Project Outcome Summary and Twitch Data Results

The aim of the project was to analyze the various sets of Twitch data, which was grouped by language region, and compare samples of the combined feature values with each other as produced by the implemented BFS, Dijkstra, and A* algorithms. In order to properly implement the algorithms to get suitable results, there were a few key design decisions that had to be made: making the unweighted graph weighted for Dijkstra, creating a heuristic function for A*, and determining how to demonstrate the success of the BFS traversal without causing the program to have an unnecessarily long runtime. The former two of these issues were solved by using the destination vertex's view count, as it intuitively made sense to determine the heuristic and edge weight upon a feature of the twitch account that was being followed. For the issue of having a BFS traversal that ran quickly enough, it was decided that only a sample of the data be printed to the terminal and be categorized by the means of assessment, which was a tabular histogram. Thus, the results that are assessed throughout this analysis are a sample of the actual data, and the commands provided in the ReadMe focus on an arbitrary connected component of a graph which is utilized to represent the tendencies of the remainder of the graph, as running the algorithms on full sets of data was too time-consuming.

Based on this data and output that is returned, the results need to be verified in a way to confirm that the algorithms work properly and did not just produce random values. This was solved in two ways: for all three algorithms, test cases were implemented on smaller and larger data sets (some of which were created specifically for testing purposes) to make sure the data was valid. Another method was by cross-referencing across algorithms so that the output for both Dijkstra and A* were the same, as both algorithms' greatest differentiator is the use of a heuristic function for A*. This combined with the test cases would confirm that both algorithms were implemented with consistency.

There were two specific issues that incentivized the group to test early and often. The first of these was with BFS, which initially confused the group when trying to implement. The issue that occurred was that the attempted implementation was similar to the approach in MP Traversals, which ended up being an overcomplicated process that outputted incorrect values. After fixing this by simplifying the process and making the BFS a part of the graph class itself, another issue that was raised that led to the development of certain test cases was the scenario for when a destination vertex passed into an algorithm

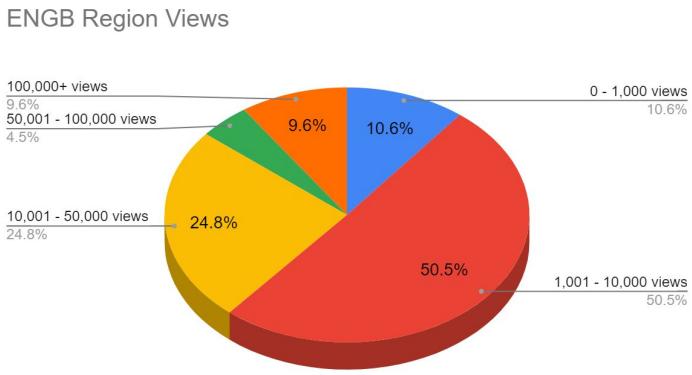
was not in the same connected component as the source vertex. To verify that the code worked properly, it was necessary to implement test cases to check this before advancing any further.

For reading the data and actual results themselves, there are different output values that are processed by the implementation of the algorithms. For BFS, the output provided by the code is in the form of a tabular histogram, as each of the sections categorizes users of a specific connected component based on one of the features (the number of views an account has). The Dijkstra and A* algorithms are used to calculate average views of an account and the average number of days an account has been active over the course of a shortest path that is found. Both of these algorithms should result in the same values for these results, thus confirming each other to be right.

