EXPLORATORY DATA ANALYSIS

To get an overview of our data, we combine the necessary values into one single table. This helps us run a correlation heat map on all columns and find out hidden trends.

Table Column Types

Let us take a look at the columns we have on our table and their data types. As we can see, most columns have expected values. Certain columns have float64 as there are empty values.

Id	**int64**
AuthorUserId	**int64**
CurrentKernelVersionId	**float64**
ForkParentKernelVersionId	**float64**
ForumTopicId	**float64**
FirstKernelVersionId	**float64**
IsProjectLanguageTemplate	**bool**
CurrentUrlSlug	**object**
Medal	**float64**
TotalViews	**int64**
TotalComments	**int64**
TotalVotes	**int64**
LanguageName	**object**

Table 1 - Data Types

Data overview

Now, let us take a look at our data. First, we recognize the frame size of the entire table. This table has 201981 data samples with 13 rows.

From the below table, we can make certain inferences.

- The values for various **Id** are not relevant in this overview as they are merely unique identifiers.
- We can see that only 7347 kernels out of over 200000 have received medals.
- Over 50% of the kernels that have received medals have received only bronze and shows that gold medals are rare.
- We can infer the users of kaggle are not inclined at commenting on a kernel or voting as the discrepancy shows in the mean of total views, comments and votes.
- The large difference between standard deviation of total votes and its mean infers that only the top kernels get votes while a majority of the kernels don't receive any votes at all.

For further inferences, we need to look at the data even more carefully.

Id	Author User Id	Current Kernel Version Id	Fork Parent Kernel Version Id	Forum Topic Id	First Kernel Version Id	Is Project Language Template	Current Url Slug	Medal	Total Views	Total Comments	Total Votes	Language Name
1	2505	205	NaN	NaN	1	FALSE	hello	NaN	24	0	0	R
2	3716	1748	NaN	26670	2	FALSE	rf-proximity	3	7547	1	12	R
4	3716	41	NaN	NaN	9	FALSE	r-version	NaN	9	0	0	R
5	28963	19	NaN	NaN	13	FALSE	test1	NaN	9	0	0	R
6	3716	21	NaN	NaN	15	FALSE	are-icons-missing	NaN	7	0	0	R

Table 2 - Data Head

	Id	Author UserId	Current Kernel Version Id	Fork Parent Kernel Version Id	Forum Topic Id	First Kernel Version Id	Medal	Total Views	Total Comments	Total Votes
count	4847544	4847544	3974304	1528440	376752	4820784	176328	4847544	4847544	4847544
mean	9884131	19064268	43893672	32389800	1050411	36889224	63	7129	13	43
std	12270520	12980954	44682816	36301128	373736	42242592	14	68365	145	520
min	24	8832	456	24	346392	24	24	0	0	0
25%	2148000	9972912	11295427	7380816	723774	7956864	48	312	0	0
50%	4511544	16531224	25438920	15022056	1048920	21191004	72	552	0	0
75%	11661264	24045312	62394360	58945944	1378284	45348672	72	2040	0	0
max	47868504	58361016	166915896	166892136	1683840	166915896	72	10381752	21456	69192

Table 3 - Statistical Overview

Missing Data

As told earlier, some part of our data is missing. This is not an error in the dataset. Rather, the simply does not exist. For example, we may find the medals column to be empty because certain kernels may not have received a medal at all. To visualize this better, the following matrix helps us. This matrix shows us missing data from a randomly picked sample of 20000 data items.

- Most of medals is obviously empty as we inferred earlier. This is because most kernels on kaggle do not receive any medal out of the three tiers.
- **ForumTopicId** and **CurrentKernelVersionId** can have missing values as some kernels may not have a forum due to inactivity or lack of comments.
- **ForkParentKernelVersionId** has missing values as many kernels are started straight from scratch. This column only has values if the kernel is forked from another kernel on kaggle.

