Analysis of Data Gathered from League of Legends and the Riot Games API

A Major Qualifying Project

Submitted to the Faculty of

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the Degree in Bachelor of Science

By

_____

Artian Kica

_____

Andrew La Manna

_____

Lindsay O'Donnell

_____

Tom Paolillo

Date: 3/15/16

Project Advisor:

_____

Professor Mark Claypool, Advisor

Abstract

Understanding the impact of game balance changes on character performance is important for game designers and players. For this study, we analyzed the effect of patches on champion win rates, pick rates, and ban rates from League of Legends, a popular online game created by Riot Games. This was achieved by categorizing patches by change type. These changes were correlated to game statistics sampled by pulling statistics from 128 champions and roughly 465,000 games from the Riot Games API. In order to be more conveniently accessible to the developers and playerbase, this information is displayed publically on a WPI hosted website on a per character basis. Analysis of the data shows that the rates of a champion have weak correlation with one another and that rates vary depending on the role of the champion being analyzed. The data also shows that Riot Games tends to increase the strength of champions more often than they decrease the strength of champions and that these increases and decreases have a weak correlation with win rate respectively.

Table of Contents

1 Introduction

Games have been an integral part of human culture since prehistory. Many of these games pit human minds, bodies, or both against one another in some form of competition. With competitive games, the issue of how to balance them as fairly as possible arises. Here, balance is defined as adjusting the rules to ensure there is no dominant strategy. A dominant strategy gives anyone using it a much higher chance of winning, independent of luck and skill. Balancing also includes supporting multiple viable strategies, which might mean making some stronger and others weaker to achieve this goal. Some games involve random elements to increase uncertainty in the game outcome, while others intentionally start the game with players on uneven ground; perhaps not with any advantage, but with different starting conditions. In some games, perfect balance and fairness is impossible. For example, in a turn based game, such as chess, one player must go first. The goal of balance is for players of roughly equal skill to have an approximately equal chance of winning and for the game to offer multiple viable strategy options to each player.[1] For example, if one character is very strong, and only one player can pick any one character, then the player that is randomly chosen to pick his or her character first is very likely to win through luck alone. In a competitive game, skill should be the main deciding factor of the game's winner; this is the intent of game balance.

[1] Sirlin, David. "Balancing Multiplayer Games, Part 1: Definitions." RSS. N.p., n.d. Web. 14 Oct. 2015.

1.1 Game Balance and Video Games

When games become more complex, keeping the game from having any one dominant strategy is even more difficult. Many modern video games, unlike board games, can be quickly and easily changed if the game is unbalanced. In order to make appropriate changes, the game developers closely study the state of the game. Many games start out in alpha and beta development before official release; these versions are used to fix bugs and polish the game itself, but are also important to develop a baseline for balance. The developers need to find out if all the varied strategies they have programmed in are equally powerful when the game is played by actual humans. To remedy this, most game studios employ playtesters at various points in development to observe the effectiveness of various strategies and characters. Changes can be made at this point to eliminate vastly superior strategies. Once a game is released, any further changes or lack thereof rely on the intent of its developers. Some games are released and never modified again by the developers, so any bugs or exploits persist. However, with the introduction of online gaming, some games are hosted on a server, downloaded through a client, and played by players around the world. This model allows the modification of game features server-side after it is released.

In this paper, the online game League of Legends (LoL), developed by Riot Games in 2009, will be our case study in how changes to the game affect the game balance. In LoL, balance changes, bug fixes, and aesthetic changes are made in large groups, called patches, approximately every 2-3 weeks. These rapid changes to the game mean that its balance is in a constant state of flux as players adapt and learn. League was chosen over other similar games for several reasons. The main reason is everyone in our group had played and enjoyed the game and would to bring a more passionate and informed view to this project than they would

with another game. An important aspect was also Riot Games' API, which allows the public to easily make queries and pull information from their servers. Another great selling point is the fact that League of Legends is one of the most popular modern day eSports, meaning its balance updates affect people's lives on a professional level. Providing insight to the effects of those changes could be very significant to these players.

