Varun Nadgir

Capstone 1 Data Wrangling

The dataset I am using for my first Capstone project is the League of Legends competitive e-sports data that is made available thanks to OraclesElixir. League of Legends is a 5v5 strategy game where players select a "Champion" to play as. The final goal is to break the opposing team's "Nexus", a structure in the enemy's base that is defended by various other structures and the opponents themselves. As the competitive scene has grown, teams have hired analysts to determine which "Champions" are the most likely to achieve victory under the game's current conditions (the game releases new "patches" frequently). Using the data available from early 2018 competitive matches, I would like to run analysis of my own.

The data from OraclesElixir comes from each of the major regions and a few minor leagues as well. The shape of the dataset is 94 columns by 15,013 rows, where each match has its own match ID. At a first glance, each match appears to have 12 entries - 10 for each of the individual players' statistics plus 2 entries for whole team averages. My approach will be to write the code for the Korean league, or "LCK", and then replace the relevant lines of code to be able to quickly build the same subsets, tables, and graphs for other regions.

Separating by region is a good start to be able to work with a smaller dataset that is still statistically significant on its own. Next, we can look at ignoring some columns depending on the context of our question. If I just want to compare the average game times and frequently picked "Champions", I can take just the first few columns and ignore the individual player statistics. On the other hand, if I want to compare players of the same role (of which there are 5), I could also choose to ignore other columns that are not relevant to their role.

In this dataset, there is not really an issue of missing data points exactly, but the Chinese region, or "LPL", has its data stored differently than the other regions. Its unique IDs are not a numerical ID, but rather an alphanumeric string built from the date and team names for the match. These matches also do not have a URL for the match page provided, unlike the others. This doesn't present any difficult in analyzing the data, but it is confusing at first when trying to filter by regions and IDs. Ultimately, the regions will get analyzed individually before comparing them to each other anyway, so this is easy to work around.

In addition to the raw data provided, I also used Python's .groupby() method to be able to aggregate players' statistics by "Champion" played. This is so that we can determine which players perform best on which "Champions". For example, just by taking the count of games and count of wins a player has on a "Champion", we can calculate a basic win-rate.

For the most part, the data collected by OraclesElixir is fairly clean to begin with. The bulk of the work is in appropriately subsetting the data for proper analysis. Once I have the lines of code to prepare the "LCK" team/player data, it is a simple task to apply it to the other regions. At that point, tables can be joined as necessary to compare teams/players across regions.