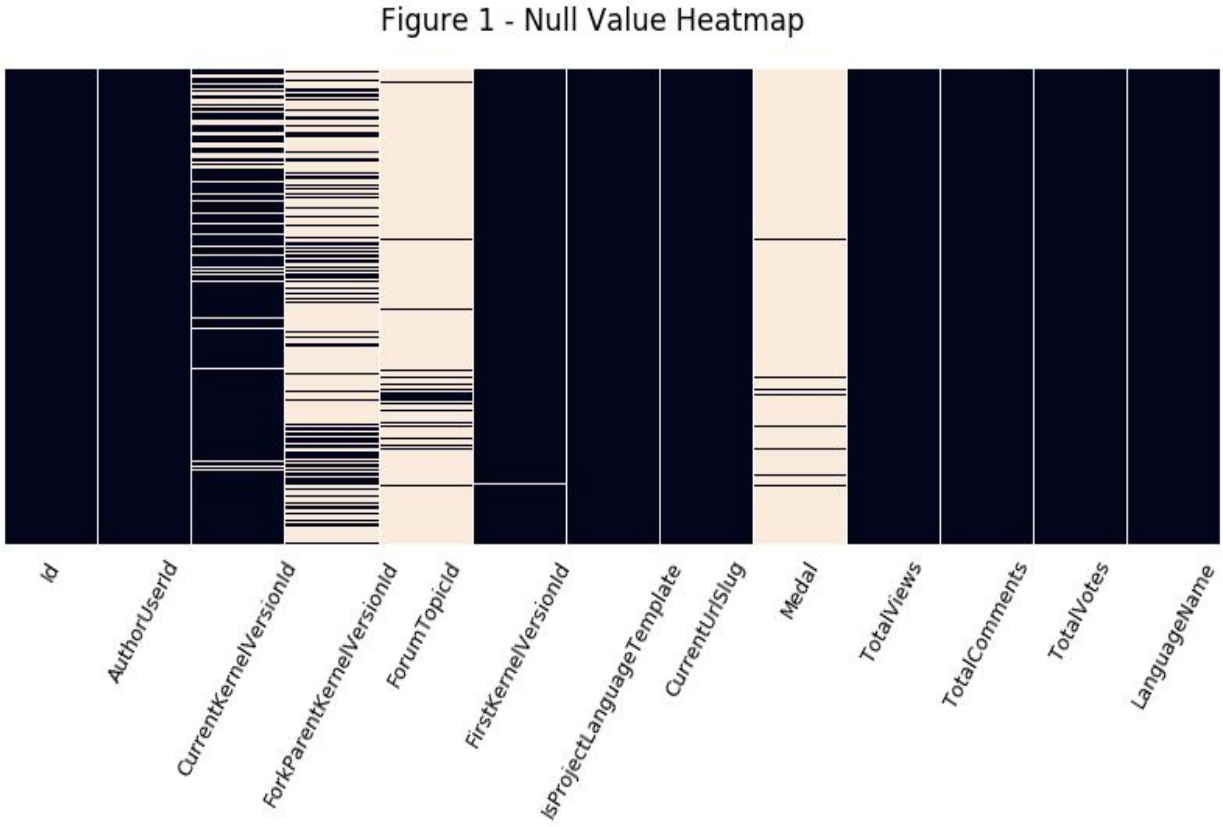

Figure 1 - Null Value Heatmap

Count Relationships

The below table shows the presence of a forum for kernels that don't have a medal. As we can see, over 95% of the kernels without a medal don't have a forum. This confirms our inference from the heatmap.

No Forum	185277
Forum Exists	9357

Table 4 - Forum Medal Relationship

The below table shows the number of kernels with respect to number of users. We can clearly see from this that over 50% of the users only have one kernel on Kaggle and over 90% of the users have less than 5 kernels submitted.

The graph paints a clearer picture. Due to extremities in the values and a highly skewed distribution, we have divided the table into two portions. The left half shows Users with a low kernel count while the right half shows users with a high kernel(>100). From the left half, it is clear that as the kernel count increases, the user count decreases. We can infer from this that fewer users have a higher number of kernels.

On the right half, we have the distribution of users who have over a 100 kernels. It is clear that a maximum of 4 users have any number of kernels over 100 and there is only one user for each kernel count above 200.

No Of Kernels	No Of Users
1	45999
2	14668
3	6758
4	3710
5	2502

Table 5 - Kernel User Count Relationship

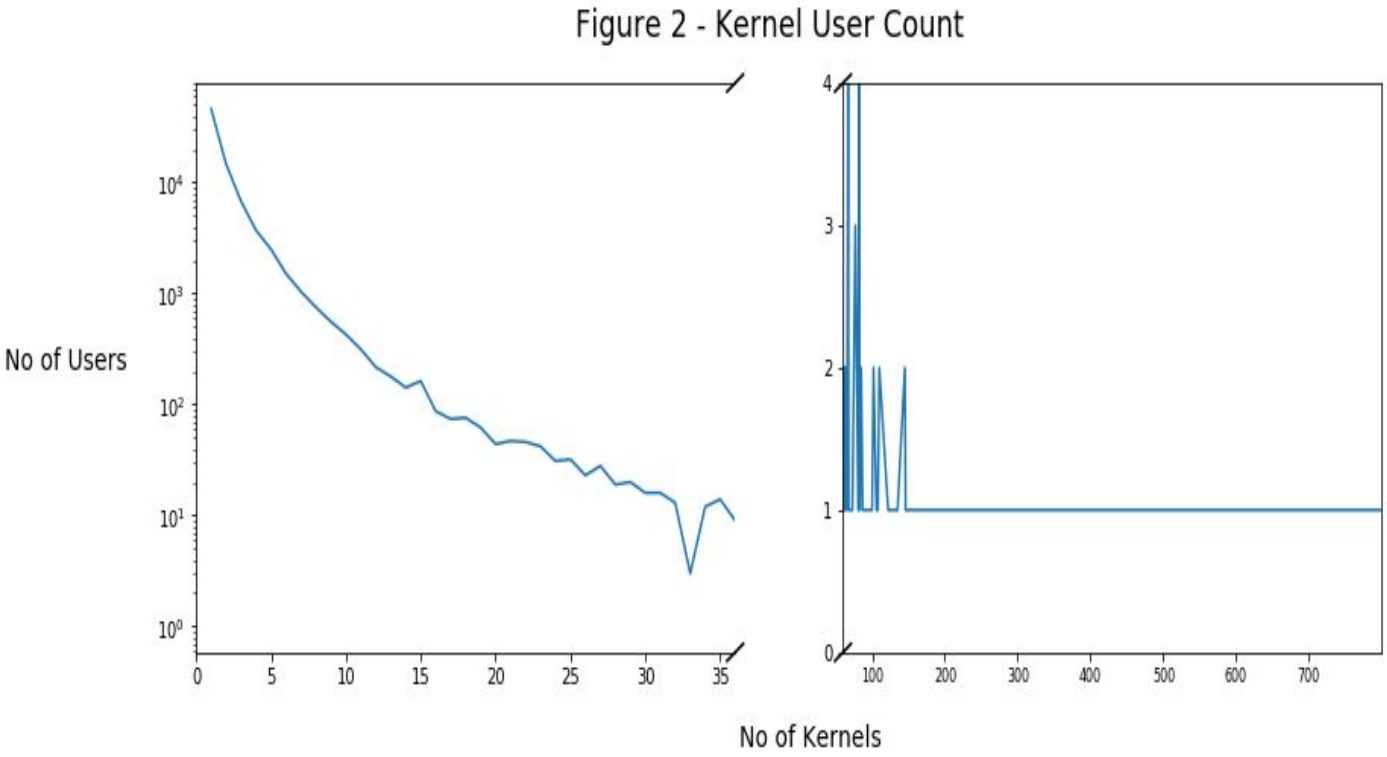

Figure 2 - Kernel User Count Chart

Language and Medal Trends

The below chart shows us the distribution of languages which kernels are submitted in. As we can see, an overwhelming majority of the users prefer using Python as their preferred language over R.