1.2 An Introduction to League of Legends

League of Legends is known as a Multiplayer Online Battle Arena, or MOBA. This means that multiple players play in the same game, connected through the Internet. Each player controls his or her own character and engages in combat with enemy characters. The ultimate goal of the game is to destroy enemy structures leading to, and including, the enemy's Nexus (team base). In a standard game, each of the two teams has five players. Each of these players selects a champion, one of 128 playable characters[2], before starting each game. Each team can ban three champions before the selection process begins. Once picked, a champion cannot be selected again by either team. After champion selection, the game begins, and one team will eventually win. As in many games, champions have attributes that allow them to fulfill certain roles in the game. The most basic roles include damage dealers, tanks (those who can take a lot of damage), and utility (healing friendly champions or controlling enemy champions). Some champions can fulfill different roles in different games depending on how they are played.

With 128 characters, and more being added periodically, the developers must carefully monitor the state of the game to determine if a champion is too powerful. If certain champions are much stronger, the team that is able to select those champions first has a much higher

[2] As of February 2016

chance of winning the game, independent of skill level. Therefore, patches to the game frequently adjust champion stats and abilities to try to buff less powerful champions (increase their strength) and nerf more powerful champions (reduce their strength).

Player perception of champion strength is important as well, and a champion's ease of use affects balancing at different skill levels. Often, there is a perception of a champion being overpowered (too strong) after some change is made to the game, causing that champion to be banned and picked much more often than before. However, this perception may or may not agree with the actual data. Champion win rates (how often that champion is on the winning side of a match, as a percentage) are available through third party websites, but usually only for the most recently released patch. Historical win rates are much more difficult, and often impossible, to obtain. This data is valuable to players, so that they can see past trends and how each patch truly affected the game, rather than relying on community opinions. This data is also valuable to game developers and designers trying to determine what kinds of changes to make. Although Riot Games has access to these statistics, they may or may not be presented in a manner that is easy to read and process. Developers of other similar competitive games with frequent balance changes may be inexperienced in how to make good changes that are healthy for the game, and looking at data like this may help them as well.

1.3 Other Websites that Use the Riot API

Many websites present data gathered using the public Riot Games API[3], which allows collection of data about summoners (the LoL term for players), champions, and past matches. Some of these websites analyze the data or present it visually, while others just display the data itself. Many websites are searchable by summoner name, a unique username identifying a

[3] https://developer.riotgames.com/

player's account, and provide information about that player. The information might be general, such as a record of past games or most played champions, or it might focus on just a single specific aspect of that player, such as number of wards placed. Other websites present data about champions such as win rates, item builds, and kill death assist ratios. Much of the data collected by the websites is similar, so only a few of the most widely used ones are described here.

Two popular websites, Lolking.net[4] and op.gg[5], present data about individual summoners, their match history, and the other players in the game currently being played(if they are in game). Neither has comprehensive historical data concerning the changes made to champions, items, or the game engine. and for both, the player data only goes back as far as the website itself has stored it. Lolking does have some champion data, but only lists a single win rate, pick rate, and ban rate for each champion based on data recorded weekly. Champion.gg presents data about each champion's current win rate, pick rate, and ban rate for the past 5 patches. It also shows item builds, runes, masteries, and other data about how the champions were played.

What all of these websites are missing is the presentation of historical and current win rate data alongside the changes made at that time, to provide context to the state of the champion and the game at that point. They also do not analyze the patch content in any way. These websites, while useful resources, require that the user find the patch notes for their champion of interest in order to compare the changes themselves, and do not hold data for more than a few months. Some users may only wish to know about the current state of the game, but studying past changes can inform how current changes might affect the game

[4] http://www.lolking.net/
[5] http://na.op.gg

similarly. This can have important effects not only for average players, but for professional players and the game's developers, whose livelihood rests on the game's continual popularity.

