JBP061-B-6 Statistics for Data Scientists

Data-Driven Recommendations for an Aspiring Twitch Streamer: Leveraging Statistics for Success

Mikołaj Hilgert

2023-05-12

1. Table of Contents

2. Project Description

The purpose of this analysis is to evaluate the data from the top 1000 Twitch streamers in order to make strategic recommendations for a content creator looking to succeed on the platform. The up-and-coming streamer has expressed their considerations and wants to leverage the tools of statistics in order to make well-informed decisions to best model them self on the successes of the top creators.

As such the following main questions will be evaluated and discussed throughout the document:

1. Should there be a focus on mature content (18+) audience? What are the consequences one decides to do so in terms of the reaction of the audience?
2. Does such a focus on mature content lower or increase the chance of becoming a Twitch partner?
3. Is the effect of the stream time larger or smaller in mature content?

3. Data Description

Firstly, the data-set and all of the relevant libraries can be loaded.

```
twitch_data <- read.csv("twitch_data.csv")
library(knitr)
library(gridExtra)
library(ggcorrplot)
library(sjPlot)
library(tidyverse)
```

We will first inspect the raw data set, such that we can take a look at the data we are working with.

```
data_dim <- dim(twitch_data)
cat("This dataset has", data_dim[1], "rows and", data_dim[2], "columns.")
```

```
This dataset has 1000 rows and 11 columns.
```

Given we now know the dimension of our data, we can inspect the first 5 rows, to be able to get a small insight into what kind of variables we have access to.

```r
head(twitch_data)
```

Channel	Watch.time.Minutes.	Stream.time.minutes.	Peak.viewers	Average.viewers
xQcOW	6196161750	215250	222720	27716
summit1g	6091677300	211845	310998	25610
Gaules	5644590915	515280	387315	10976
ESL_CSGO	3970318140	517740	300575	7714
Tfue	3671000070	123660	285644	29602
Asmongold	3668799075	82260	263720	42414

Followers	Followers.gained	Views.gained	Partnered	Mature	Language
3246298	1734810	93036735	True	False	English
5310163	1370184	89705964	True	False	English
1767635	1023779	102611607	True	True	Portuguese
3944850	703986	106546942	True	False	English
8938903	2068424	78998587	True	False	English
1563438	554201	61715781	True	False	English

The data contains various metrics related to the individual channels, such as watch time in minutes, stream time in minutes, peak viewers, average viewers, followers gained, views gained, and other non-numerical characteristics like twitch partner status, content maturity, language and their channel name.

Next, we check if there are any missing/null values present in our data set. If there is any, we will have to deal with the missing values accordingly.

```r
any(sapply(twitch_data, is.null))
```

```
[1] FALSE
```

Luckily, there is no missing data in any of the rows. As such, this means that we do not have to do any preliminary data cleaning.

Finally however, some of the raw column names are in need of renaming for more consistency. With this, we also transform the minute metrics into hours to aid interpretability.

```r
twitch_data <- twitch_data %>%
  rename(Watch.time.hours = Watch.time.Minutes.,
         Stream.time.hours = Stream.time.minutes.,
```

```
        Followers.delta = Followers.gained) %>%
  mutate(
    Watch.time.hours = Watch.time.hours / 60,
    Stream.time.hours = Stream.time.hours / 60
  )
```

4. Exploratory Data Analysis (EDA)

As a first step, we can gain some initial insight into the data by looking at the statistical summary.