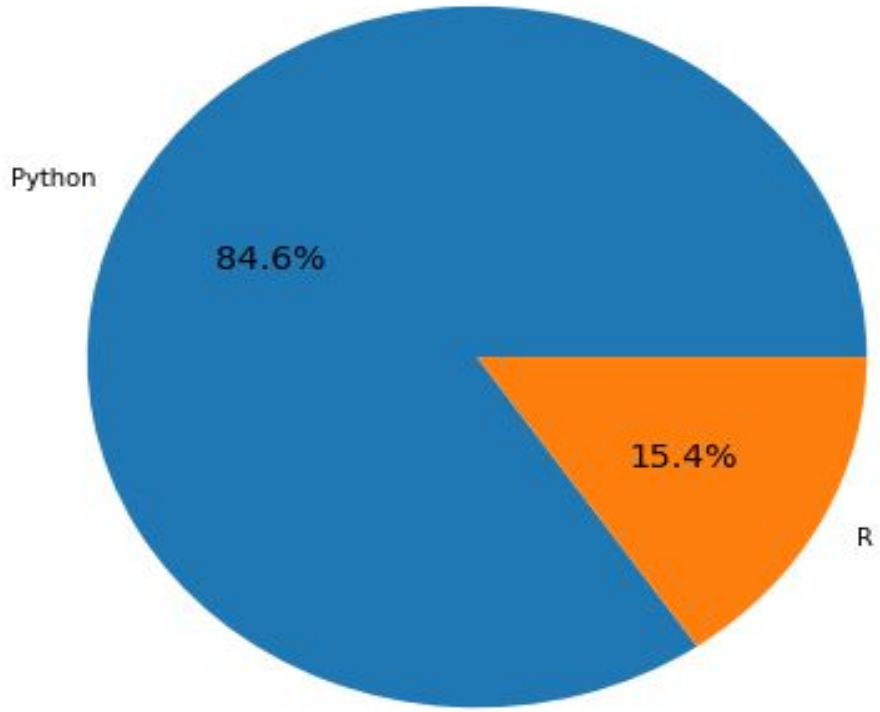

Figure 3 - Language Distribution

From our earlier inference in data overview, it was clear that gold was not given to many kernels.In the graph below, we can see the ratio. Nearly three quarters of the kernels that have received medals have received bronze. We should note however, the below pie chart is not representative of all kernels on Kaggle. Rather, this is the medal distribution among **kernels that have received medals only.**

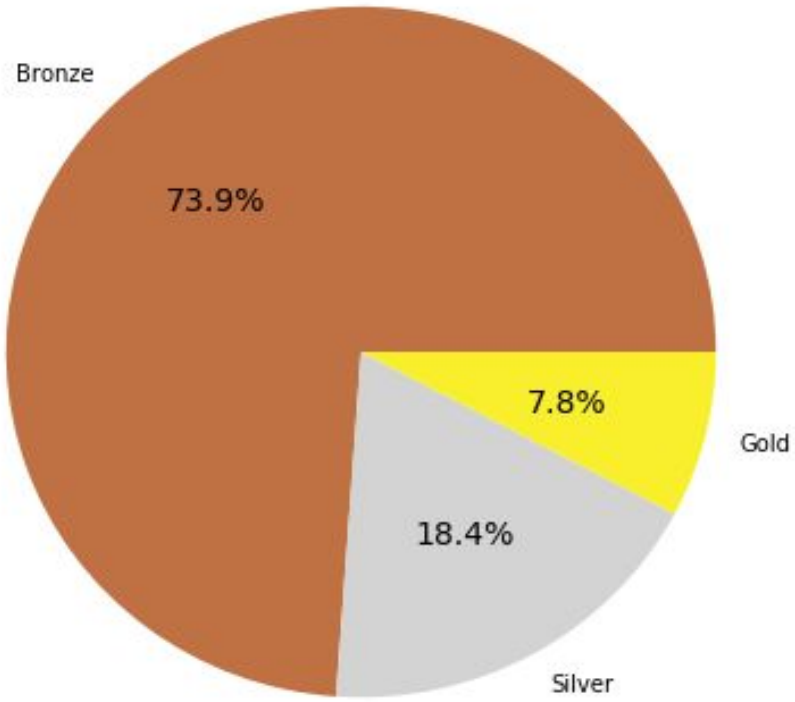

Figure 4 - Medal Distribution

The following chart is a visual representation of medal distribution spread across language. There are two important things to note here. This graph is shown as a representation of percentage of medals of each category given to the kernels of a language and not a sum of kernels. The data disproportion comes from a large difference of available kernels between R and Python

From this, we can see that the medal algorithm does not prefer one language over another. However, a higher percentage of R kernels receive Bronze medals whereas Python kernels receive a better portion of the Gold tier medals.

