JBP061-B-6 Statistics for Data Scientists

Data-Driven Recommendations for an Aspiring Twitch Streamer: Leveraging Statistics to model success

Mikołaj Hilgert

2023-06-09

1. Table of Contents

2. Project Description

The purpose of this analysis is to evaluate the data from the top 1000 Twitch streamers from the year 2020 in order to make strategic recommendations for a content creator looking to make a start on the platform. The up-and-coming streamer has expressed their considerations and wants to leverage the tools of statistics to make well informed decisions, such that they can model their journey based on the successes of 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.

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

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

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

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

Given we now know the dimension of our data, we can inspect the first 6 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. We also update column names and scale.

```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)

In this section, we set out explore the data through graphs and basic statistics. This section helps in gaining a deeper understanding of the data set. This allows us to gain some insight and potentially be able to gauge some initial trends. This section will let us also see if there are any issues with out data. Not just from the lens of missing data, but rather if our data set can be used to generalize for the population, as that is our ultimate goal.

As a first step, we can take a look at the statistical summary of the data.

```
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  
```

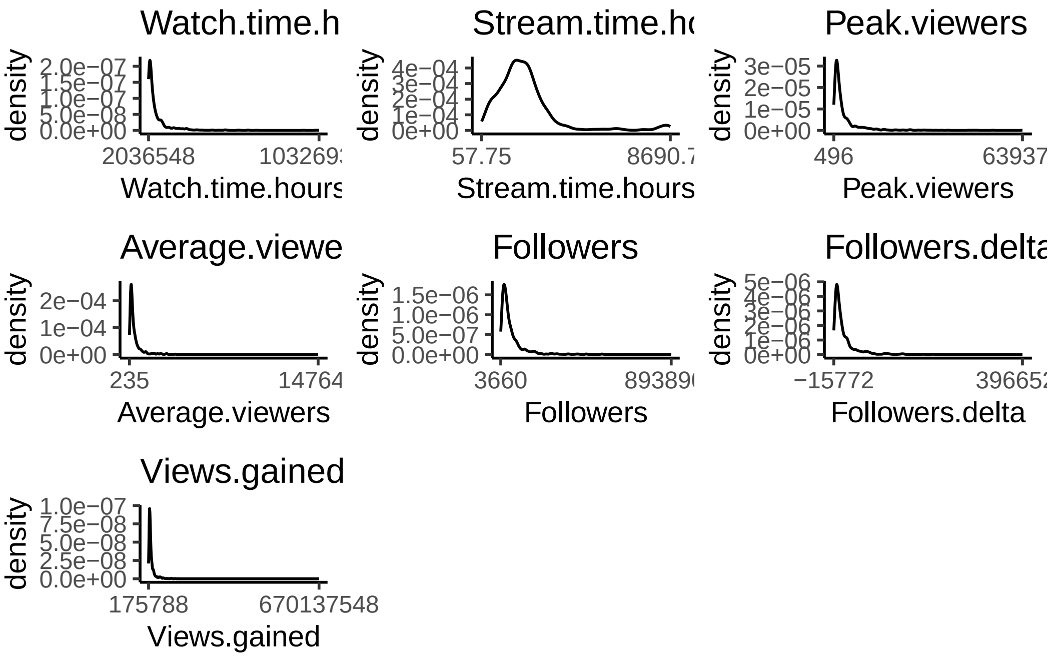
These values provide an overview of the range, central tendency, and distribution of the metrics in the data set. As mentioned in the end of the previous section, the motivation of renaming to `Followers.delta` is that we can observe that a minimum value of that column is actually negative (-15772). This implies that a streamer lost followers in total over the duration of a year.

From these summary statistics, it is apparent that the data set is skewed, as indicated by the large differences between the minimum and maximum values for most variables. Additionally, the mean values are higher than the median values for several variables, suggesting a positive skewness. We can do some data type manipulation:

```
numeric <- twitch_data%>%select_if(is.numeric)
twitch_data <- twitch_data %>%  mutate(
    Mature = as.logical(Mature), Partnered = as.logical(Partnered),
    Language = as.factor(Language) %>% relevel("English"))
```

After separating our numerical variables into their own separate variable, we may now look at the distribution of each.

```
plots <- lapply(names(numeric), function(col) {
  ggplot(numeric, aes(x = !!sym(col))) + geom_density() + ggtitle(col) +
    scale_x_continuous(breaks = c(min(numeric[[col]]), max(numeric[[col]])))})
grid.arrange(grobs = plots, ncol = 3)
```



From this, we can see that our intuition was correct, Each of the variables is largely positively skewed. As such, it can be beneficial to apply a log scale to all of these numerical variables.

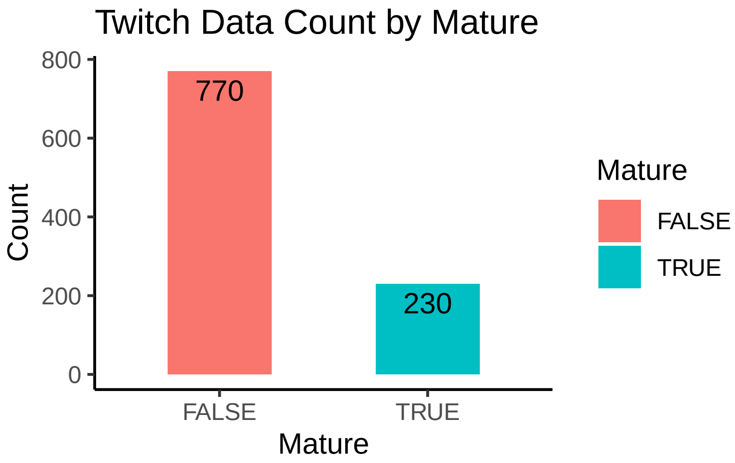
Next, To assess the effect of mature content on audience reactions, we compare the engagement and response levels between streams with and without mature content. We can evaluate the average viewer engagement (Amount of viewers on average) against the mature content rating, in order to see if there is a relationship between the two.