1.4 LoL Crawler

Our project is to develop a website that presents the win rate, pick rate, ban rate, and patch note data for each champion, going back approximately 2 years (the limit that the Riot Games API allows). In order obtain data to analyze, we first needed to choose a sample of players to pull information from. Due to the large number of players, we could not include every player available, so we obtained a sample that is representative of the entire population. We obtained summoner account information using the Riot API and then used the match history of each account to pull information about the games played. Specifically, we gathered data for each team regarding champions picked, champions banned, and whether or not that team won. We also collected a chronological, categorized patch history for each champion. This information is not contained in the API, so we instead used a series of scripts to parse the information from the League of Legends Wiki[6]. The Wiki is a community edited resource with information on each champion, all obtained from patch notes released by Riot. We categorized the patch notes and stored them in a database. These rates and patch notes are displayed together on a graph to allow further interpretation and analysis. The user can filter which rates are being displayed.

The Riot Games API does not facilitate querying for an arbitrary amount of games. Therefore, we developed a random sampling method to obtain a sample of games and store the champion data associated with those games. This sampling method is described in detail in Section 3.1. With the sampling method, we were able pull a sample of games of a statistically

[6] http://leagueoflegends.wikia.com/

sufficient size. This was important as the sample needed to represent the general League of Legends playerbase as accurately as possible.

From each game we were then able look at data for each champion involved. Within a game, ten champions can be picked, six champions can be banned, and five champions can win. With the sample of games, we calculated win rate, pick rate, and ban rate for each champion on each patch. This data is a representation of champion performance. Win rates typically are associated with the actual strength and performance of a champion. Pick and ban rates typically are associated with how popular a champion is or strong it is perceived to be. These statistics were essential for the final analysis as they represent a champion's "success".

We have created a website that displays champion changes and their effects on the balance of the game in a concise and meaningful way. This allows players and developers to understand the state of the game through interesting observations of the outcomes of certain changes. For the developers, these observations also offer insight on the success of past changes and how to approach future changes to the game.

Chapter 2 of this paper covers background information on League of Legends game mechanics as well as research on game balance and the Riot Games API. Chapter 3 describes the methodology used to collect and categorize the data presented on our website. Chapter 4 analyzes our findings, plotting relationships between change types and resulting win, pick, and ban rates. Chapter 5 summarizes our conclusions and presents possible future work.

2 Background

The background chapter presents various information necessary to understand the project methodology, results, and conclusions. Section 2.1 describes League of Legends' game mechanics, ranking system, and other nuances of the game. Section 2.2 characterizes similar projects' features and shortcomings. Section 2.3 presents our research on game balance. Section 2.4 outlines the capabilities and limitations of the Riot Games API.

2.1 League of Legends

League of Legends is an intricate game that requires knowledge of many elements to be played proficiently. In the game, two teams of five players each battle against each other on a battlefield called Summoner's Rift, displayed below in Figure 1. There are other battlefields with different objectives and layouts, but they will not be discussed in this paper. Summoner's Rift is comprised of each team's base, three lanes, and the jungle. The goal of League of Legends is to destroy the opposing team's Nexus, a structure located in the enemy base.

A team's base contains many important structures that the team must protect. The bases are located in the bottom-left and top-right corners of the map respectively. Each base contains its team's Nexus. If a team's Nexus is destroyed, that team loses the match. However, the Nexus is not unguarded. Also inside the base are three defensive structures called inhibitors, which are seen in Figure 1 as glowing structures surrounding the nexus. In order to damage the Nexus, at least one inhibitor must be destroyed. Inhibitors are rebuilt five minutes after they are destroyed.