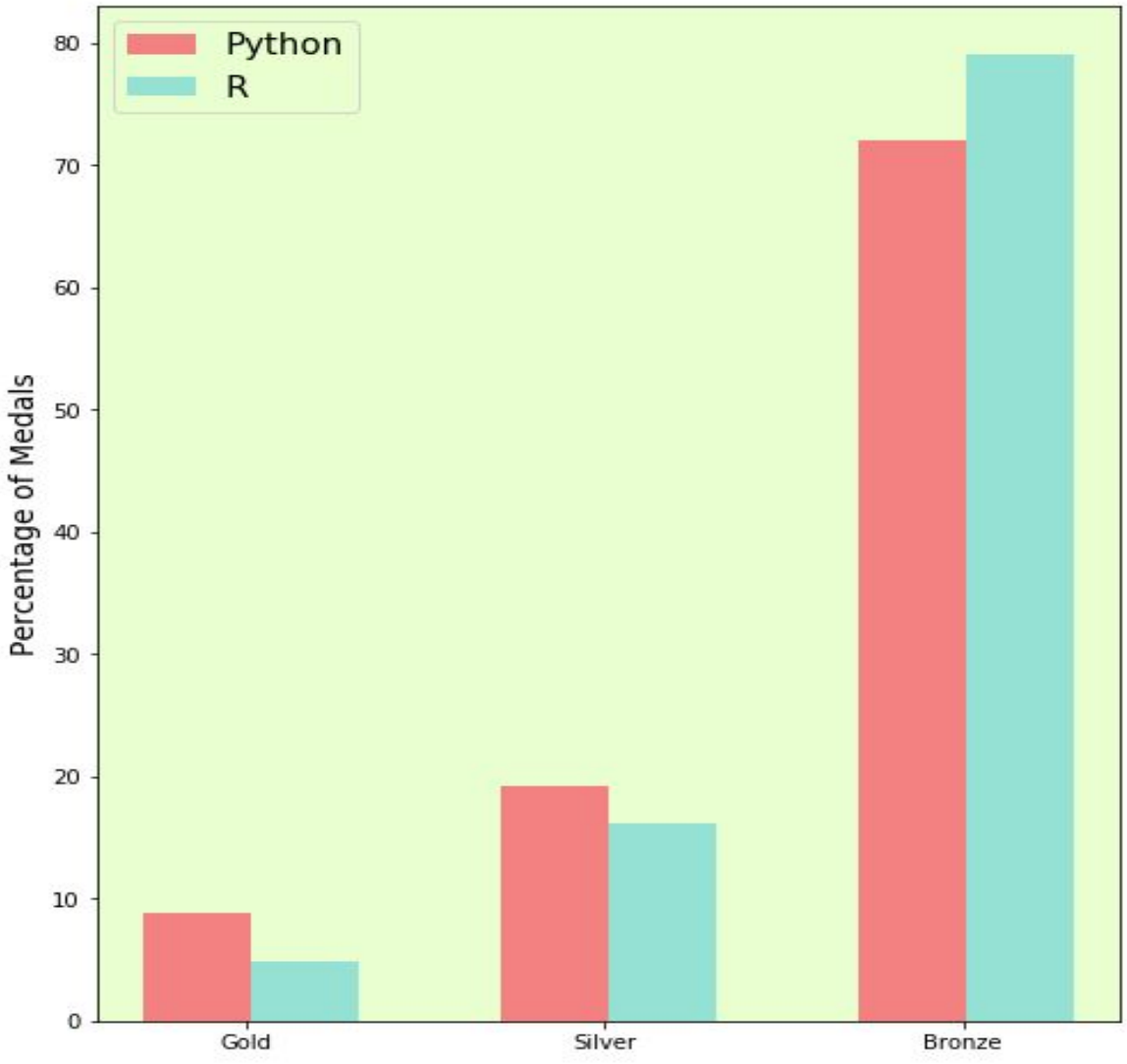

Figure 5 - Medal Distribution among Languages

Votes, Comments and Views

All Kaggle kernels have these values which combine to tell us the visibility it has which in turn tells how popular/good the kernel is. The dataset we have have has no missing values for these three columns.

First, we get an overall view of the three columns by visualizing it in the form of a venn diagram. The left diagram shows some key points

- A kernel does not have votes or comments without views. This is obvious as there are no intersections of views with other columns that are exclusive.
- A vast majority of the kernels simply do not have comments or votes. While 98% of the kernels have at least 1 view, only 18% of the kernels have votes and 7% have comments.
- If a kernel has a comment, it is likely to have a vote. This shows that users have a tendency to comment on kernels that only have votes.
- Nearly 2/3rd of the users who vote on a kernel prefer not to comment.

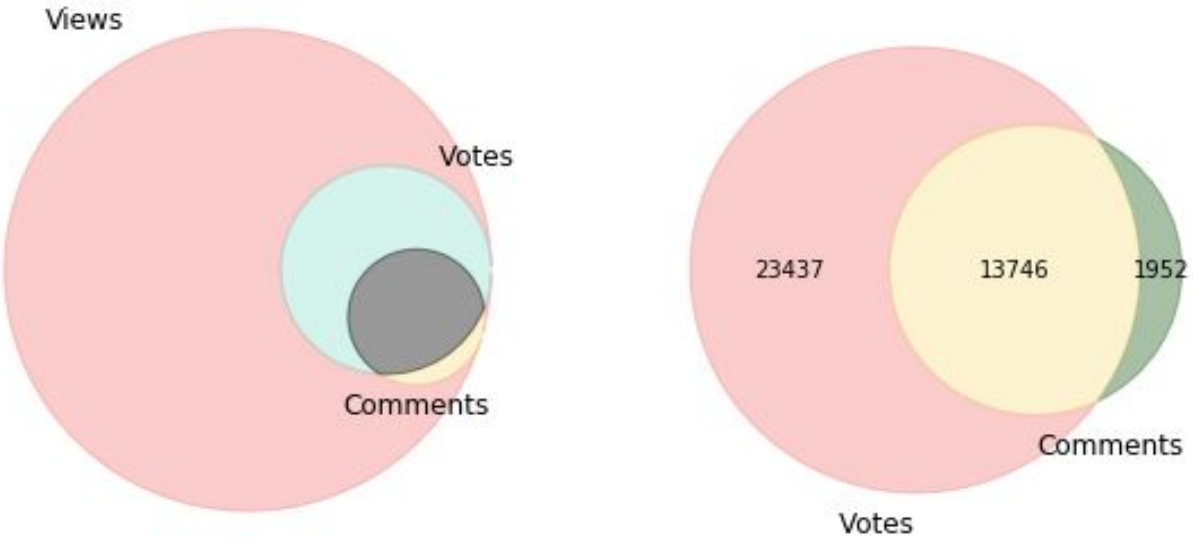

Figure 6 - Comment,View and Vote Tendency

Based on comments and views, we have devised a **popularity score**. In this score, **votes are given twice the weightage of a comment** while views are not included. The score has been normalized to a range of 100. As we can see from the distribution, much of the scores are below 5 indicating low popularity for most of the kernels.