```
twitch_data %>%
    group_by(Mature) %>% summarise(Mean_Viewers = round(mean(Average.viewers)))
```

```
# A tibble: 2 x 2
  Mature Mean_Viewers
  <lgl>         <dbl>
1 FALSE          5158
2 TRUE           3519
```

This seems to indicate, that on average the non mature streamers in the top 1000, have more viewers on average than those who have mature content warning. This by itself is quite misleading, as we do not yet know how many streamers fall under each group. As such, we should look at the proportional representation of each group. There may be a data imbalance.

```
mature_dist <- twitch_data %>%
    group_by(Mature) %>%
    summarise(Count = n()) %>% mutate(Percentage = Count / sum(Count) * 100)
ggplot(mature_dist, aes(x = Mature, y = Count, fill = Mature)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_text(aes(label =Count),position =position_dodge(width =0.9),vjust =1.4) +
  labs(x = "Mature", y = "Count", title = "Twitch Data Count by Mature")
```

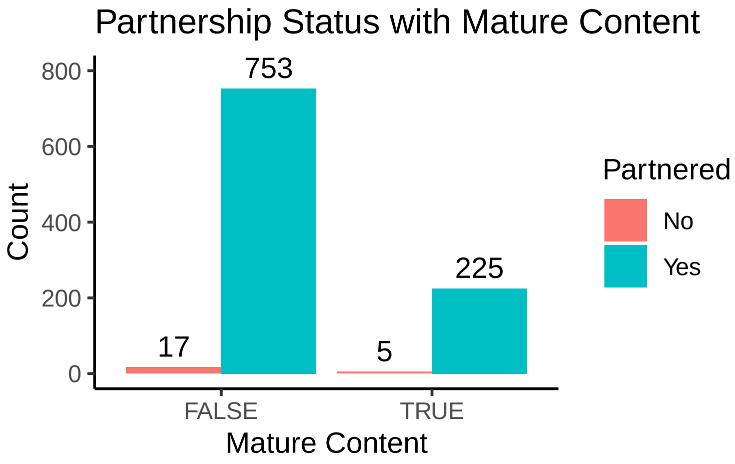


As mentioned, looking at the distribution of Mature across the data set, we can see that only

230 of our data set of the top 1000 streamers, have stream to a mature audience. This means that we have quite a unbalanced data set.

Another area of interest is the partnership status of mature channels, in the following plot, we wish to investigate the relationship between partnership status and the content they produce.

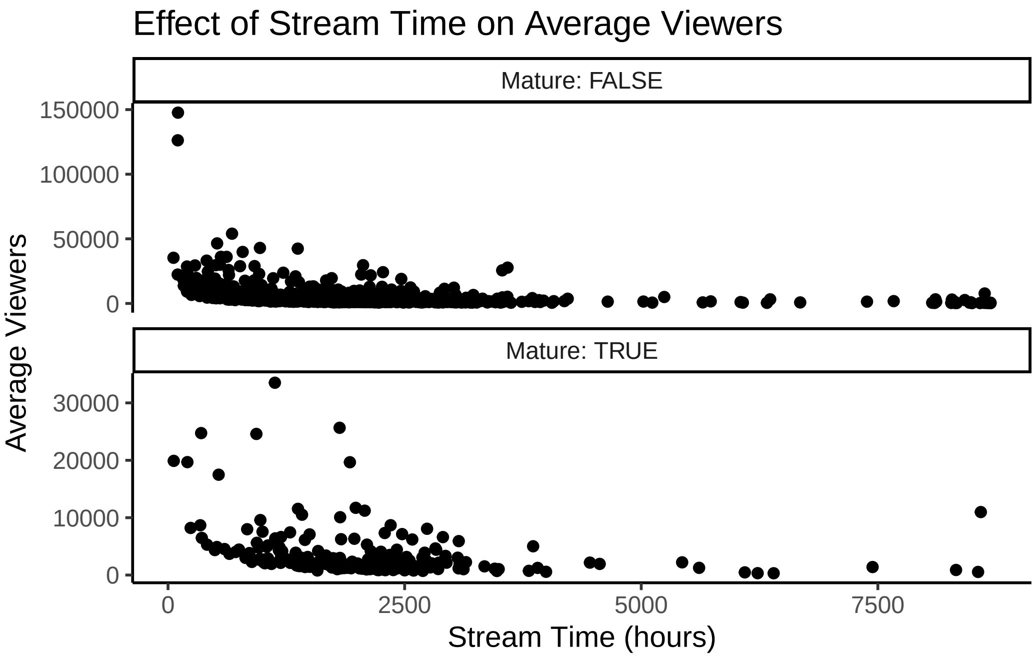
```
mature_partnered <- twitch_data %>%
  group_by(Mature, Partnered) %>% summarise(Count = n(), .groups = 'drop')

