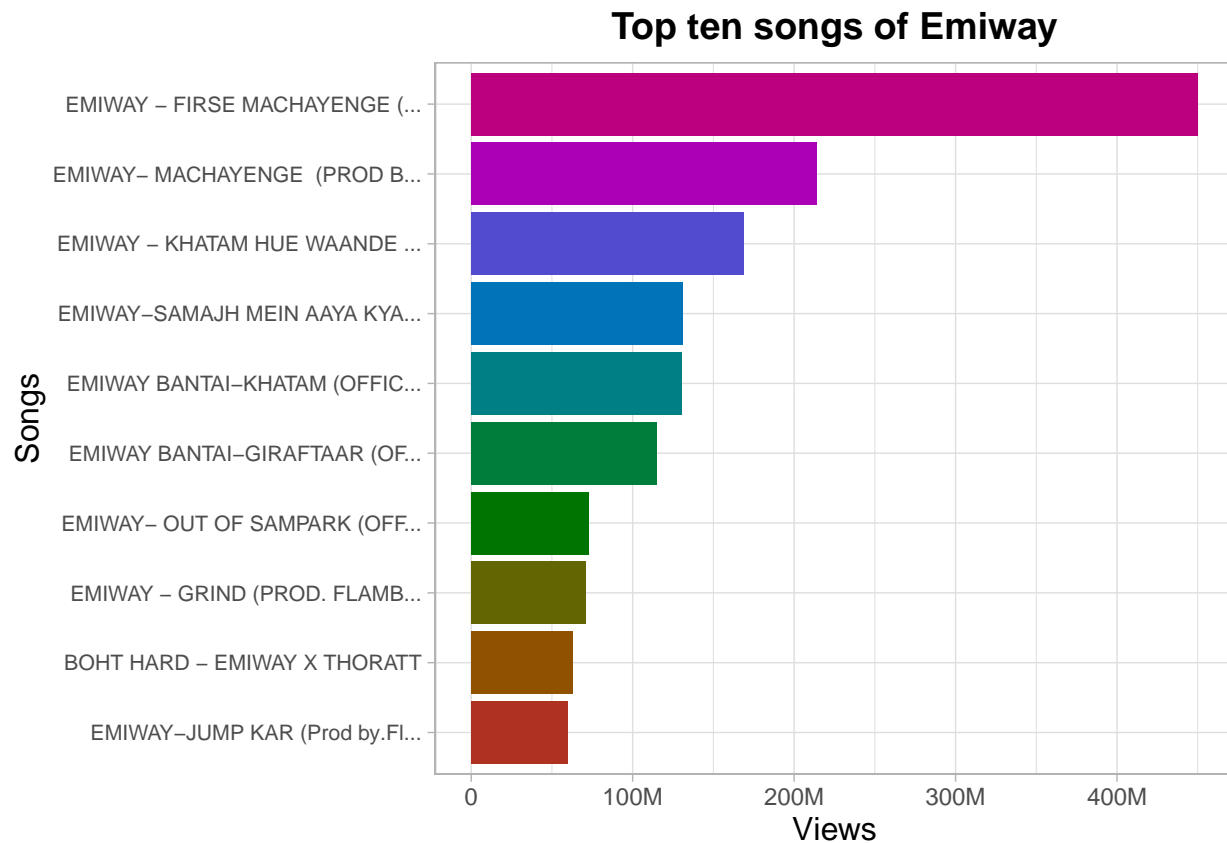
emiway_top_songs <- ggplot(emiway_top_ten, aes(x= Views, y =Song_Name, fill = Song_Name))

emiway_top_songs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_x_continuous(labels = scale_units) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 30))+
  guides(fill = guide_legend(reverse = T))+

  labs(fill = "Songs",
        title = "Top ten songs of Emiway",
        x = "Views",
        y = "Songs") +
  theme_light() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5
),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 8),
        legend.title = element_text(size = 14,
                                      face = "bold",
                                      colour = "black"),
        legend.position = "none")

```



Divine

```
## selecting and filtering rows and columns required

divine_top_ten <- video_stats %>%
  select(Channel_title, Song_Name, Views) %>%
  filter(Channel_title == "DIVINE") %>%
  arrange(desc(Views)) %>%
  slice(1:10) %>%
  mutate(Song_Name = fct_reorder(Song_Name, Views, .desc = FALSE))

## Plotting bar plots indicating top ten songs

divine_top_songs <- ggplot(divine_top_ten, aes(x= Views, y =Song_Name, fill = Song_Name))

divine_top_songs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_x_continuous(labels = scale_units) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 30))+
  guides(fill = guide_legend(reverse = T))+

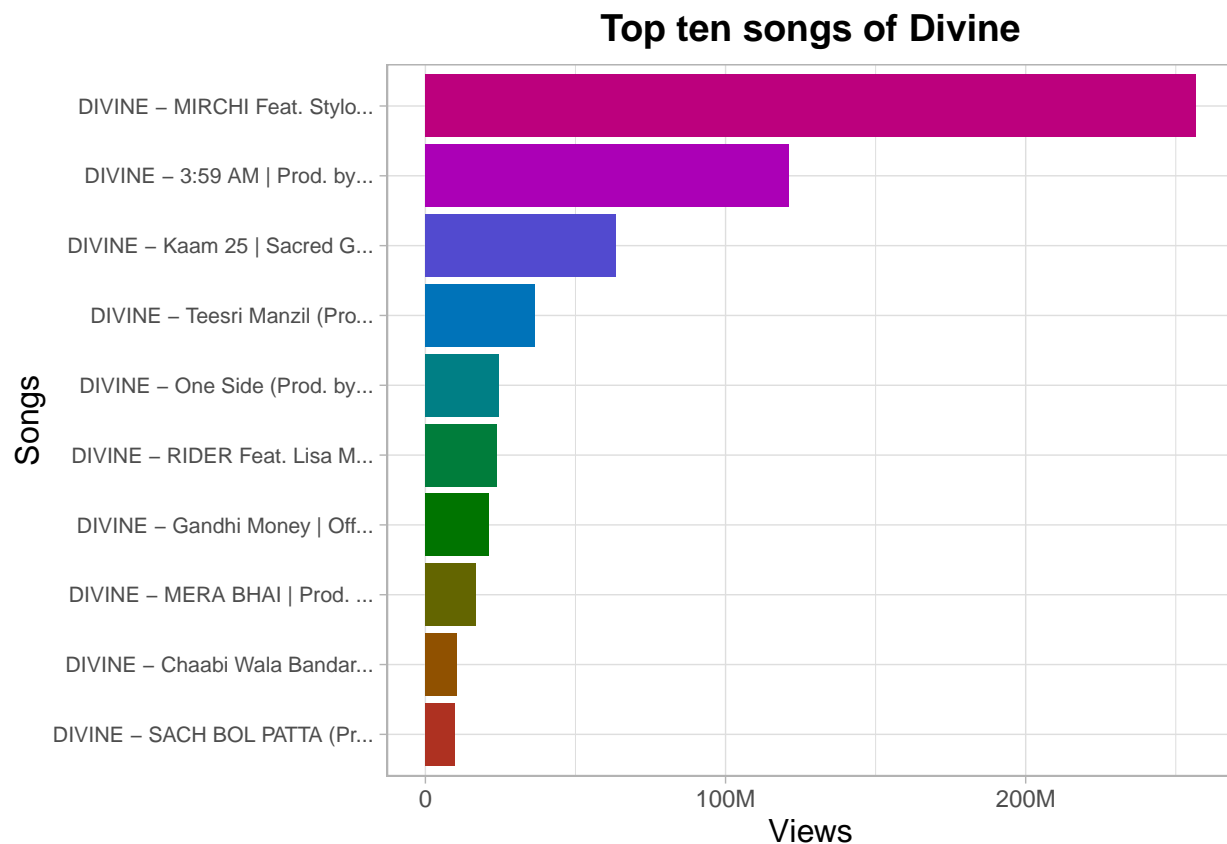
  labs(fill = "Songs",
        title = "Top ten songs of Divine",
        x = "Views",
```