The right hand side graph shows the popularity scores for kernels but **kernels with no medals have been removed**. The graph shape is similar but a lot of kernels with minute popularities have been eliminated.

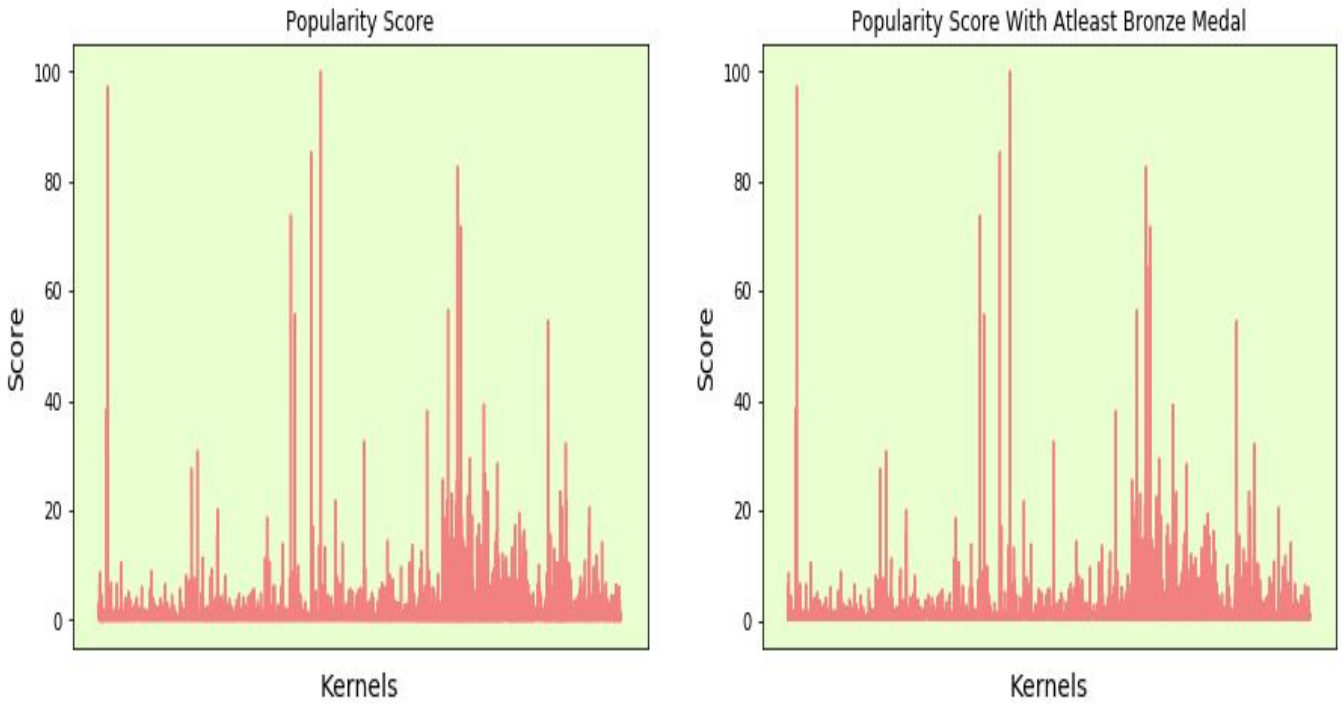

Figure 7 - Popularity Score Distribution

Now, we find and plot the popularity scores and see how it correlates to the medal system. We get the expected result from the graph. We can infer two things -

- The number of bronze medals awarded are higher compared to silver and silver is higher compared to bronze
- The scores of bronze medals are consistently low i.e. below 10.
- Silver medal kernels have a better score but not by much. Most scores are still below 20.
- Gold medal kernels are of significantly higher quality but with much more variance. Most scores are above 20