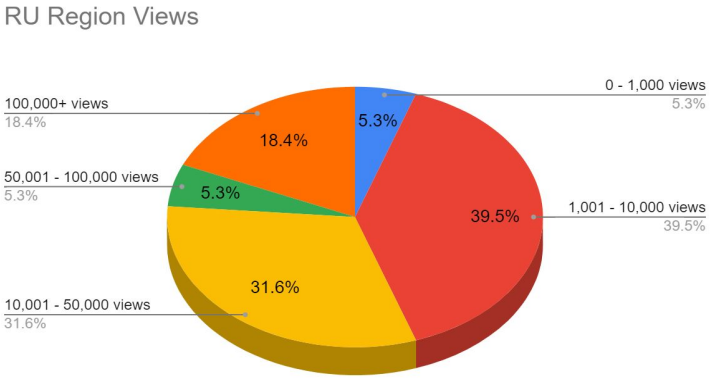
The findings within the data itself were quite fascinating, as the different output values proved a variety of things. A definite trend within the BFS pie chart results was that an overwhelming number of Twitch accounts across all regions were under 50,000 viewers, as shown in figures 1-6 in the appendix. This makes sense and proves that the provided data in the SNAP Twitch dataset was of a wide range of accounts, not only the ones ran by the most famous Twitch users across the different regions. However, the data across all the regions also showed that there tended to be more accounts with over 100,000 views than within the range of 50,000 and 100,000, which is interesting as this proves that there is not a linear correlation between a large number of views and how many accounts cross said threshold. As for the results of the Dijkstra and A* algorithms, which are summarized in figures 7-8 in the appendix, there were differences in the variation of data between the two figures. The traced paths between the different regions had a large variation in average view counts, as some were in the millions while others were short of 1,000. One thing to note is that this data may be skewed because for some paths, the users that follow each other may all be relatively close in terms of view counts (for example, multiple Twitch celebrities who interact). However, for figure 8, the average number of days these Twitch accounts existed seemed to be much more consistent across the different paths and regions. These numbers were all between 1,000 and 2,000, as perhaps this may have come from a certain point where Twitch surged in popularity. Overall, the numbers made sense, which provided a sanity check to the group, as reasonable data is complementary to testing to ensure the fact that the algorithms and traversal were properly implemented, and thus work.