```

y = "Songs") +
theme_light() +

theme(plot.title = element_text(size = 14,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5
),
axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12),
axis.text.x = element_text(size = 9),
axis.text.y = element_text(size = 8),
legend.title = element_text(size = 14,
                              face = "bold",
                              colour = "black"),
legend.position = "none")

```



Raftaar

```

## selecting and filtering rows and columns required

raftaar_top_ten <- video_stats %>%

```

```

select(Channel_title, Song_Name, Views) %>%
filter(Channel_title == "Raftaar") %>%
arrange(desc(Views)) %>%
slice(1:10) %>%
mutate(Song_Name = fct_reorder(Song_Name, Views, .desc = FALSE))

## Plotting bar plots indicating top ten songs

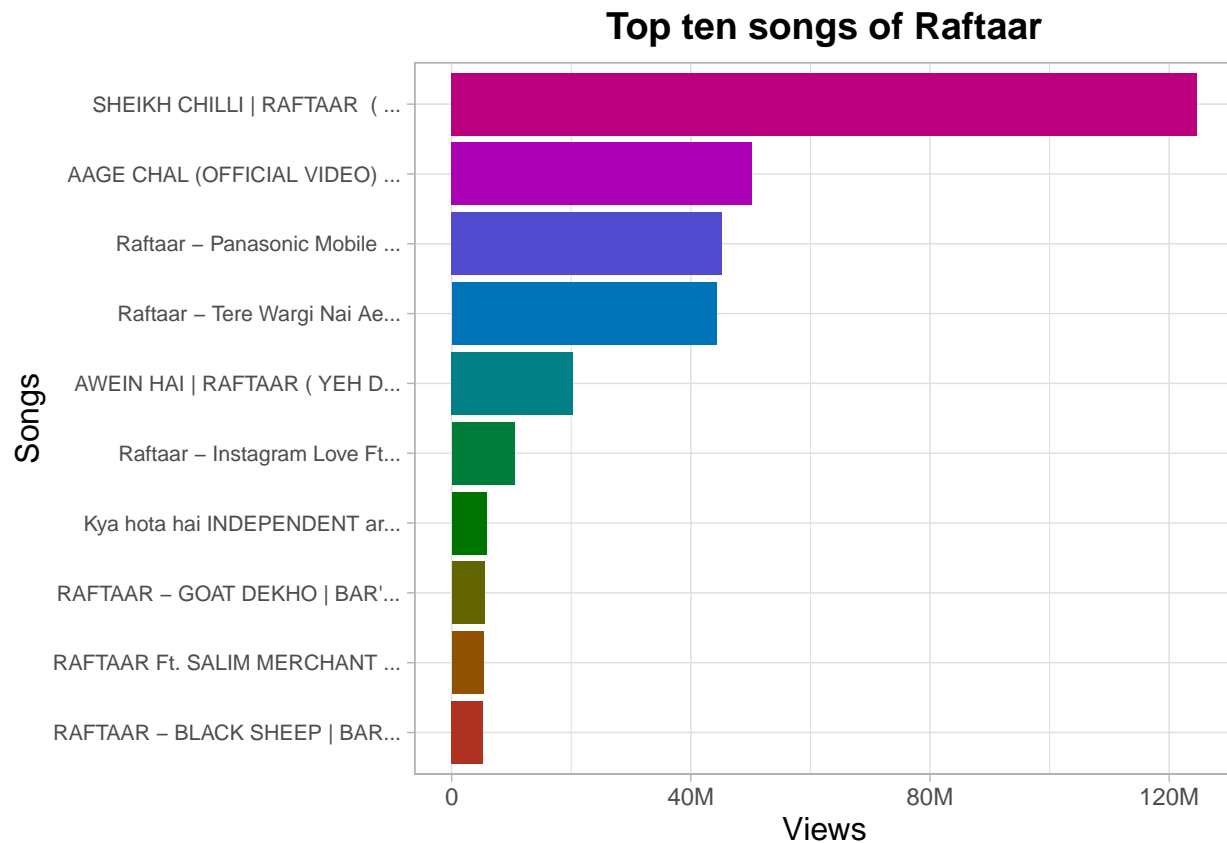
raftaar_top_songs <- ggplot(raftaar_top_ten, aes(x= Views, y =Song_Name, fill = Song_Name))

raftaar_top_songs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_x_continuous(labels = scale_units) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 30))+
  guides(fill = guide_legend(reverse = T))+

  labs(fill = "Songs",
        title = "Top ten songs of Raftaar",
        x = "Views",
        y = "Songs") +
  theme_light() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5
),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 8),
        legend.title = element_text(size = 14,
                                      face = "bold",
                                      colour = "black"),
        legend.position = "none")

```

Kalamkaar

```
## selecting and filtering rows and columns required

kalamkaar_top_ten <- video_stats %>%
  select(Channel_title, Song_Name, Views) %>%
  filter(Channel_title == "Kalamkaar") %>%
  arrange(desc(Views)) %>%
  slice(1:10) %>%
  mutate(Song_Name = fct_reorder(Song_Name, Views, .desc = FALSE))

## Plotting bar plots indicating top ten songs

kalamkaar_top_songs <- ggplot(kalamkaar_top_ten, aes(x= Views, y =Song_Name, fill = Song_Name))

kalamkaar_top_songs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_x_continuous(labels = scale_units) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 55))+
  guides(fill = guide_legend(reverse = T))+