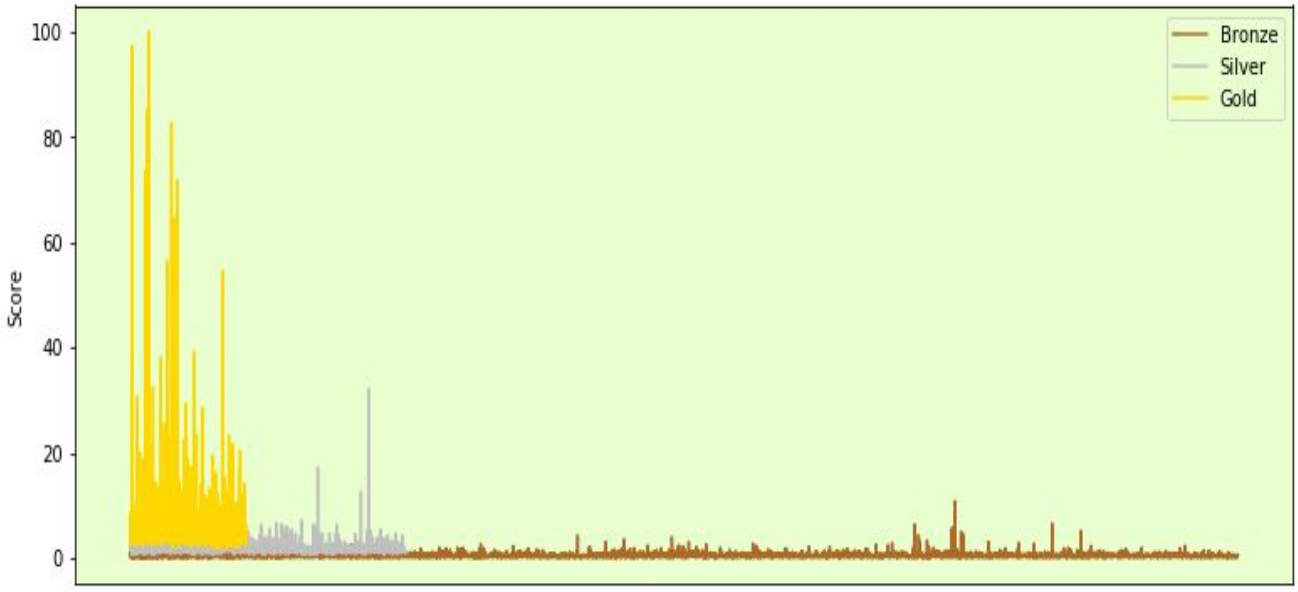

Figure 8 - Popularity Scores for each Medal

Correlation Heatmap

The correlation heatmap is a heatmap of a matrix of correlation of the rows. This heatmap can easily help us visualize any trends we have missed within the table and understand them.

From our heatmap, it is clear that there are no negative correlations at all. But as we can see, there is tightly knit relationship between total views, columns and comments.

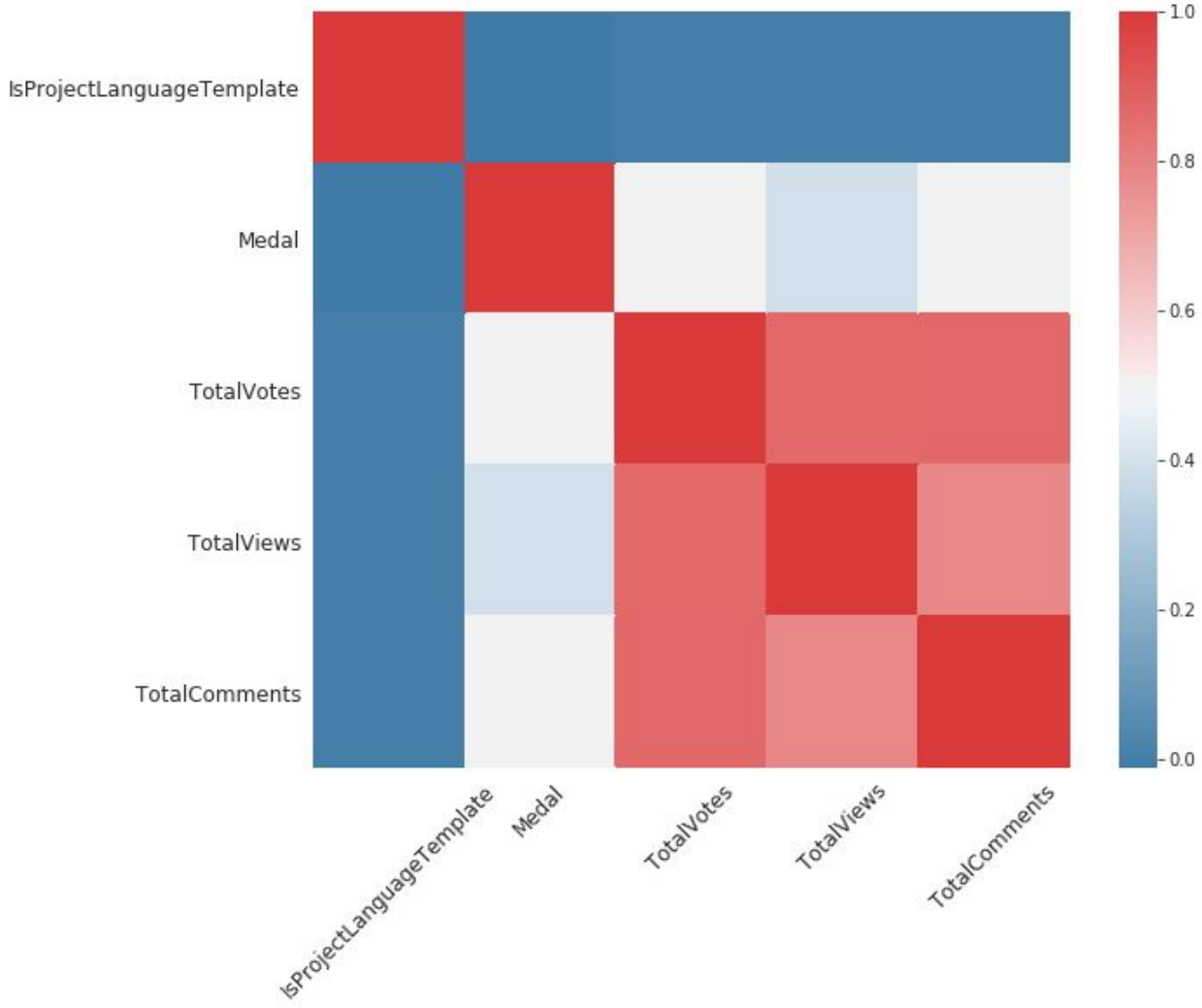

Figure 9 - Correlation Heatmap

Scatter plots based on Heatmap

Based on our heatmap, we must plot three scatter plots -

- Views - Votes(correlation of 0.867)
- Views - Comments(correlation of 0.783)
- Votes - Comments(correlation of 0.871)

In our Views - Votes scatterplot, we need some fine tuning. As far as the data is concerned, the correlation is correct. There is no divergence and apart from a few outliers at the 100000 mark, most kernels have a high vote count if they have a high view count. However, to make the data clearer, we filter the values upto 100000 views on the right. Now, we see a lot more outliers spread across the map. The inferences we can draw from is that for very popular kernels, high vote count is accompanied by high view count. **But even for kernels with over 100000 views, votes are very scarce**. **This shows tendency of users to not vote even for popular kernels.**