ggplot(mature_partnered, aes(x = Mature, y = Count, fill = Partnered)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  geom_text(aes(label = Count), position = position_dodge(width = 0.9),
            vjust = -0.5) +
  xlab('Mature Content') +   ylab('Count') +
  scale_y_continuous(limits = c(0, 800)) +
  ggtitle('Partnership Status with Mature Content') +
  scale_fill_discrete(name = "Partnered", labels = c("No", "Yes"))
```



From the plot above, we can see that the non partnered representation in our data-set is extremely small, namely only 22 of the 1000 channels are *not* partnered, regardless of their content maturity. This can already indicate a problem with our analysis and data set, since our data comprises of the **top** 1000 channels, they are also likely to have been partnered with Twitch, which does not necessarily accurately describe the entire population.

Finally, to gain insight for our last area of interest, specifically on the effect of stream time in mature content. We can take a look at a scatter plot of these variables.

```
ggplot(twitch_data, aes(x = Stream.time.hours, y = Average.viewers)) +
  geom_point() +
  labs(x = "Stream Time (hours)", y = "Average Viewers") +
  ggtitle("Effect of Stream Time on Average Viewers") +
  facet_wrap(~ Mature, ncol = 1,
             scales = "free_y",
             labeller = labeller(Mature = c("FALSE" = "Mature: FALSE", "TRUE" = "Mature: T
```



As can be seen in the plot above, there is a big difference in the y axis of the two classes. Moreover, the non-mature class has very large outliers. Which could be of interest. We can also notice that more stream hours doesn't really result in larger average viewership. We will investigate this further in th modelling section.

We can further investigate which channels these are:

```
twitch_data %>%
  group_by(Channel, Mature, 'Hours_streamed' = round(Stream.time.hours)) %>%
  summarise(Average_viewership = mean(Average.viewers), .groups = 'drop') %>%
  arrange(desc(Average_viewership)) %>% head(10)
```

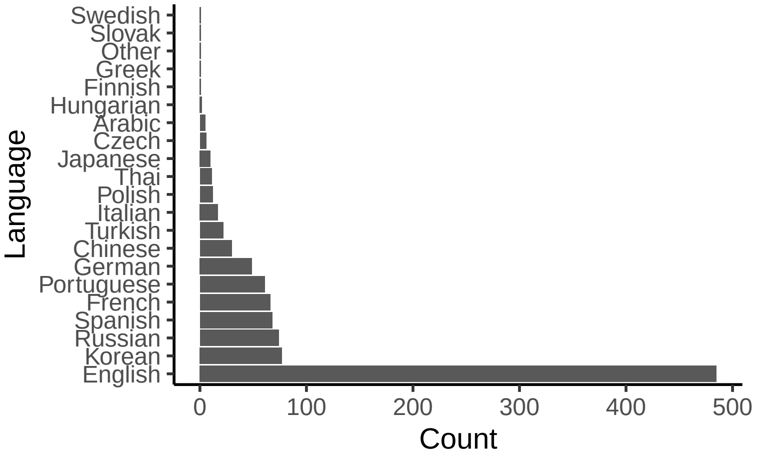
```
# A tibble: 10 x 4
```

```
   Channel     Mature Hours_streamed Average_viewership
   <chr>       <lgl>           <dbl>              <dbl>
 1 dota2ti     FALSE             105             147643
 2 dota2ti_ru  FALSE             103             126232
 3 auronplay   FALSE             676              53986
 4 LCS         FALSE             519              46459
 5 Rubius      FALSE             971              42948
 6 Asmongold   FALSE            1371              42414
 7 LCK_Korea   FALSE             789              39848
 8 RocketLeague FALSE            559              36086
 9 LCK         FALSE             619              36030
10 KEEMSTAR    FALSE              58              35333
```

Out of the top 10 ordered by average viewership, 6 of these channels are E-sports event channels. As such they are not representative of individuals streaming.

We may also take a look at the channel distribution by language using a bar plot.

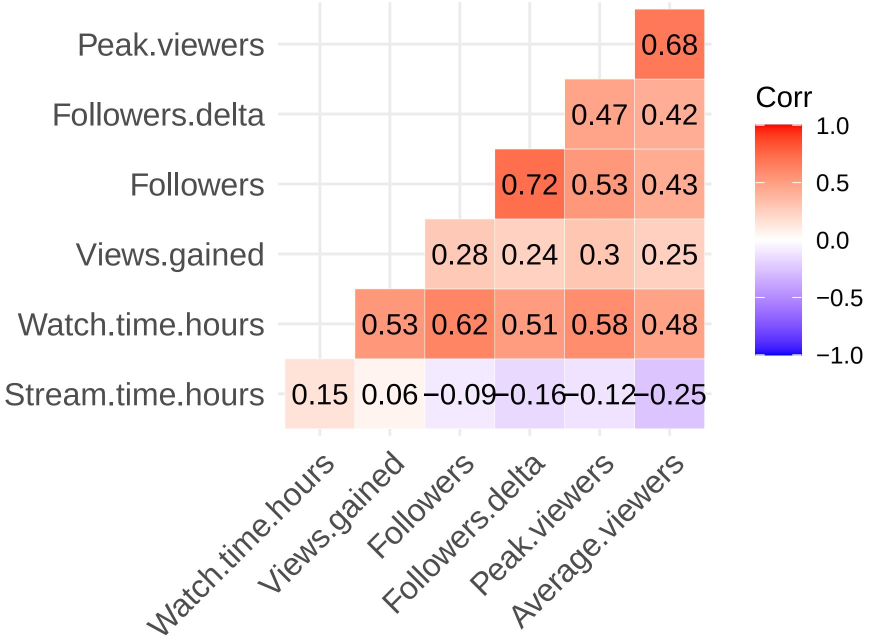
```
languages <- twitch_data %>%
  group_by(Language) %>%
  summarize(Count = n())
ggplot(languages, aes(x = reorder(Language, -Count), y = Count)) +
  geom_bar(stat = "identity") +
  labs(x = "Language", y = "Count") +
  coord_flip()
```



Since, nearly 50% of our data is of English streamers, this also means that this is quite unbalanced. For this study, it is also not feasible to recommend for someone to stream some language which we do not know they can speak, as such language will be mostly omitted for this report.

Finally we can also check for correlation via a matrix:

```
ggcorrplot(cor(numeric), lab = TRUE, type = "lower", outline.col = "white",
           hc.order = TRUE)
```



Looking at the results, we can notice a stronger relationships between `Followers` and `Followers.delta` as well as, between `Watch.time.hours` and `Followers`. This suggests that increasing watch time results in an increase in followers. We must keep in mind however, correlation does not imply causation. Even though it may make sense logically, we must still evaluate further.

5. Statistical Analysis

In this section, we will employ generalized linear regressions to delve into the data further and attempt to create models that will facilitate informed decision making. For each question we will try to come up with one model that can let us analyze relationships in the best way.

To address the first, we wish to evaluate whether streaming mature content for an audience that is 18+ can have consequences on the audience reaction. We need to find a way to quantify 'audience reaction'. This is not a trivial task as would involve trying to set a value to peoples feelings. Moreover, if we look at the columns in our data set, we do not have any metric or feature that can describe feelings or reaction.

Instead we have a arrangement of channel related metrics. Namely we have figures such as `Average.viewers`, `Views.gained`, `Peak.viewers` or `Followers.delta`. Of these, I find that `Average.viewers` can describe audience reaction. This is because it represents a quantitative measurement of the number of people who watched a channel. Moreover, it being a mean, it represents the viewership numbers over multiple streams. Although, it would be even more valuable if we had time series data, such that we can observe effects over periods of time. It must also be reiterated that no single metric is entirely fault-proof in capturing the entire situation.

With that, we can create our first model. To explore the relationship between `Average.viewers` (response variable) and the other predictors.

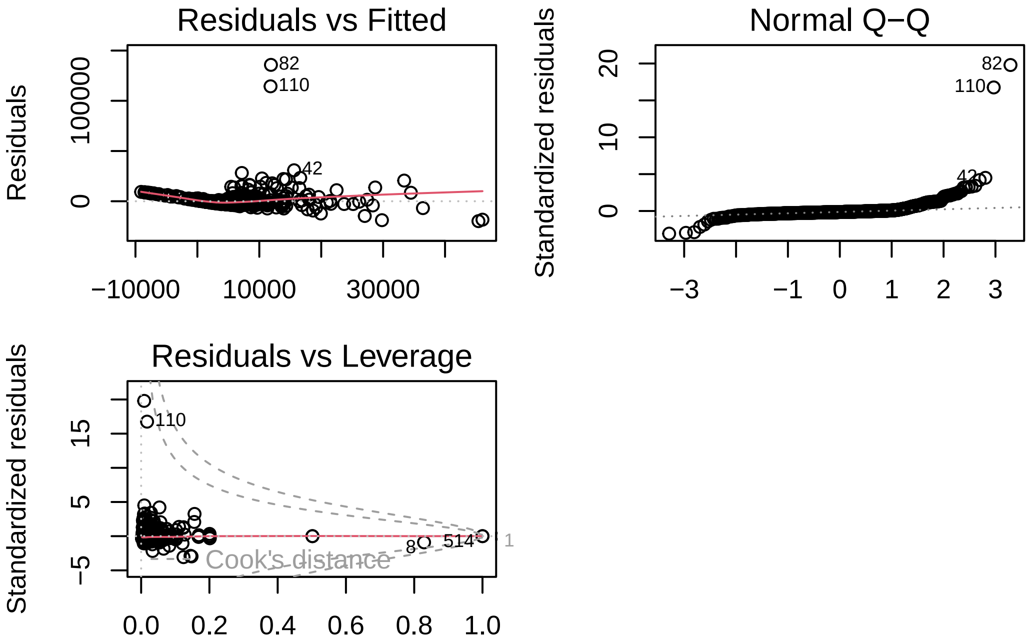
```r
average_viewers <- lm(Average.viewers ~ Mature + Stream.time.hours +
                      Language + Watch.time.hours + Followers + Followers.delta +
                      Views.gained + Partnered, data = twitch_data)

summary(average_viewers)
```

Looking at the summary (Appendix 1) results of the model, we can see that: `MatureTRUE` is negative but not statistically significant (p value > 0.05), suggesting that there is no strong evidence that mature content has a significant effect on viewership even when controlling for other factors. We also do see that, `Stream.time.hours`, `Watch.time.hours` and `Followers.delta` are all highly significant predictors, indicating they have a substantial influence on the average viewership metric. Interestingly `Stream.time.hours` is also negative, this suggests that streaming more does not increase average viewership. This agrees with the trend noticed during EDA.

As such, we will analyze plots for the given model and evaluate.