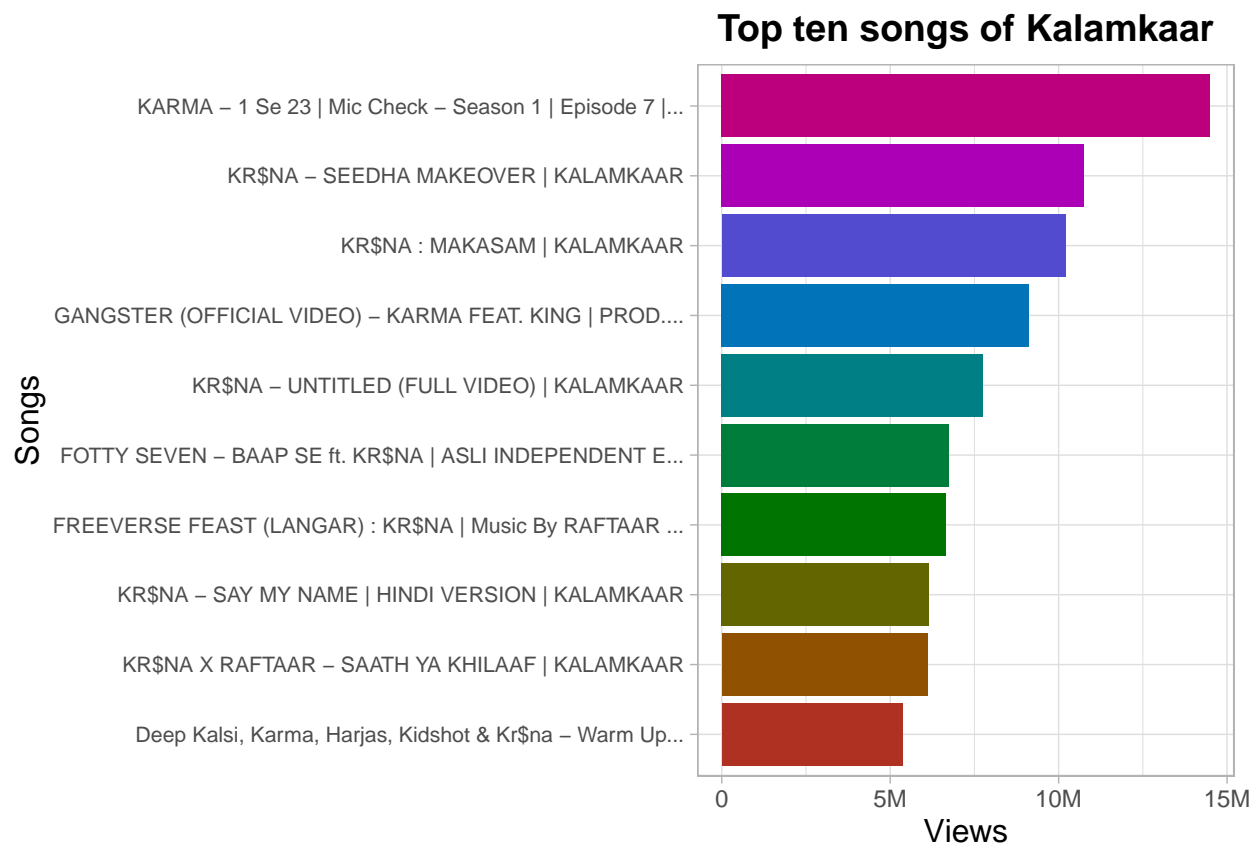
  labs(fill = "Songs",
        title = "Top ten songs of Kalamkaar",
        x = "Views",
```

```

    y = "Songs") +
  theme_light() +

  theme(plot.title = element_text(size = 14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5
    ),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    axis.text.x = element_text(size = 9),
    axis.text.y = element_text(size = 8),
    legend.title = element_text(size = 14,
                                   face = "bold",
                                   colour = "black"),
    legend.position = "none")

```



Krnsa

```

## selecting and filtering rows and columns required

krsna_top_ten <- video_stats %>%

```

```

select(Channel_title, Song_Name, Views) %>%
filter(Channel_title == "KRSNA") %>%
arrange(desc(Views)) %>%
slice(1:10) %>%
mutate(Song_Name = fct_reorder(Song_Name, Views, .desc = FALSE))

## Plotting bar plots indicating top ten songs

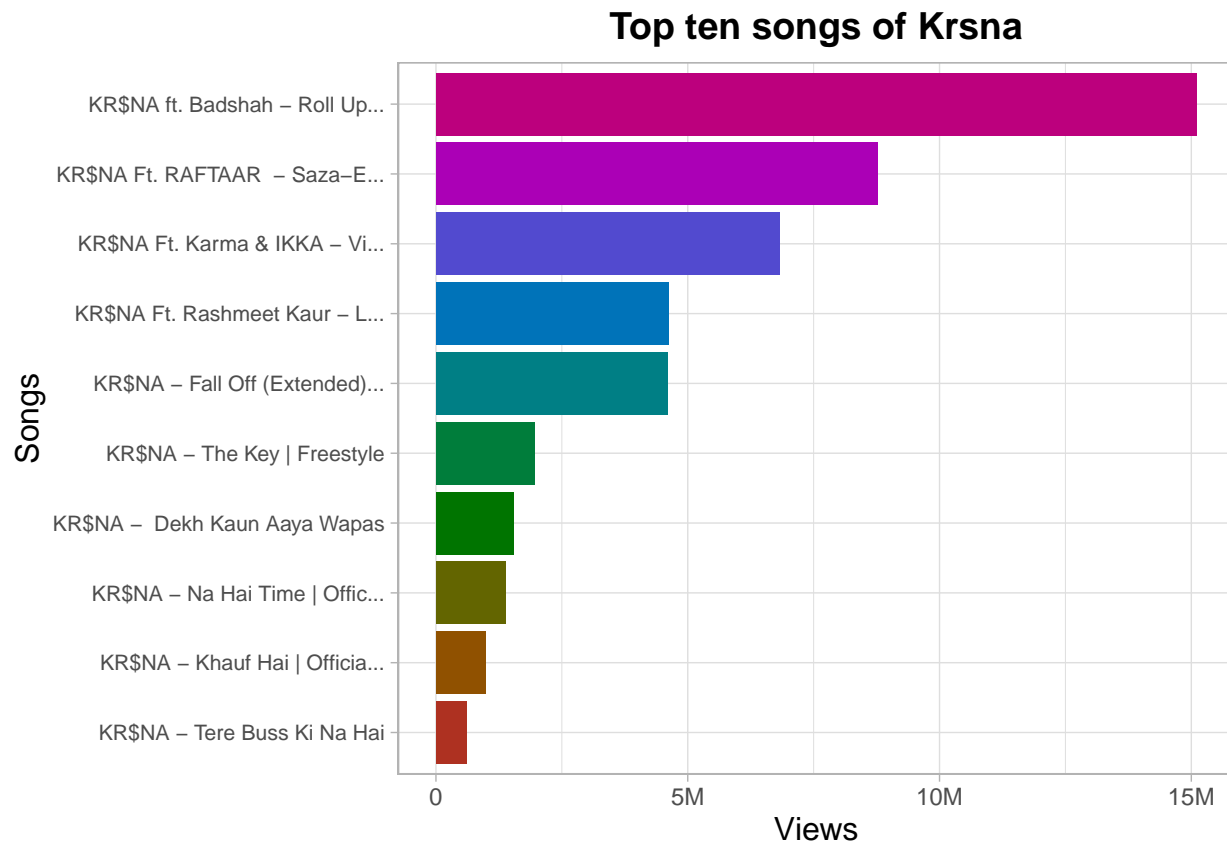
krsna_top_songs <- ggplot(krsna_top_ten, aes(x= Views, y =Song_Name, fill = Song_Name))