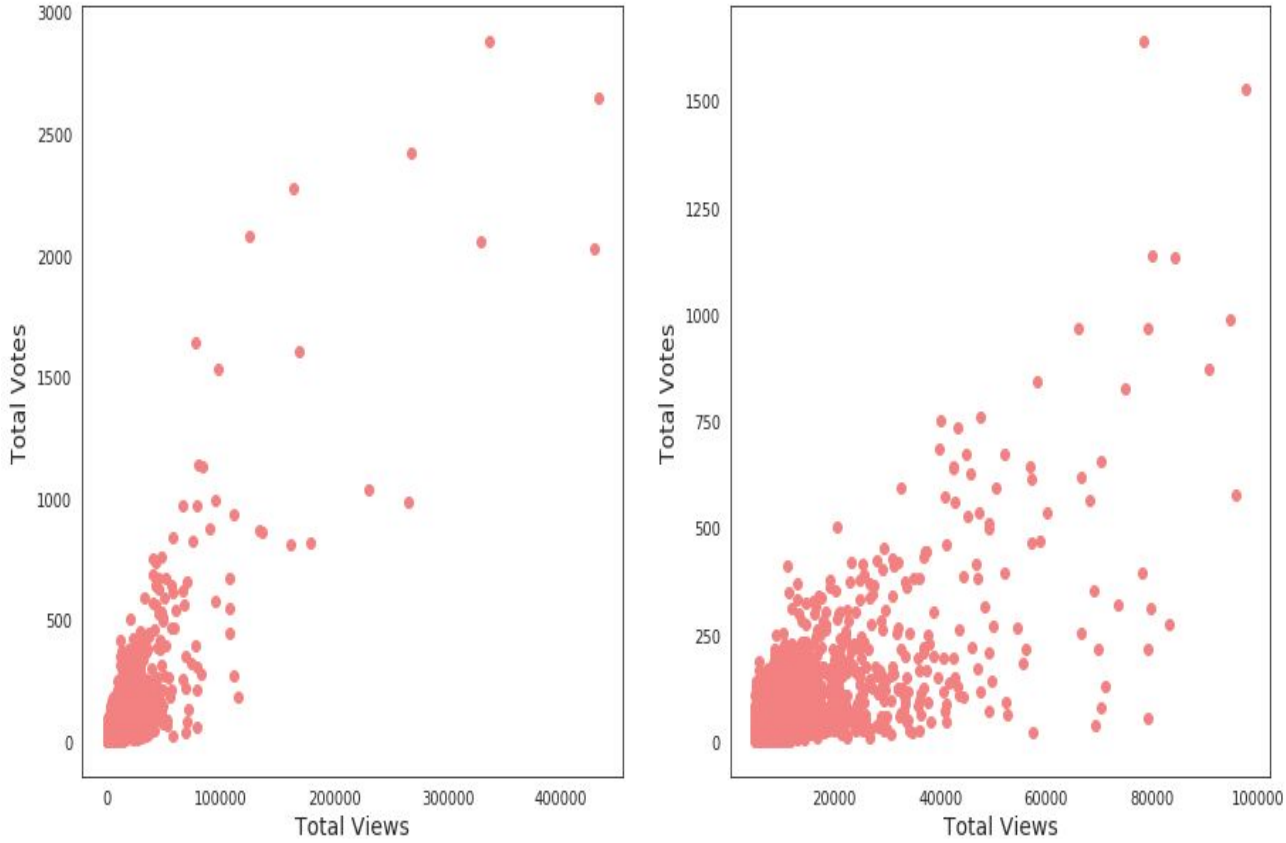

Figure 10 - Views Votes Scatter Plot

When we come to Views-Comments, it is a very similar story. Popular Kernels have many comments corresponding to views. But when we filter for kernels that have less than 100000 views, we can see it clearly. Many Kernels with a very high view count don't have comments.

Also, if we look at the scale, it is slightly lesser than votes. This shows that a kernel irrespective of view count, tends to have a higher vote count than comment count. Kaggle users may tend to vote less but they tend to comment even lesser.

From the above two trends, we can draw two possible conclusions -

- It could be due to the users that the results are the way they are. Users generally do not prefer to vote or comment after seeing a kernel. Sometimes, the kernel may be highly viewed. This maybe because the kernel got wrong popularity and is not as good as it seemed.
- Kaggle algorithm for views need correction. It may seem that kaggle is boosting kernel views for kernels with less views.