# Appendix

## Figure 1




## Figure 2




## Figure 3

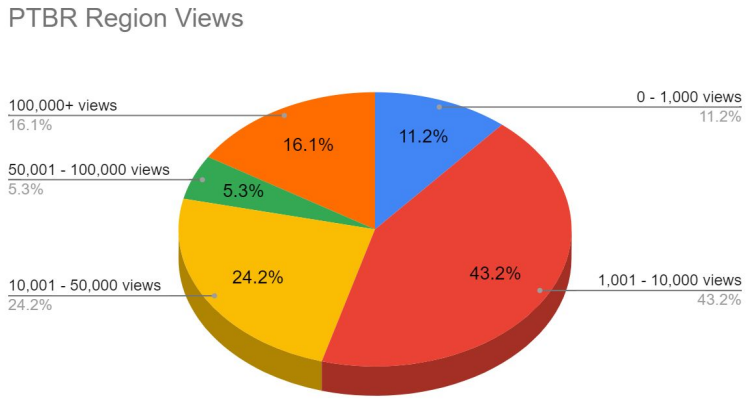


**Figure 4**

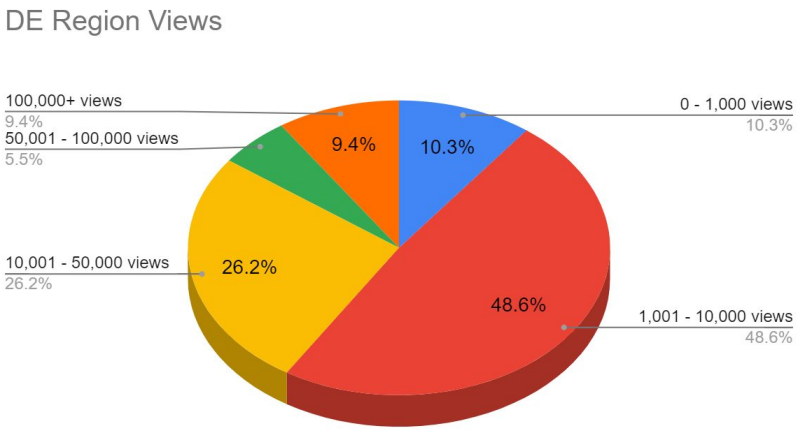



**Figure 5**

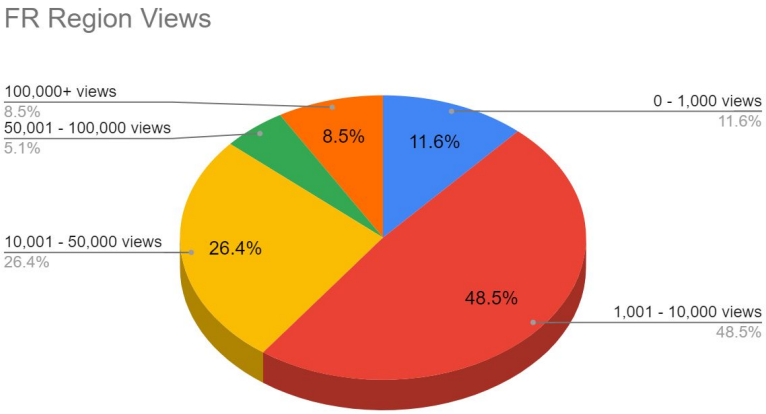



**Figure 6**

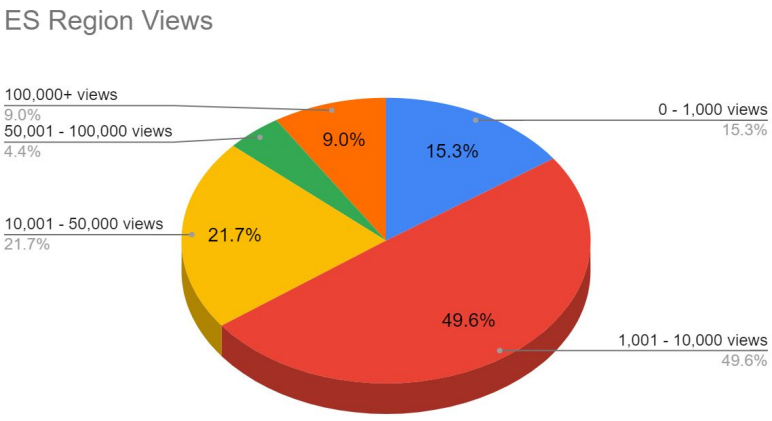


**Figure 7**

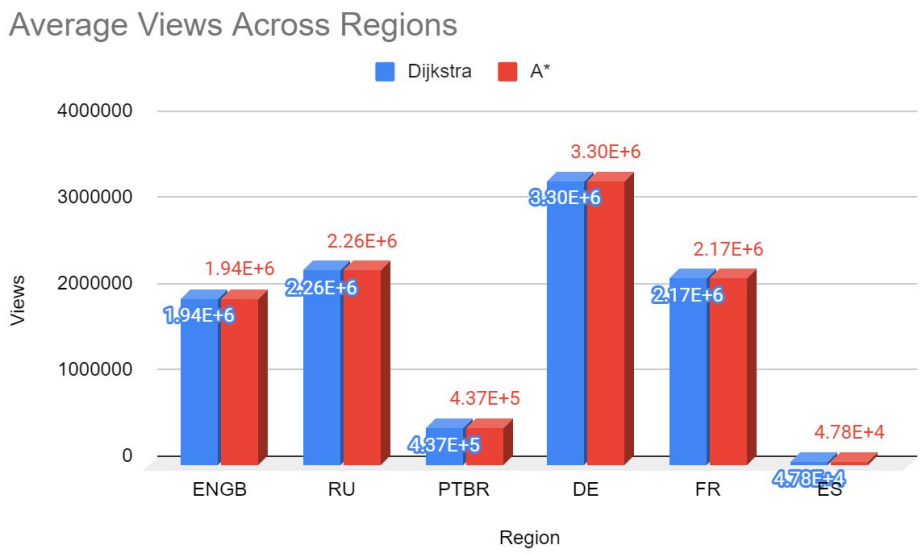



**Figure 8**