krsna_top_songs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_x_continuous(labels = scale_units) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 30))+
  guides(fill = guide_legend(reverse = T))+

  labs(fill = "Songs",
        title = "Top ten songs of Krsna",
        x = "Views",
        y = "Songs") +
  theme_light() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5
),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 8),
        legend.title = element_text(size = 14,
                                      face = "bold",
                                      colour = "black"),
        legend.position = "none")

```



King

```
## selecting and filtering rows and columns required

king_top_ten <- video_stats %>%
  select(Channel_title, Song_Name, Views) %>%
  filter(Channel_title == "King") %>%
  arrange(desc(Views)) %>%
  slice(1:10) %>%
  mutate(Song_Name = fct_reorder(Song_Name, Views, .desc = FALSE))

## Plotting bar plots indicating top ten songs

king_top_songs <- ggplot(king_top_ten, aes(x= Views, y =Song_Name, fill = Song_Name))

king_top_songs + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_x_continuous(labels = scale_units) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 30))+
  guides(fill = guide_legend(reverse = T))+

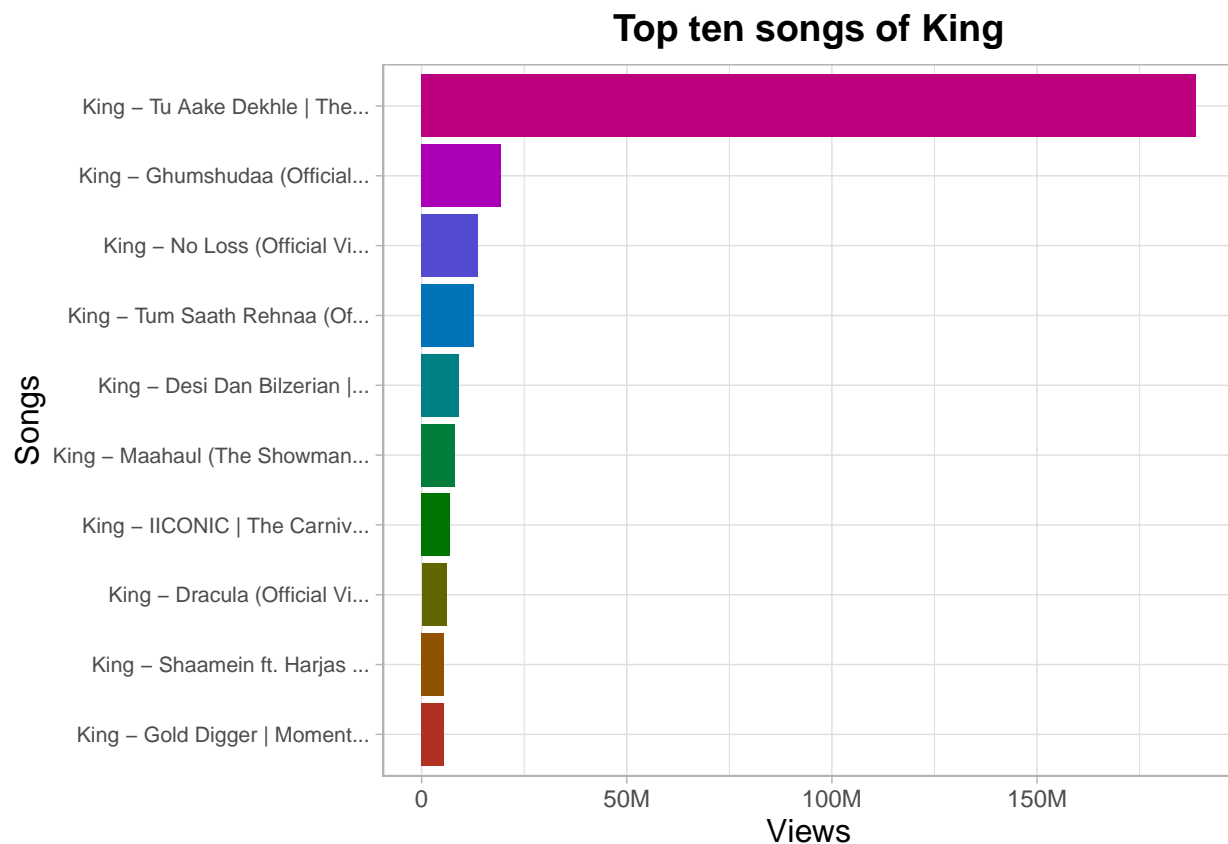
  labs(fill = "Songs",
        title = "Top ten songs of King",
        x = "Views",
```

```

y = "Songs") +
theme_light() +

theme(plot.title = element_text(size = 14,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5
),
axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12),
axis.text.x = element_text(size = 9),
axis.text.y = element_text(size = 8),
legend.title = element_text(size = 14,
                             face = "bold",
                             colour = "black"),
legend.position = "none")

```



Comparison of Average views, likes, comments of channels

Average Views of channels

```

## selecting and filtering rows and columns required
avg_views <- video_stats %>%
  group_by(Channel_title) %>%
  summarise(mean_views = mean(Views)) %>%
  mutate(Channel_title = fct_reorder(Channel_title, mean_views, .desc = TRUE))