```r
par(mfrow = c(2,2), mar = c(3, 4, 1.4, 1))
plot(average_viewers, which = c(1,2,5))
```

```
Warning: not plotting observations with leverage one:
  378, 384, 923


Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced


Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



We first look at the `Residuals vs Fitted` plot, Used to check the linear relationship assumptions. A horizontal line, without distinct patterns is an indication for a linear relationship. `[1]` From the plot above, there is no pattern in the residual plot. This suggests that we can assume linear relationship between the predictors and the outcome variables. That does fit the adhere to the first of the assumptions of Linear regressions.

The `Normal-QQ` plot of residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow the dotted line. `[1]` From this we can clearly see that this plot does not do so, hence we reinforce the point that our data is not normally distributed and may benefit from a log scaling.

The final plot we will examine is the `Residual vs Leverage` plot, this plot helps us identify outliers and points with high leverage, which are data points that have large influence on the model. `[1]`

The plot above highlights the 3 most extreme points, however none of them appear outside of the limits of the cooks distance. However, there are 3 points namely: 378, 552, 923 that are of leverage 1. That mean they are very influential for the model.

Given that we do our residuals are not normal, we will try to log scale of the numerical variables to attempt to make our data more normal. It must be mentioned that there is a cost to such an operation, since one of our columns namely `Followers.delta` contains negative values, these cannot be log transformed, therefore are removed (replaced with NA's) as a result.

`Note, i dont do this analysis again because of how much space it takes.`

```r
log_twitch_data <- twitch_data %>%
  mutate(across(colnames(numeric), ~log(.)))
```

Given that, lets run the same model but now using the log scaled data on the same model that we defined before.

```r
average_viewers_log <- lm(Average.viewers ~ Mature + Stream.time.hours +
               Language +  Watch.time.hours + Followers + Followers.delta +
               Views.gained + Partnered, data = log_twitch_data)
summary(average_viewers_log)
```

According to the summary (Appendix 2) It appears that log transformation have improved the model by resulting in additional `significant` relationships between the response and predictor variables, that were statistically insignificant before the transformation. However, our goal of identifying a the interaction between viewer reception (average viewers) and content Maturity is still of a p-value of `0.12158` which is still not statistically significant.

On the other hand, the log transformation of the residuals did offer other insights. Namely, our R^2 has increased significantly, from `0.3361` to `0.9854`. Also, that there is a positive linear relationship between `Watch.time.hours` and `Followers` with `Average.viewers`. This is sort of natural reasoning, that if you have more viewers on average then people have watched you for more hours as well as you have more followers. Another outcome seen, is that certain languages are also more significant. As mentioned in EDA however, we will not focus on language in this study as we find its not possible to recommend someone to stream in another language entirely.

Given that fact, we now will attempt to model the relationship between production of mature content and effect that has on Twitch Partnership status.

For this we can make use of a logistic regression. Since the dependent variable that we are trying to predict is being a Twitch partner or not, which is binary. Using a logistic regression also allows us to handle both continuous and categorical variables such that we can also control for all of the other independent variables.

```r
partnered_mature <- glm(Partnered ~ Mature + Watch.time.hours +
                  Stream.time.hours + Peak.viewers +
                  Followers + Followers.delta + Views.gained +
                  Average.viewers + Language,
                  data = twitch_data, family = binomial())
summary(partnered_mature)
```

Looking at the summary from this model (Appendix 3), we can find that there is no significance between maturity and partnership status. This is expected as we found that there are only `22` non partnered streamers in this data set. As mentioned, this means that our minority class is very small. As a consequence of that, there is no power/significance to predict.

Finally, we want to address the final point to investigate. Namely, whether the effect of stream time is larger or smaller for mature content creators.

To do this, we again can make use of a linear regression. We include an interaction term between `Stream.time.hours` and `Mature`. The interaction term allows the effect of stream time on the average viewership to vary depending on whether the content is mature or not. We again, like in the previous models control for the rest of our independent variables.

In this context, we might suspect that there is an effect of streaming time on the success. Specifically, it might differ depending on whether or not the streamer streams mature content. A potential example reasoning could be; longer stream times might be more beneficial for streamers who focus on mature content. As they could be able to stream at non-standard hours.

By including the interaction term `Stream.time.hours*Mature` in the model, we can test this hypothesis directly.

```
streamMatureModel <- lm(Average.viewers ~ Stream.time.hours * Mature +
                          Watch.time.hours + Peak.viewers + Followers +
            Followers.delta + Views.gained + Partnered + Language,
            data = twitch_data)
summary(streamMatureModel)
```

From the summary (Appendix 4) we can notice that the coefficient of the interaction term between `Stream.time.hours` and `Mature` is -0.1402. This negative coefficient suggests that the effect of stream time on Average.viewers is slightly smaller in mature content compared to non-mature content. Again, sadly the interaction is not statistically significant as the p value is larger than 0.05. There were other points of significance however, The intercept term suggests a baseline average viewership of around 4,882. Then, again the negative coefficient for `Stream.time.hours` indicates that an increase in streaming hours is associated with a decrease in `Average.viewers`. This is something present also in the other models. Conversely, higher values for variables like `Watch.time.hours`, `Peak.viewers`, and `Followers.delta` have positive coefficients, suggesting that more watch time hours, a larger number of peak viewers, and an increase in follower count will correspond to higher average viewership on Twitch. These are quite natural conclusions.