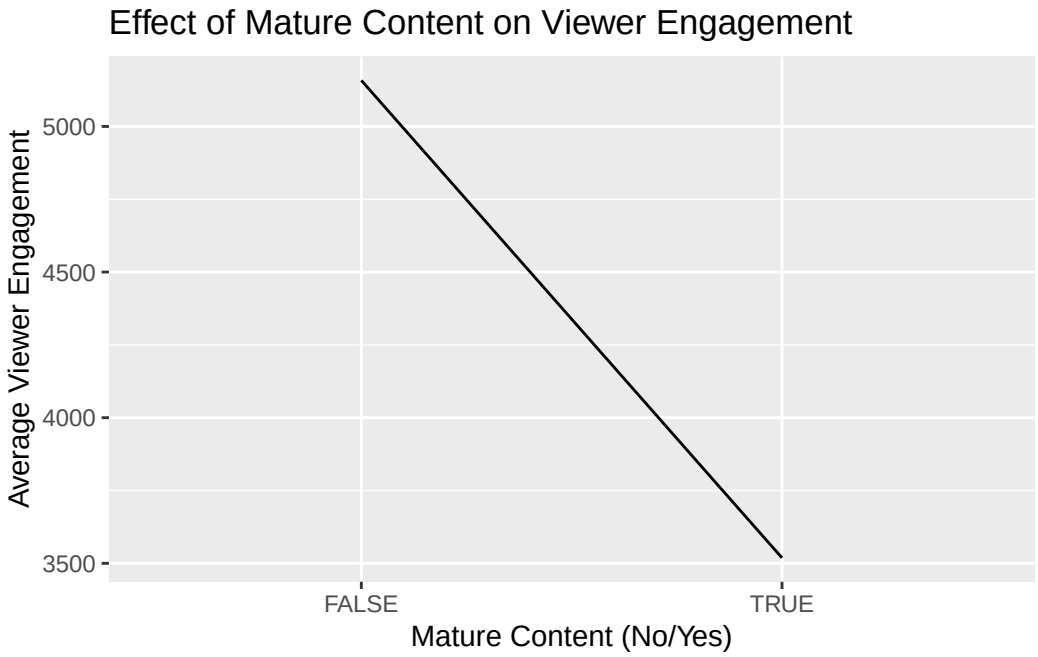
```
summary(twitch_data)
```

```
   Channel           Watch.time.hours    Stream.time.hours  Peak.viewers
 Length:1000        Min.   :  2036548   Min.   :  57.75   Min.   :   496
 Class :character   1st Qu.:  2719832   1st Qu.:1229.31   1st Qu.:  9114
 Mode  :character   Median :  3916513   Median :1804.00   Median : 16676
                    Mean   :  6973799   Mean   :2008.59   Mean   : 37065
                    3rd Qu.:  7228999   3rd Qu.:2364.06   3rd Qu.: 37570
                    Max.   :103269362   Max.   :8690.75   Max.   :639375
 Average.viewers    Followers       Followers.delta    Views.gained
 Min.   :   235   Min.   :   3660   Min.   : -15772   Min.   :   175788
 1st Qu.:  1458   1st Qu.: 170546   1st Qu.:  43758   1st Qu.:  3880602
 Median :  2425   Median : 318063   Median :  98352   Median :  6456324
 Mean   :  4781   Mean   : 570054   Mean   : 205519   Mean   : 11668166
 3rd Qu.:  4786   3rd Qu.: 624332   3rd Qu.: 236131   3rd Qu.: 12196762
 Max.   :147643   Max.   :8938903   Max.   :3966525   Max.   :670137548
  Partnered            Mature            Language
 Length:1000        Length:1000        Length:1000
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

From this we can see bla bla bla.

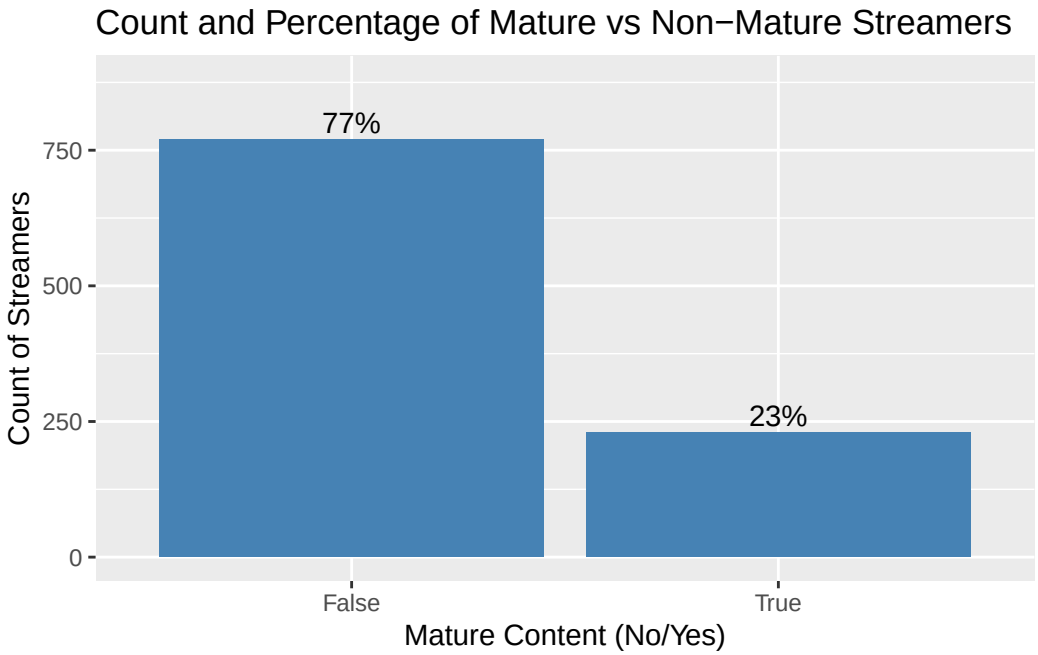
Next, To assess the effect of mature content on audience reactions, it is crucial to compare the engagement and response levels between streams with and without mature content. For this purpose, we employ a line chart or bar chart to illustrate the average viewer engagement metrics over time.