## Plotting bar graph representing the comparison of average views of each channel

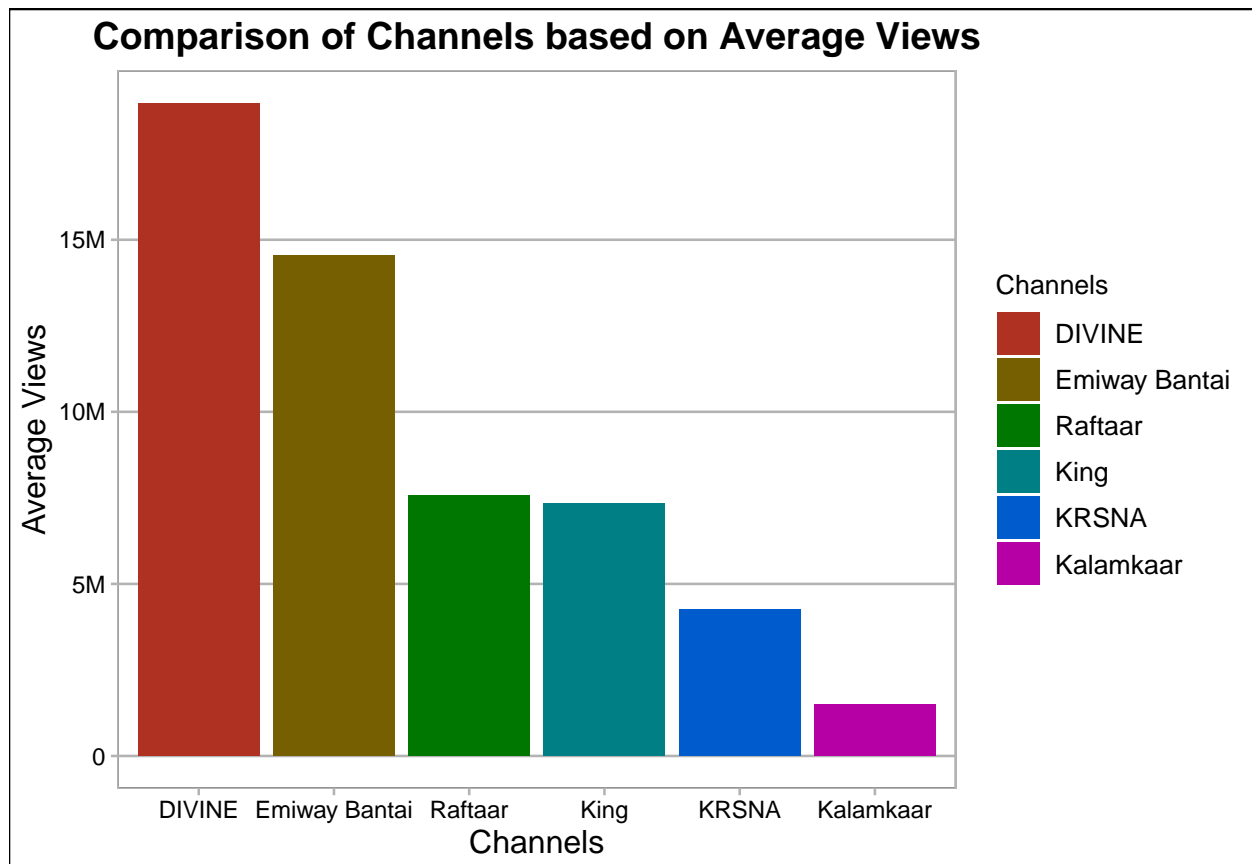
bar_mean_views <- ggplot(avg_views, aes(x= Channel_title, y= mean_views, fill =Channel_title))

bar_mean_views + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_y_continuous(labels = scale_units) +

  labs(fill = "Channels",
        title = "Comparison of Channels based on Average Views",
        x = "Channels",
        y = "Average Views") +
  theme_calc() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9))

```



“Divine” has more average views as compared to other channels. There isn’t much difference between “Raftaar” and “King” when it comes to average views. Interestingly they have uploaded same number of videos.

Average Likes of channels

```
## selecting and filtering rows and columns required
avg_likes<- video_stats %>%
  group_by(Channel_title) %>%
  summarise(mean_likes = mean(Likes)) %>%
  mutate(Channel_title = fct_reorder(Channel_title, mean_likes, .desc = TRUE))

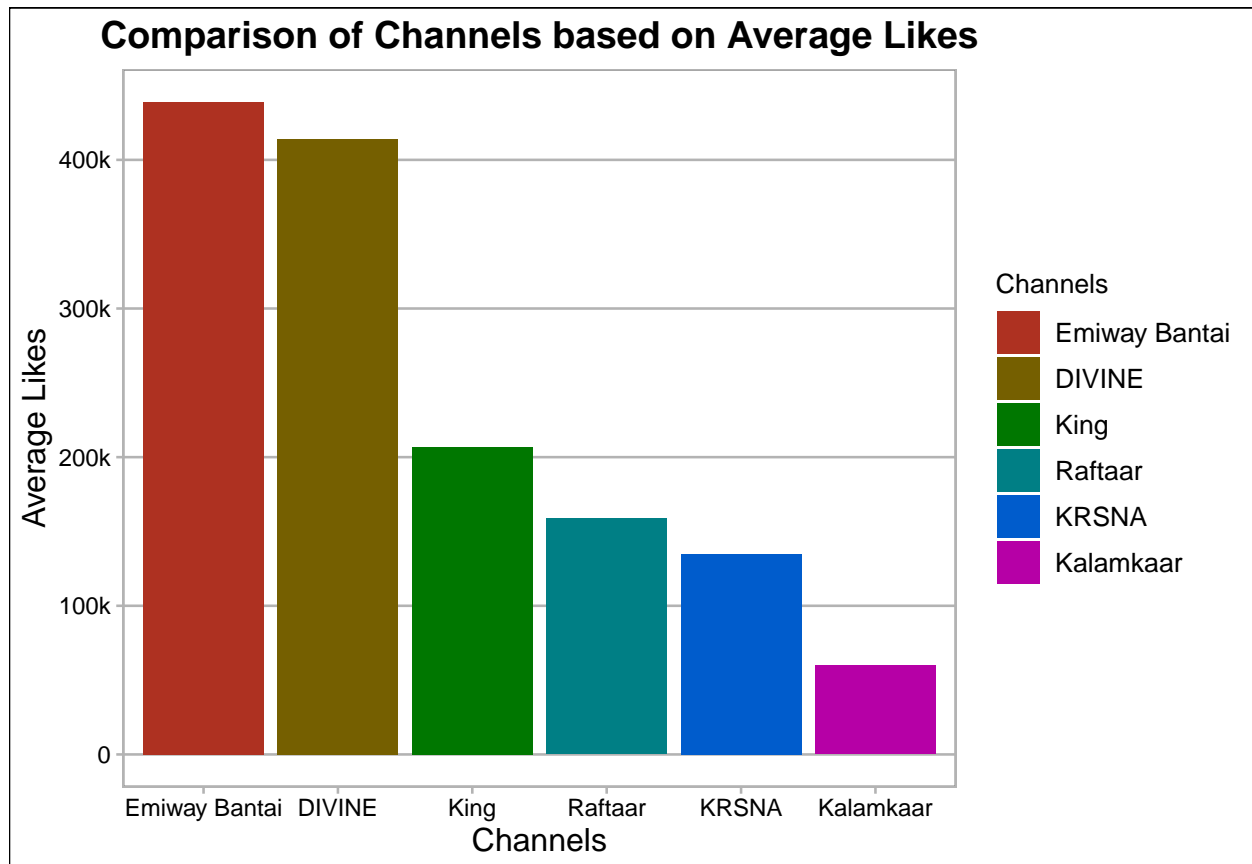
## Plotting bar graph representing the comparison of average likes of each channel

bar_mean_likes <- ggplot(avg_likes, aes(x= Channel_title, y= mean_likes, fill =Channel_title))

bar_mean_likes + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_y_continuous(labels = scale_units) +

  labs(fill = "Channels",
       title = "Comparison of Channels based on Average Likes",
       x = "Channels",
       y = "Average Likes") +
  theme_calc() +
```