6. Recommendation & Discussion

Recommendation

The aforementioned models have been created and analysed to then give recommendations to the given questions.

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?

As it was mentioned in the analysis of the model, there is no significant relationship between average viewership and maturity rating. As such, the maturity rating does not seem to have an effect on the reaction of audience. As such the recommendation is that you can pick whether you want to stream to a mature audience. As it should not have any consequences.

2. Does such a focus on mature content lower or increase the chance of becoming a Twitch partner?

For this, again there is nothing that suggests that there is a relationship between steaming mature content and becoming a Twitch partner.

3. Is the effect of the stream time larger or smaller in mature content?

Based on the analysis of the model created for this, the suggestion is that streaming more hours actually has a negative effect on average viewership regardless of the content type you stream. As such, you should not be streaming for too many hours, as that can lower your average viewership.

Discussion

Data limitations

As mentioned at many instances during this report, the data was limited to the top 1000 streamers on twitch. This intrinsically means that whatever insight is extracted throughout this document may not extrapolate the same for the rest of the population. This is often a problem in using statistics, and it is called sampling biased, as we purposefully select the top 1000 streamers, a trend that we could observe here does not necessarily have to be present globally in the population. An example of this could be the analysis of two of the models that suggested that `Stream.time.hours` has a significant negative coefficient for `Average.viewers`. This seems to suggest that to have more viewers on average, you should stream for less hours. This can be the case for this small subset of streamers, however one can imagine that in a highly competitive market such as streaming, not streaming, especially when you are not known yet is not a very good approach for growth. A possible reasoning for this trend could be that large streamers already have established fan bases, such that you will have viewers regardless.

Another problem that arises is that nearly everyone is partnered as they are a big streamer. As mentioned within the data set there are only 22 non-partnered streamers (regardless of their maturity status). That is 2.2%, this is definitely not a representation of the population. The reason why we have such high representation of Twitch partners in the top 1000, can be due to the benefits that being partner brings to a channel: It allows for monetization options [3]. Most top streamers as a consequence have made streaming their full time jobs. This is not the situation for a lot of streamers in the total population.

Moreover, the data was fetched in 2020, which in the age of the internet is a long time ago. As such, any recommendation made now could potentially have applied in 2020, but this doesn't mean it can reflect what would work now. A potential fix to this issue would be the addition of time series or even potentially panel data such that you can observe results over multiple years. This has the added benefit to potentially observe how a channel was performing before partnership and after.

Yet another limitation of the data is the available features. They are all success factors that do not allow us to differentiate from one another. For most of the features, the streamer is not directly able to affect them, such as watch time. This is more a consequence of their success. To aid this, new data that gives context on their streaming preferences, such as their stream time per category. Other metrics that could be useful are for example the tags they have in their streams, this would allow us to differentiate more between streamers as Twitch does not only have gaming categories, but also Just Chatting as well as a multitude of others. [3]

Suggestion for improvement

To aid this study such that more insight can extracted would be to include more data.

Most importantly, to include data from the entire population, not just the hyper successful. This is integral for such a study as insight gained based on the hyper successful does not necessary carry over to the rest of the population, who is also trying to succeed in this hypercompetetive field. Moreover, if we could include some features that describe the type of content the streamer makes. This data is readily available on TwitchTracker [2] and could be accessible through web-scraping or API. Lastly, if we could also scrape more up to date data and or over multiple years (such as adding time series or panel data).

7. Conclusion

In conclusion, it is possible to give a recommendation that answer the given question, however as seen through the length of this report, many problems with the validity of such a recommendation are highlighted. These problems were reiterated and explained extensively within the discussion section, with many possible suggestions to improve this study such that the outcomes can be more reflective of the entire streaming landscape rather than the hyper successful subset.

8. Bibliography

[1]: Kassambara, soyan, R., Vividdiagnostics, Eva, Visitor,Mann, T. (2018, March 11). Linear regression assumptions and diagnostics in R: Essentials. STHDA. http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials

[2]: Twitch channels, games and Global Statistics. (n.d.). Retrieved from https://twitchtracker.com/