Figure 1: Summoner's Rift

Exiting each base are three paths called lanes that lead to the opposing base. Each lane contains three structures called towers leading up to and defending a team's base. Unlike inhibitors, towers automatically attack enemy units that come too close. Towers can only be damaged in order, starting from the tower furthest from the base. Towers must also be destroyed if a team wishes to destroy the opposing team's inhibitors. If all three towers in a lane are destroyed, then the inhibitor which that lane leads up to will be able to be damaged.

Each team's base periodically generates waves of non-player characters (NPCs) called minions. Minions will exit the base through each of the three lanes and continue traveling through the lane until they reach an enemy unit or structure. Enemy towers will first target a team's minions, allowing a team to damage opposing towers without taking damage from the turrets themselves. When an enemy inhibitor is destroyed, minions generated in that lane will be slightly more powerful. Additionally, a team's base will generate a more powerful minion called a

super minion along with each minion wave in that lane. If all three enemy inhibitors are destroyed, each wave will spawn two super minions in each lane instead of one.

Between the three lanes are areas which make up the jungle. Inside the jungle are other various NPCs called jungle monsters. However, jungle monsters are not affiliated with either team and will not attack a player unless provoked. They are generated in predefined areas in the jungle and do not move from their respective areas. Some jungle monsters offer unique powers when slain. Upon being slain, jungle monsters will be replaced by more powerful versions of the slain monsters after a certain amount of time. This amount of time is different for each type of jungle monster.

Champions start the game at level 1 and progress to the level cap of 18 by gaining experience. Experience can be gained by destroying units such as minions, jungle monsters, or enemy champions. With each level, the champion becomes more powerful and is granted a skill point which can be used to improve their abilities.

Each champion has a unique set of abilities that can be used to hurt enemies, hinder enemies, or assist allies. Champions have four main abilities and an innate ability. The innate ability is unlocked at the start of the game and does not need to be unlocked. However, each of the four main abilities must be unlocked by putting at least one skill point into the ability and increasing its level to one. Additional skill points beyond the first will level up the ability to make it more powerful. The first three main abilities can reach level five and the fourth ability can reach level three. The fourth ability is a champion’s ultimate ability. Ultimate abilities are generally a champion’s strongest and most iconic ability. As a result, ultimate abilities can only be leveled up every sixth level.

All champions can also deal damage with their basic attack. Each champion's basic attack has a different range and deals damage with each hit. Upon completing a basic attack, a player cannot perform another basic attack for a small period of time.

Every champion also has the same set of attributes which can be selectively improved and optimized for playing that champion. These are: health, mana, health regeneration, mana regeneration, attack damage, ability power, attack speed, cooldown reduction, critical strike chance, armor, magic resistance, movement speed, lifesteal, and spell vamp. Health determines how much damage a champion can take before they are slain. Mana determines the amount of resources you have left to cast abilities. Health and mana regeneration determine the rate at which their respective attributes passively replenish themselves over time. Attack damage determines the amount of damage a champion's basic attacks deal. Attack damage and ability power may increase the amount of damage a champion's abilities deal. Whether the ability is improved by attack damage, ability power, both attributes, or another champion stat is dependent on the ability. Attack speed reduces the amount of time that a champion must wait after a basic attack to execute another basic attack. Cooldown reduction reduces the amount of time that a champion must wait after using an ability in order to be able to use that ability again. Critical strike chance determines the likelihood of landing a critical strike with a basic attack, which makes the basic attack deal double damage. Armor and Magic Resistance reduce the amount of damage that a champion takes. Movement speed determines the speed at which a champion can move across Summoner's Rift. Lifesteal and spell vamp allow a champion to replenish their health with a percentage of the damage they deal with basic attacks and abilities respectively. Champions with different roles have different attributes emphasized. For example, a tank has weak killing power but is extremely durable, while marksmen are adept at dealing damage and are not very durable. The champion Amumu, who is a tank, will want to build up as

much health, magic resistance, and armor as possible. These stats will allow him to survive more damage so that he fulfills his role more effectively. The champion Ashe, who is a marksman, will want to build up as much attack damage, attack speed, and critical strike chance as possible. These stats will allow her to deal more damage at a faster rate so that she fulfills her role more effectively.