```
theme(plot.title = element_text(size = 14,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9),
      axis.text.y = element_text(size = 9))
```



“Emiway Bantai” has more number of average likes and “Divine” is not far from Emiway. “King” has more average likes than “Raftaar” although they have uploaded same number of videos and their average views is also the same.

Average comments of channels

```
## selecting and filtering rows and columns required
avg_comments<- video_stats %>%
  group_by(Channel_title) %>%
  summarise(mean_comments = mean(Comments)) %>%
  mutate(Channel_title = fct_reorder(Channel_title, mean_comments, .desc = TRUE))

## Plotting bar graph representing the comparison of average comments of each channel
```



```

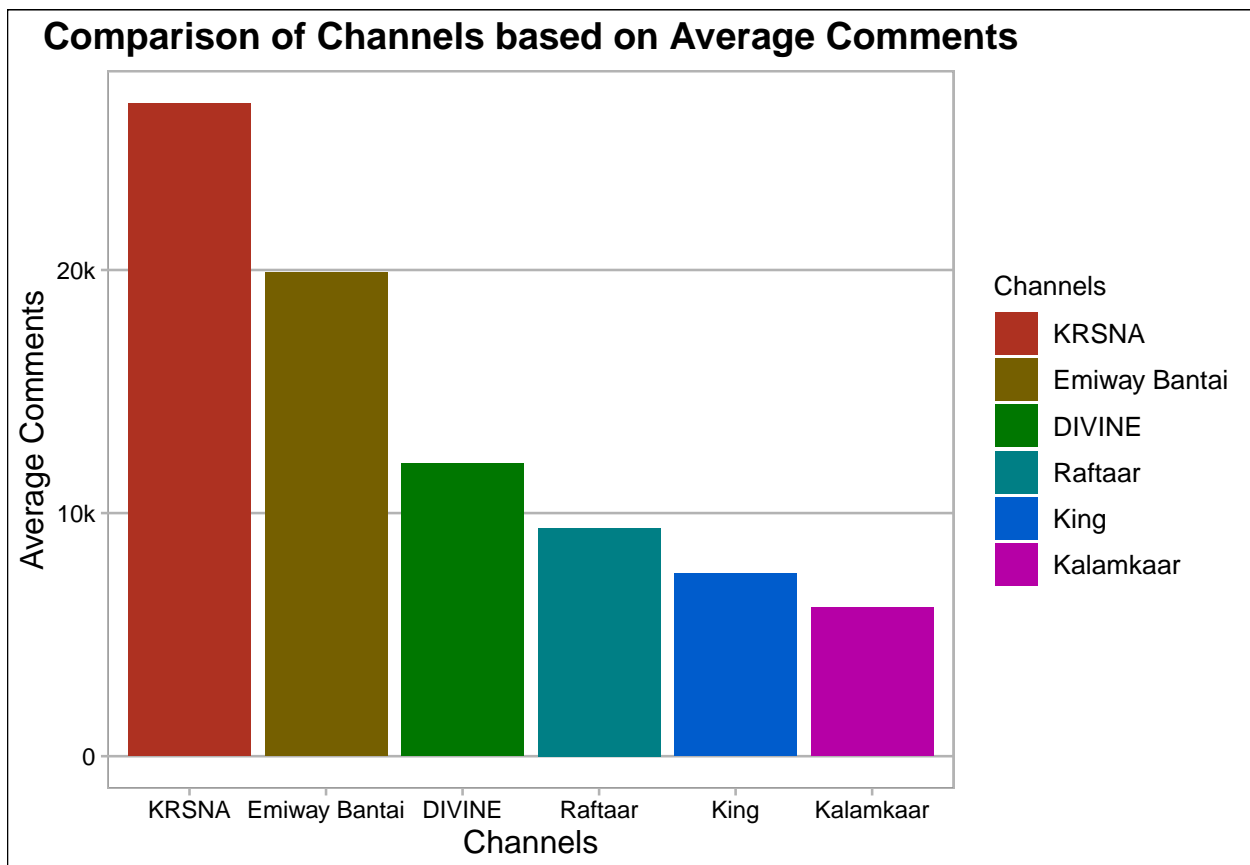
bar_mean_comments <- ggplot(avg_comments, aes(x= Channel_title, y= mean_comments, fill =Channel_title))

bar_mean_comments + geom_bar(stat = "identity") +
  scale_fill_hue(l=40) +
  scale_y_continuous(labels = scale_units) +

  labs(fill = "Channels",
        title = "Comparison of Channels based on Average Comments",
        x = "Channels",
        y = "Average Comments") +
  theme_calc() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9))

```



Interesting fact here is that Krsna has uploaded very few videos (11) as compared to other channels but still has got more average comments. The second best is “Emiway” who has uploaded 189 videos. This is

totally abrupt.

Comparison of channels based Most views, likes, comments

Most viewed videos on channels

```
## selecting and filtering rows and columns required

most_views <- video_stats %>%
  group_by(Channel_title) %>%
  summarise(max_views = max(Views)) %>%
  mutate(Channel_title = fct_reorder(Channel_title, max_views, .desc = TRUE))