[3]: Twitch. (n.d.). https://www.twitch.tv/

9. Appendix

1 average_viewers:

```
summary(average_viewers)
```

```
Call:
lm(formula = Average.viewers ~ Mature + Stream.time.hours + Language +
    Watch.time.hours + Followers + Followers.delta + Views.gained +
    Partnered, data = twitch_data)

Residuals:
   Min     1Q Median     3Q    Max
-19785  -1699   -949    -33 135761

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        6.266e+03  1.604e+03   3.908 9.96e-05 ***
MatureTRUE        -6.937e+02  5.393e+02  -1.286  0.19867
Stream.time.hours -1.704e+00  1.662e-01 -10.253  < 2e-16 ***
LanguageArabic    -1.194e+03  3.118e+03  -0.383  0.70182
LanguageChinese    8.201e+02  1.306e+03   0.628  0.53015
LanguageCzech     -8.416e+02  2.835e+03  -0.297  0.76661
```

```
LanguageFinnish     1.330e+03  6.911e+03   0.192  0.84748    
LanguageFrench     -6.121e+02  9.095e+02  -0.673  0.50113    
LanguageGerman     -3.469e+02  1.036e+03  -0.335  0.73777    
LanguageGreek      -9.741e+02  6.900e+03  -0.141  0.88776    
LanguageHungarian  -1.049e+03  4.883e+03  -0.215  0.83000    
LanguageItalian    -1.421e+03  1.712e+03  -0.830  0.40663    
LanguageJapanese    1.110e+03  2.215e+03   0.501  0.61656    
LanguageKorean     -1.745e+02  8.709e+02  -0.200  0.84127    
LanguageOther      -3.746e+02  7.082e+03  -0.053  0.95783    
LanguagePolish     -1.010e+03  2.018e+03  -0.500  0.61706    
LanguagePortuguese -2.852e+02  9.458e+02  -0.301  0.76310    
LanguageRussian     1.362e+03  8.873e+02   1.535  0.12519    
LanguageSlovak     -6.660e+02  6.899e+03  -0.097  0.92312    
LanguageSpanish    -8.345e+02  9.678e+02  -0.862  0.38871    
LanguageSwedish    -3.544e+02  6.907e+03  -0.051  0.95909    
LanguageThai        2.905e+02  2.108e+03   0.138  0.89040    
LanguageTurkish    -8.165e+02  1.511e+03  -0.540  0.58907    
Watch.time.hours    3.902e-04  3.657e-05  10.670  < 2e-16 ***
Followers           5.390e-04  4.457e-04   1.209  0.22683    
Followers.delta     3.267e-03  1.014e-03   3.221  0.00132 ** 
Views.gained       -3.156e-06  1.047e-05  -0.301  0.76317    
PartneredTRUE      -1.526e+03  1.540e+03  -0.991  0.32194    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6888 on 972 degrees of freedom
Multiple R-squared:  0.354, Adjusted R-squared:  0.3361 
F-statistic: 19.73 on 27 and 972 DF,  p-value: < 2.2e-16
```

2 average_viewers_log:

```
summary(average_viewers_log)
```

```

Call:
lm(formula = Average.viewers ~ Mature + Stream.time.hours + Language + 
    Watch.time.hours + Followers + Followers.delta + Views.gained + 
    Partnered, data = log_twitch_data)

Residuals:
     Min       1Q   Median       3Q      Max 
```

```
-0.93952 -0.02366  0.01652  0.04802  0.54452

Coefficients:
                    Estimate Std. Error  t value Pr(>|t|)    
(Intercept)        -0.216887   0.084194   -2.576  0.01014 *  
MatureTRUE          0.013644   0.008805    1.550  0.12158    
Stream.time.hours  -0.965407   0.006070 -159.055  < 2e-16 ***
LanguageArabic      0.139944   0.050818    2.754  0.00600 ** 
LanguageChinese     0.044208   0.021905    2.018  0.04385 *  
LanguageCzech      -0.075730   0.045954   -1.648  0.09969 .  
LanguageFinnish     0.022201   0.111693    0.199  0.84248    
LanguageFrench      0.028315   0.014757    1.919  0.05532 .  
LanguageGerman     -0.002521   0.016686   -0.151  0.87993    
LanguageGreek       0.128710   0.111480    1.155  0.24856    
LanguageHungarian   0.009019   0.078848    0.114  0.90896    
LanguageItalian     0.014900   0.027855    0.535  0.59284    
LanguageJapanese    0.026981   0.036144    0.746  0.45556    
LanguageKorean      0.055215   0.014368    3.843  0.00013 ***
LanguageOther      -0.056919   0.115319   -0.494  0.62172    
LanguagePolish     -0.009697   0.032632   -0.297  0.76640    
LanguagePortuguese  0.036731   0.015516    2.367  0.01811 *  
LanguageRussian     0.026740   0.014877    1.797  0.07257 .  
LanguageSlovak      0.066004   0.111328    0.593  0.55340    
LanguageSpanish     0.025263   0.015315    1.650  0.09936 .  
LanguageSwedish     0.050162   0.111654    0.449  0.65334    
LanguageThai        0.054147   0.034412    1.573  0.11593    
LanguageTurkish     0.032483   0.024795    1.310  0.19049    
Watch.time.hours    0.989235   0.008932  110.746  < 2e-16 ***
Followers           0.018233   0.005809    3.139  0.00175 ** 
Followers.delta    -0.009801   0.004505   -2.176  0.02982 *  
Views.gained       -0.002527   0.006822   -0.370  0.71115    
PartneredTRUE      -0.026124   0.025228   -1.036  0.30068    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1111 on 969 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.9858,    Adjusted R-squared:  0.9854 
F-statistic:  2493 on 27 and 969 DF,  p-value: < 2.2e-16
```