```r
# Group by 'Mature' and calculate mean of 'Average.viewers'
df_grouped <- twitch_data %>%
    group_by(Mature) %>%
    mutate(Mean_Viewers = mean(Average.viewers, na.rm = TRUE))

# Create line chart
ggplot(df_grouped, aes(x = as.logical(Mature), y = Mean_Viewers)) +
    geom_line(group = 1) +
    xlab('Mature Content (No/Yes)') +
    ylab('Average Viewer Engagement') +
    ggtitle('Effect of Mature Content on Viewer Engagement')
```

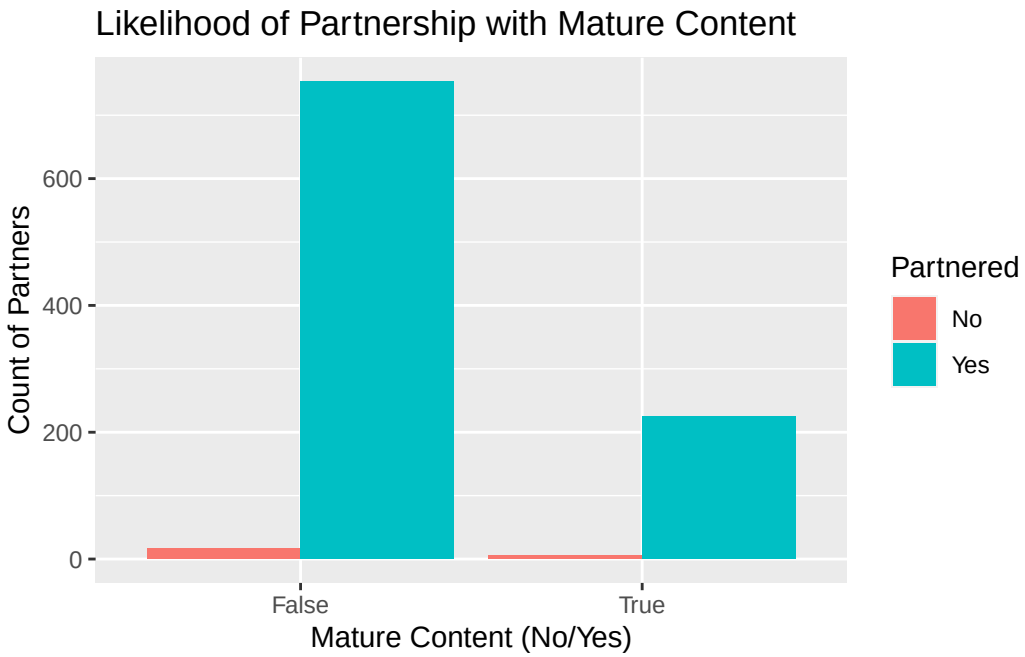