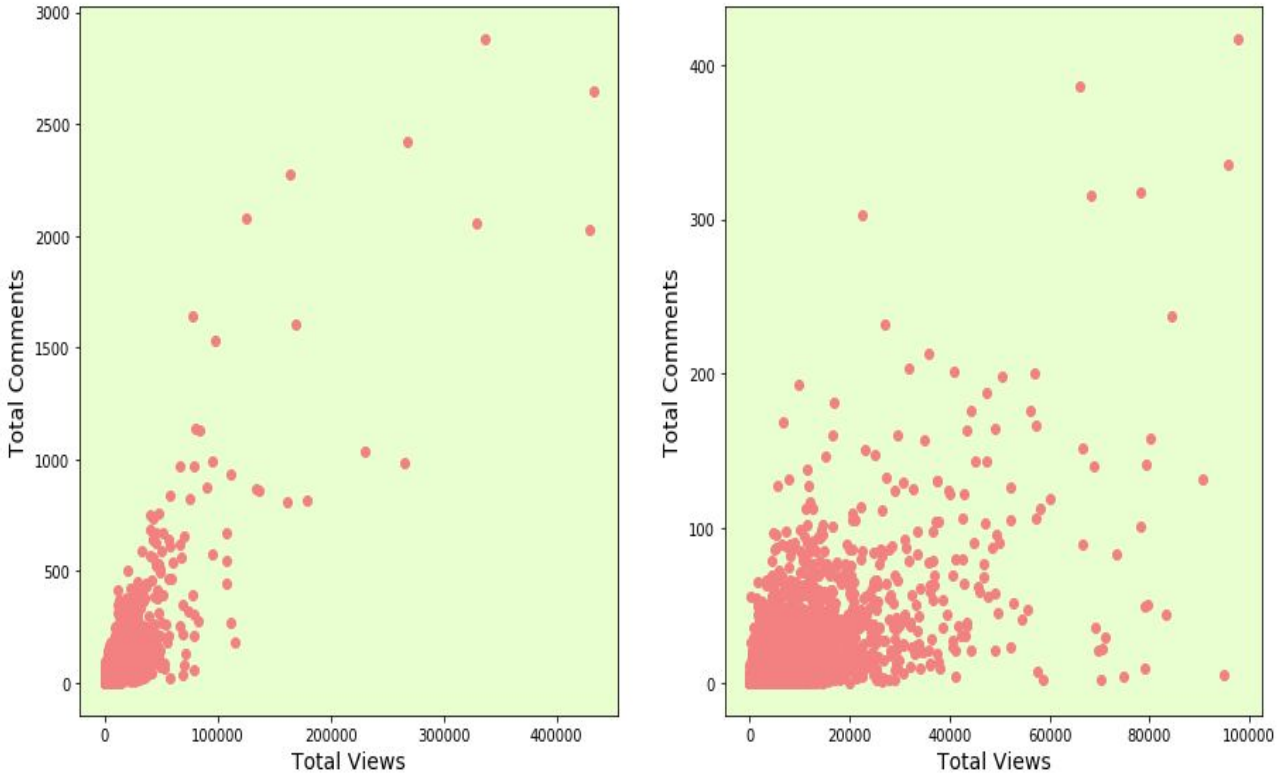

Figure 11 - Views Comments Scatter Plot

When it comes to votes and comments, the relationship is straight forward without many outliers. **A high vote count encourages users to comment more which highlights the tendency of kaggle users to prefer the popular.** However the graph data is skewed.

Filtering the votes less than 500 gives us a clearer image. Even though the correlation seems to be linear, the scatter tends to favor the x axis. This confirms our conclusion that comments are rarer than votes. I.e. a kernel will tend to have more votes than comments.
The extreme outliers on the left graph are those with very few comments but a lot of votes. **These may be 'diamond in the rough kernels' which users have noticed but not started a discussion on.**

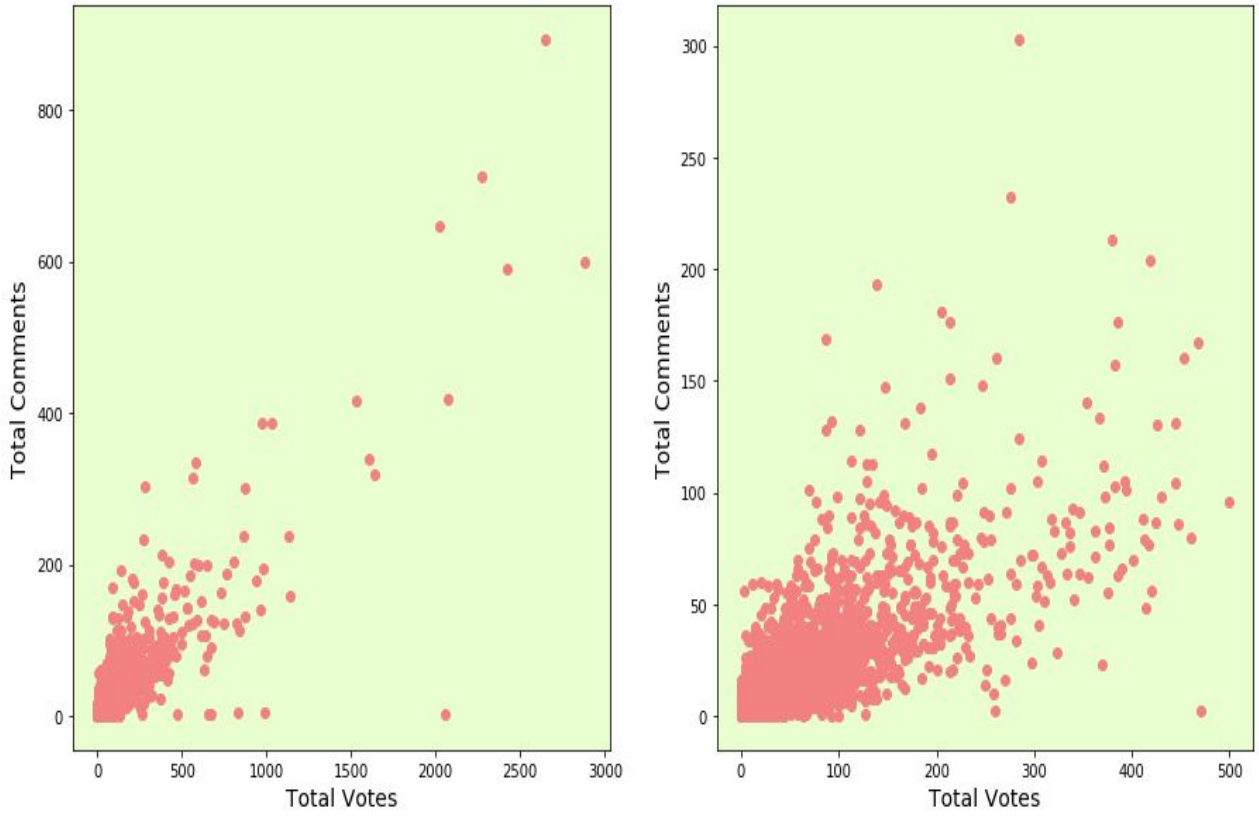

Figure 12 - Votes Comments Scatter Plot