3 partnered_mature:

```
summary(partnered_mature)
```

```

Call:
glm(formula = Partnered ~ Mature + Watch.time.hours + Stream.time.hours +
    Peak.viewers + Followers + Followers.delta + Views.gained +
    Average.viewers + Language, family = binomial(), data = twitch_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4013   0.0832   0.1764   0.2204   0.9238

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         3.479e+00  6.458e-01   5.387 7.15e-08 ***
MatureTRUE         -1.499e-01  5.539e-01  -0.271  0.78667
Watch.time.hours    1.337e-07  9.249e-08   1.446  0.14823
Stream.time.hours  -1.309e-04  1.662e-04  -0.788  0.43099
Peak.viewers        1.942e-05  1.579e-05   1.230  0.21865
Followers           7.065e-07  1.009e-06   0.701  0.48360
Followers.delta    -1.593e-06  1.507e-06  -1.057  0.29042
Views.gained       -1.194e-08  7.699e-09  -1.551  0.12094
Average.viewers    -8.954e-05  5.449e-05  -1.643  0.10036
LanguageArabic      1.542e+01  4.709e+03   0.003  0.99739
LanguageChinese     1.583e+01  1.948e+03   0.008  0.99352
LanguageCzech       1.594e+01  4.354e+03   0.004  0.99708
LanguageFinnish     1.611e+01  1.075e+04   0.001  0.99880
LanguageFrench      3.141e-01  1.072e+00   0.293  0.76945
LanguageGerman     -3.931e-02  1.071e+00  -0.037  0.97071
LanguageGreek       1.597e+01  1.075e+04   0.001  0.99882
LanguageHungarian   1.561e+01  7.597e+03   0.002  0.99836
LanguageItalian     1.594e+01  2.584e+03   0.006  0.99508
LanguageJapanese    1.575e+01  3.308e+03   0.005  0.99620
LanguageKorean      5.158e-01  1.090e+00   0.473  0.63608
LanguageOther      -2.257e+01  1.075e+04  -0.002  0.99833
LanguagePolish      1.559e+01  3.063e+03   0.005  0.99594
LanguagePortuguese  3.158e-01  1.083e+00   0.292  0.77061
LanguageRussian    -1.591e+00  5.573e-01  -2.854  0.00431 **
LanguageSlovak      1.572e+01  1.075e+04   0.001  0.99883
```

```
LanguageSpanish     5.727e-01  1.212e+00   0.473  0.63653
LanguageSwedish     1.630e+01  1.075e+04   0.002  0.99879
LanguageThai        1.595e+01  3.203e+03   0.005  0.99603
LanguageTurkish     1.565e+01  2.252e+03   0.007  0.99445
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 211.45  on 999  degrees of freedom
Residual deviance: 177.62  on 971  degrees of freedom
AIC: 235.62

Number of Fisher Scoring iterations: 18
```

4 streamMatureModel:

```
summary(streamMatureModel)
```

```

Call:
lm(formula = Average.viewers ~ Stream.time.hours * Mature + Watch.time.hours +
    Peak.viewers + Followers + Followers.delta + Views.gained +
    Partnered + Language, data = twitch_data)

Residuals:
   Min     1Q Median     3Q    Max
-35484  -1117   -264    666 105098

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.882e+03  1.379e+03   3.540 0.000419 ***
Stream.time.hours            -1.182e+00  1.570e-01  -7.532 1.14e-13 ***
MatureTRUE                    2.750e+01  8.633e+02   0.032 0.974595
Watch.time.hours              1.577e-04  3.389e-05   4.653 3.72e-06 ***
Peak.viewers                  7.651e-02  4.101e-03  18.657  < 2e-16 ***
Followers                    -4.521e-04  3.875e-04  -1.167 0.243596
Followers.delta               2.340e-03  8.726e-04   2.682 0.007447 **
Views.gained                 -1.572e-06  8.997e-06  -0.175 0.861327
PartneredTRUE                -1.840e+03  1.329e+03  -1.384 0.166682
LanguageArabic               -2.213e+03  2.679e+03  -0.826 0.408962
```

```
LanguageChinese               1.632e+03  1.126e+03   1.450 0.147451
LanguageCzech                 9.007e+01  2.435e+03   0.037 0.970497
LanguageFinnish               1.334e+03  5.949e+03   0.224 0.822673
LanguageFrench               -7.070e+02  7.814e+02  -0.905 0.365813
LanguageGerman               -4.115e+01  8.896e+02  -0.046 0.963119
LanguageGreek                 1.863e+02  5.925e+03   0.031 0.974917
LanguageHungarian            -1.174e+03  4.195e+03  -0.280 0.779671
LanguageItalian              -6.915e+02  1.471e+03  -0.470 0.638320
LanguageJapanese              1.702e+03  1.903e+03   0.895 0.371243
LanguageKorean                7.703e+02  7.495e+02   1.028 0.304321
LanguageOther                -7.755e+02  6.086e+03  -0.127 0.898626
LanguagePolish               -1.162e+02  1.734e+03  -0.067 0.946569
LanguagePortuguese           -4.590e+02  8.192e+02  -0.560 0.575356
LanguageRussian               6.866e+02  7.628e+02   0.900 0.368272
LanguageSlovak                4.851e+02  5.924e+03   0.082 0.934752
LanguageSpanish              -1.193e+03  8.313e+02  -1.435 0.151714
LanguageSwedish               2.197e+02  5.931e+03   0.037 0.970465
LanguageThai                  4.076e+02  1.810e+03   0.225 0.821902
LanguageTurkish              -1.470e+02  1.298e+03  -0.113 0.909853
Stream.time.hours:MatureTRUE -1.402e-01  3.491e-01  -0.401 0.688184
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5915 on 970 degrees of freedom
Multiple R-squared:  0.5247,    Adjusted R-squared:  0.5105
F-statistic: 36.93 on 29 and 970 DF,  p-value: < 2.2e-16
```