Players can also strengthen their champion through runes and masteries. Runes provide small bonuses to specific attributes and are grouped into sets called rune pages. Rune pages are selected before the game and are comprised of up to 30 runes. Masteries also provide bonuses but are often more unique and interesting than simply increasing the strength of an attribute. Examples of masteries include extra damage to low health enemies or replenishing health upon slaying an enemy champion. Runes and masteries are intended to complement the type of champion the player is playing and can be tuned to suit various playstyles. They allow players to personalize their champions and gain some type of advantage over the other players early on.

While a champion can be made more powerful via runes and masteries, the vast majority of a champion's power is derived from the items the champion has purchased within the game. A champion can hold up to six items which can be bought from a shop behind their Nexus. Items are purchased with gold which can be acquired by slaying units and destroying structures. Gold is also passively generated over time at a slow rate. A champion starts each game with no items but a small amount of gold. Each champion can only hold six items at once, so optimizing which ones to buy is an important part of the game's strategy. Basic items often simply improve a champion's attributes but can be combined in order to create more powerful items with more interesting effects. Examples of unique item effects include increasing critical strike damage and gaining the ability to make oneself invulnerable for a brief period.

League of Legends also has a system for ranking players by skill level. Players can play ranked games and normal games. Ranked games will affect a player's rank while normal games are just played for fun. A player's profile will start out unranked, but after playing ten ranked games it will be assigned a rank. There are seven tiers for ranking players, most having five divisions each. In order of least skillful to most skillful, the tiers are Bronze, Silver, Gold, Platinum, and Diamond. Each of these tiers has divisions I, II, III, IV, and V with V being the least skillful and I being the most skillful. After Diamond I, players can be placed into the Master tier, a tier for players who are close to being accepted into the highest tier when room is available. The highest possible tier is Challenger. Challenger is made up of the top 200 players and is capped at that amount. When players in Master tier surpass any of the 200 from Challenger, those players swap places. The Master and Challenger tiers have only one division, as they are very exclusive. Players advance through divisions by earning League Points (LP) for winning games and losing LP for losing games. Players start at 0 LP in a division and get a chance to advance divisions when they reach 100 LP. The amount of LP earned or lost in a match is dependent on the rankings of the players on each team. At 100 LP, the player is put into a best of three series to determine their promotion to the next division, or a best of five series if determining a promotion to another tier. This means they must win at least two of their next three matches in the case of a best of three, or three of their next five matches in the case of a best of five. If the player loses their series, they remain in their current division and lose a small portion of LP.

League of Legends is played all around the world by millions of players. To counter latency and avoid language barriers, Riot Games hosts ten official servers in many different countries and a player's profile can only be affiliated with one of them at a time. These servers are known as North America, Europe West, Europe Nordic & East, Korea, Russia, Turkey, Latin

America North, Latin America South, Oceania, and Brazil. Players can only play with other players who are on their server, meaning European players cannot play with North American players unless they transfer their profile to the North American server. The most popular servers are Korea, Europe West, North America, and Europe Nordic & East. In our study, we gathered data from the North American servers as this was the region we were most familiar with.

To address game balance issues and provide new content for players, League of Legends is consistently updated through sets of changes called patches. Each patch consists of changes or additions to the game. Every game of League of Legends is played on the most recent patch, meaning the game is using the most recent changes to the game. This frequent patching schedule combined with the public API is motivation for using League of Legends for this project.