```r
# Calculate the count and percentage of each group
df_grouped <- twitch_data %>%
    group_by(Mature) %>%
    summarise(Count = n()) %>%
    mutate(Percentage = Count / sum(Count) * 100)

# Create bar chart with percentage labels
ggplot(df_grouped, aes(x = as.factor(Mature), y = Count)) +
    geom_bar(stat = 'identity', fill = 'steelblue') +
```

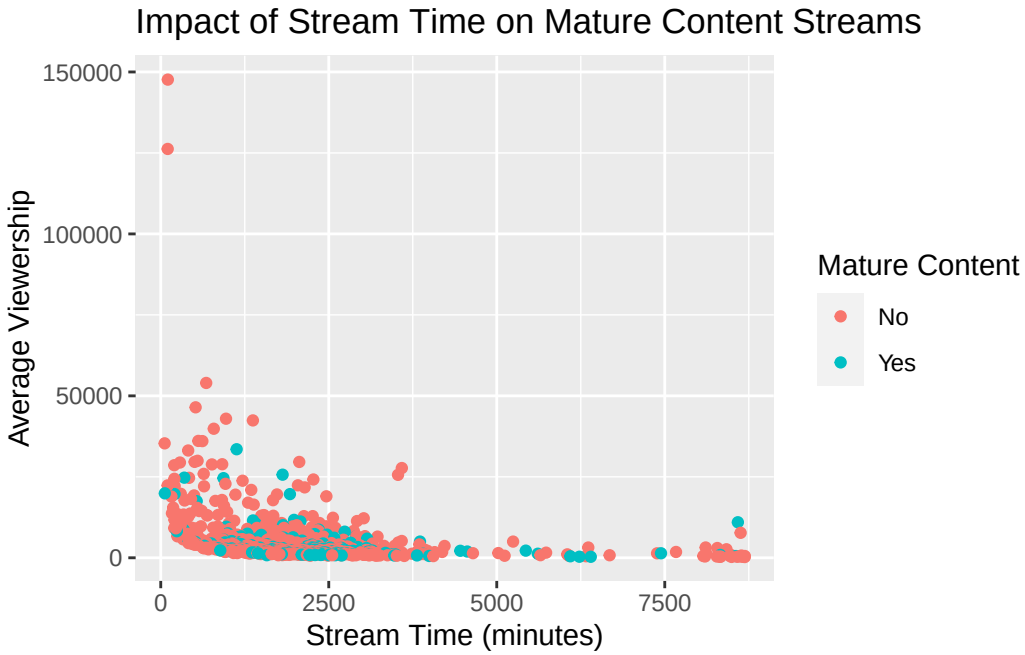
```
    geom_text(aes(label = paste0(round(Percentage, 1), "%")),
              vjust = -0.3) +
    xlab('Mature Content (No/Yes)') +
    ylab('Count of Streamers') +
    scale_y_continuous(limits = c(0, 880)) +
    ggtitle('Count and Percentage of Mature vs Non-Mature Streamers')
```




```
# Group by 'Mature' and 'Partnered' to count the number of each group
df_grouped <- twitch_data %>%
  group_by(Mature, Partnered) %>%
  summarise(Count = n(), .groups = 'drop')


# Create grouped bar chart
ggplot(df_grouped, aes(x = as.factor(Mature), y = Count,
                       fill = as.factor(Partnered))) +
    geom_bar(stat = 'identity', position = 'dodge') +
    xlab('Mature Content (No/Yes)') +
    ylab('Count of Partners') +
    ggtitle('Likelihood of Partnership with Mature Content') +
    scale_fill_discrete(name = "Partnered", labels = c("No", "Yes"))
```




```
# Create scatter plot
ggplot(twitch_data, aes(x = Stream.time.hours, y = Average.viewers,
                        color = as.factor(Mature))) +
    geom_point() +
    xlab('Stream Time (minutes)') +
    ylab('Average Viewership') +
    ggtitle('Impact of Stream Time on Mature Content Streams') +
    scale_color_discrete(name = "Mature Content", labels = c("No", "Yes"))
```


Impact of Stream Time on Mature Content Streams
Average Viewership
150000
100000
50000
0
0
2500
5000
7500
Stream Time (minutes)
Mature Content
No
Yes