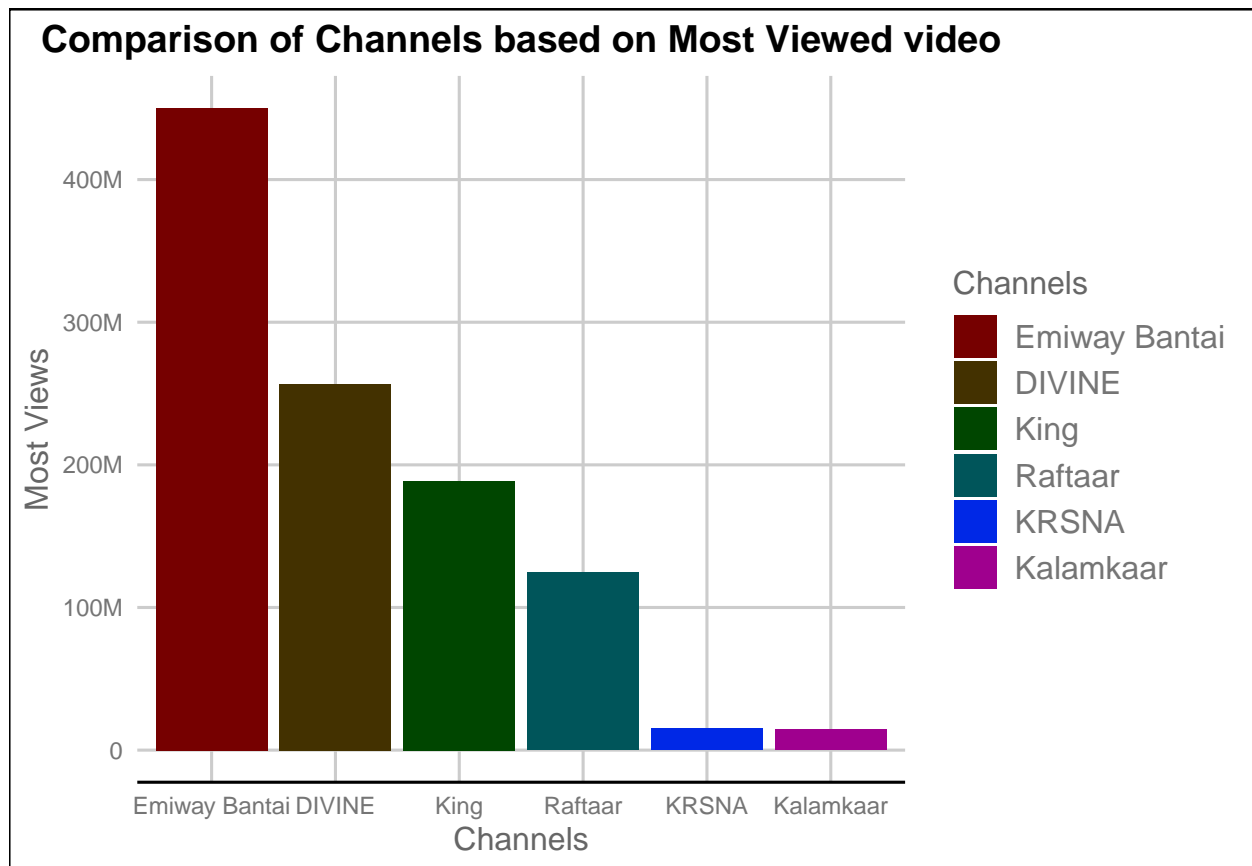
## Plotting bar graph representing the comparison of most views of each channel

bar_max_views <- ggplot(most_views, aes(x= Channel_title, y= max_views, fill =Channel_title))

bar_max_views + geom_bar(stat = "identity") +
  scale_fill_hue(l=20) +
  scale_y_continuous(labels = scale_units) +

  labs(fill = "Channels",
        title = "Comparison of Channels based on Most Viewed video",
        x = "Channels",
        y = "Most Views") +
  theme_gdocs() +

  theme(plot.title = element_text(size =14,
                                   face = "bold",
                                   colour = "black",
                                   lineheight = 1.2,
                                   hjust = 0.5),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9))
```



Most liked video on channels

```
## selecting and filtering rows and columns required

most_likes <- video_stats %>%
  group_by(Channel_title) %>%
  summarise(max_likes = max(Likes)) %>%
  mutate(Channel_title = fct_reorder(Channel_title, max_likes, .desc = TRUE))

## Plotting bar graph representing the comparison of most likes of each channel

bar_most_likes <- ggplot(most_likes, aes(x= Channel_title, y= max_likes, fill =Channel_title))

bar_most_likes + geom_bar(stat = "identity") +
  scale_fill_hue(l=20) +
  scale_y_continuous(labels = scale_units) +

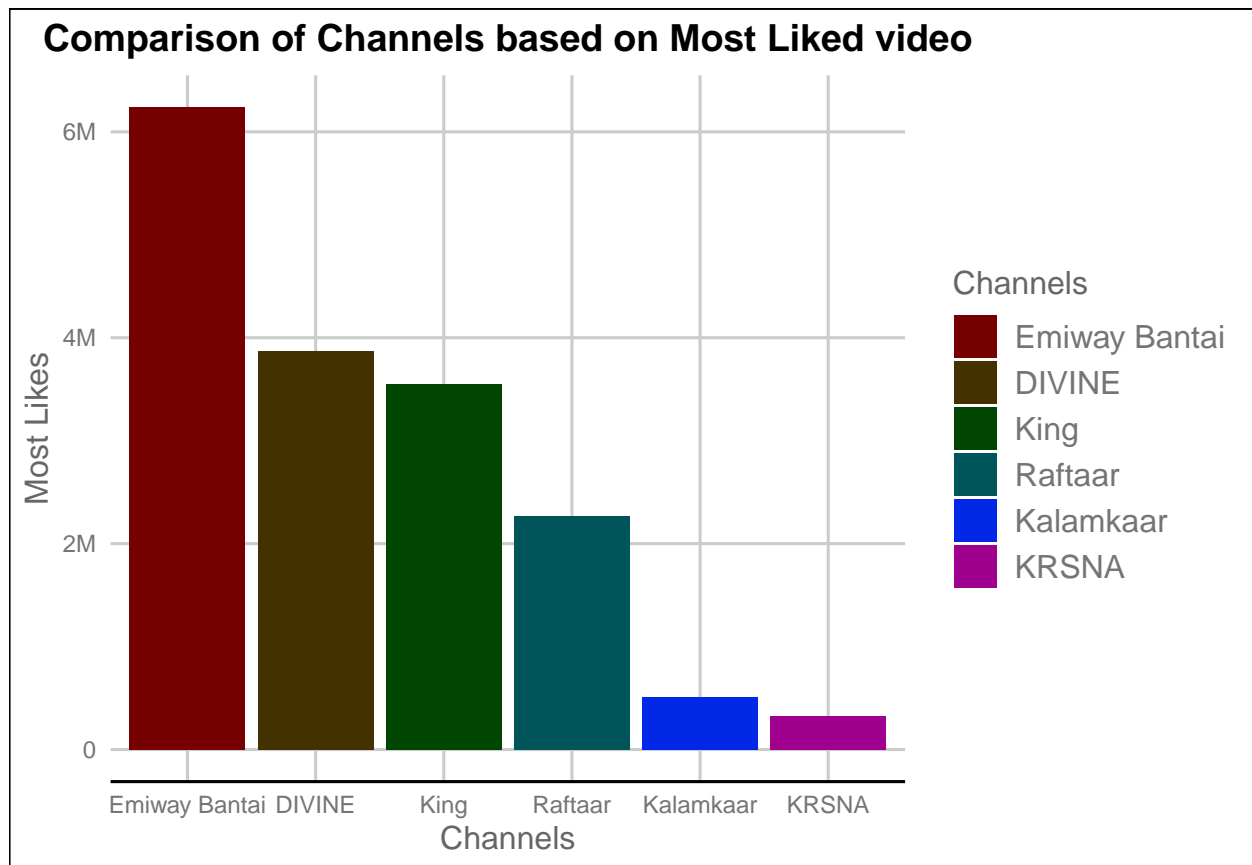
  labs(fill = "Channels",
        title = "Comparison of Channels based on Most Liked video",
        x = "Channels",
        y = "Most Likes") +
  theme_gdocs() +

  theme(plot.title = element_text(size =14,
```

```

    face = "bold",
    colour = "black",
    lineheight = 1.2,
    hjust = 0.5),
axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12),
axis.text.x = element_text(size = 9),
axis.text.y = element_text(size = 9))

```



Most commented video on channels

```

## selecting and filtering rows and columns required
most_comments <- video_stats %>%
  group_by(Channel_title) %>%
  summarise(max_comments = max(Comments)) %>%
  mutate(Channel_title = fct_reorder(Channel_title, max_comments, .desc = TRUE))

## Plotting bar graph representing the comparison of most comments of each channel

bar_most_comments <- ggplot(most_comments, aes(x= Channel_title, y= max_comments, fill =Channel_title))

bar_most_comments + geom_bar(stat = "identity") +
  scale_fill_hue(l=20) +

```

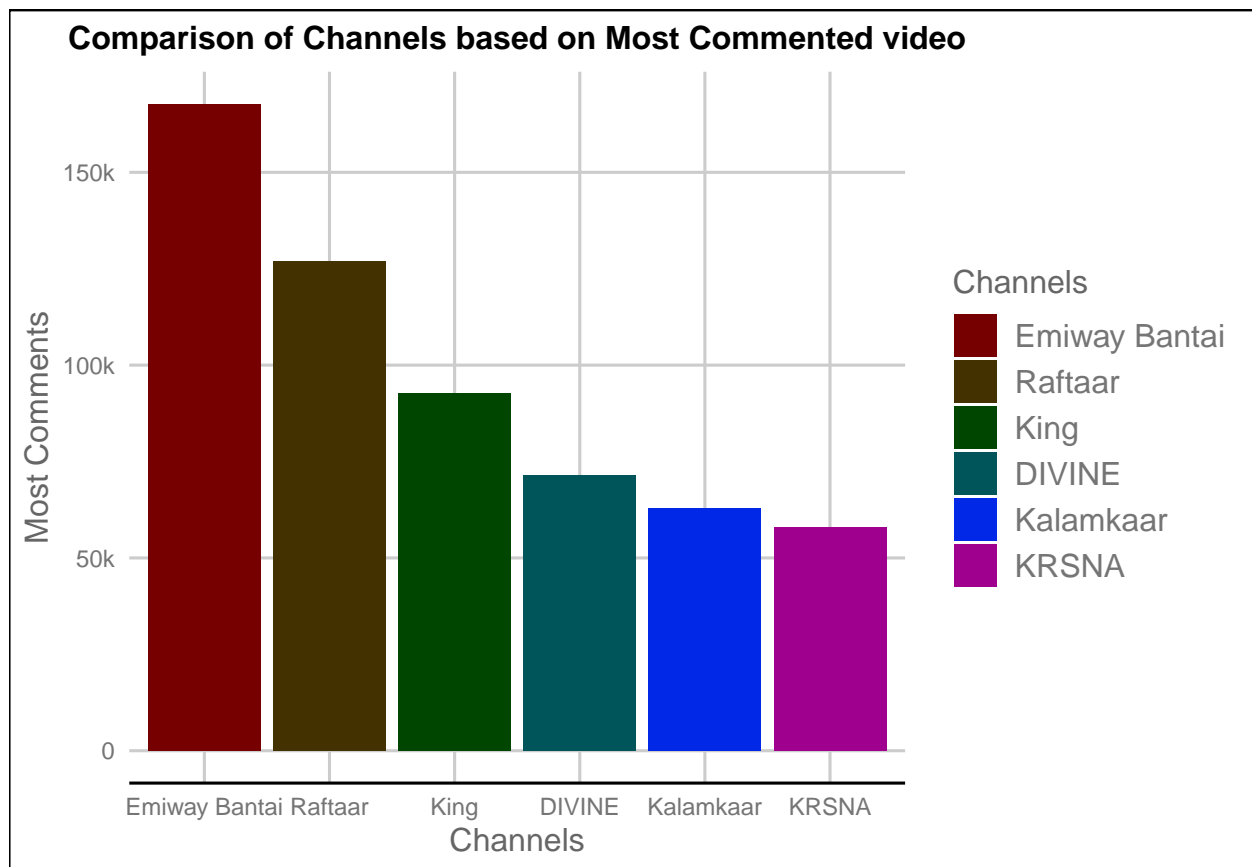
```

scale_y_continuous(labels = scale_units) +

labs(fill = "Channels",
      title = "Comparison of Channels based on Most Commented video",
      x = "Channels",
      y = "Most Comments") +
theme_gdocs() +

theme(plot.title = element_text(size = 12,
                                face = "bold",
                                colour = "black",
                                lineheight = 1.2,
                                hjust = 0.5),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9),
      axis.text.y = element_text(size = 9))

```



Conclusion:

The following conclusions are derived from a very limited data. These conclusions may not be true for other channels. These are only limited to the six channels we considered.

- 1) **Highest Number of Subscribers: Emiway Bantai**

- 2) **Highest Number of Total videos uploaded: Emiway Bantai**
- 3) **Highest Number of Total views: Emiway Bantai**
- 4) More the subscribers, more the views. They have a strong correlation 0.99.
- 5) Subscribers and views count depends on how early the channel was started
- 6) Every year “Emiway Bantai” uploads more songs than any other channels. Maybe that is why his views and subscribers are on the higher side. Rest all channels except Kalamkaar upload on an average around than 10 songs per year
- 7) Since, videos uploaded by “Emiway Bantai” are more the views per year on his channel are more too. He consistently upload videos on his channel. “Divine” has less video uploads than other channels but still has more views per year. This is an interesting point to note.
- 8) **Highest average views: “Divine”**
- 9) **Highest average likes: “Emiway”**
- 10) **Highest average comments: “Krsna”**
- 11) **Most viewed video: “From Emiway Bantai’s Channel”**
- 12) **Most liked video: “From Emiway Bantai’s Channel”**
- 13) **Most commented video: “From Emiway Bantai’s Channel”**
- 14) Though “Emiway” has highest viewed video and more subscribers but still average view count of videos of “Divine” is higher. This tells us that, the views on channels does not depend on subscriber count only. Even though there is a strong relationship between them, there may be other factors involved.