2.2 Other Websites that Use the Riot Games API

LoLKing.net[7] provides a searchable database of players as well as champions. It also acts as a hub for discussion and activities regarding League of Legends. Player profiles present each player's current ranking, wins and losses with each champion played, and average kills, deaths, and assists with each champion played. For champions, win rate and play rate (the percentage of games in which that champion was played) are shown in a graph for the last month, and the user can isolate different skill levels by rank using filters. Also included are average kills, deaths, and assists for that champion across all players. LoLKing also includes player written guides about how to play each champion. LoLKing is a useful resource for players that want to keep track of their own statistics, or see how they compare to others. However, it does not present any data concerning the patches themselves. Below in Figures 2 and 3 are

[7] http://www.lolking.net/

screenshots of LoLKing.net. Figure 2 displays the home page and an article analyzing the most recent patch notes. Figure 3 displays the rating and statistics of a player named “SSBM Hax”.

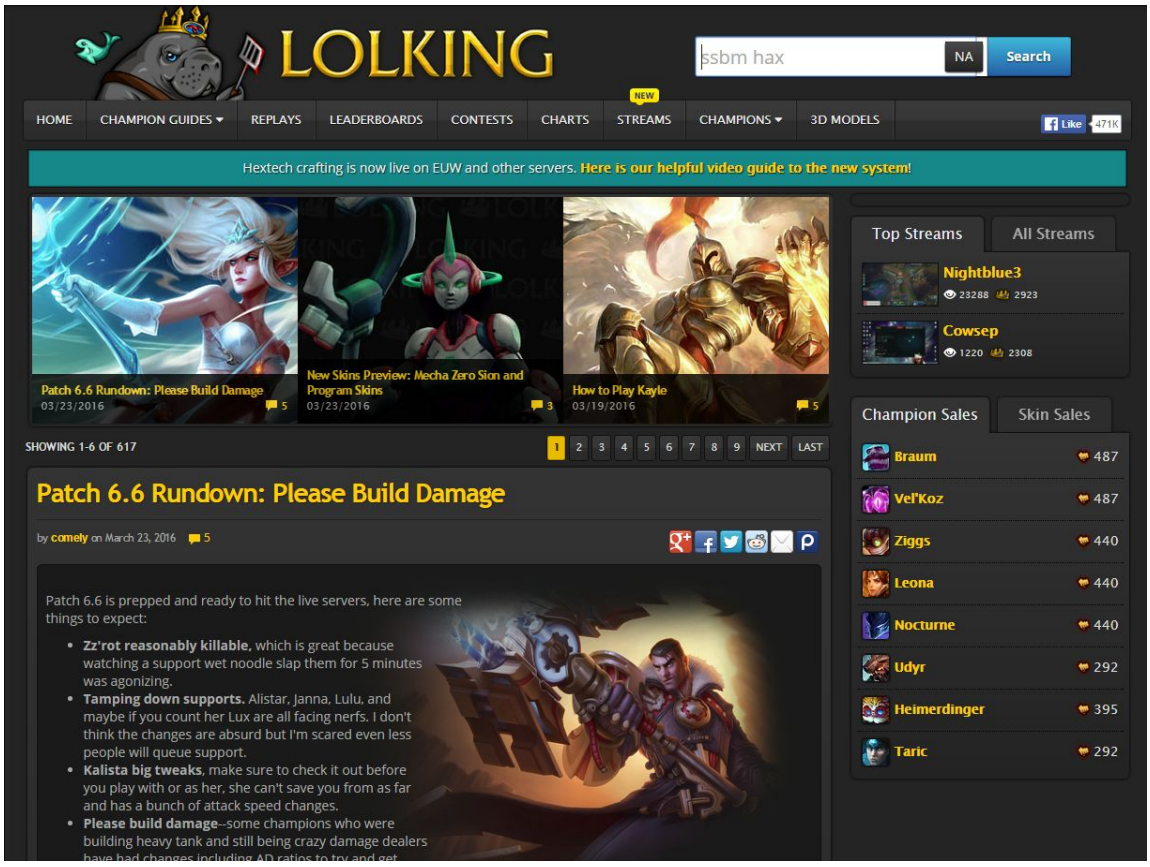

Figure 2: LoLKing.net homepage

Figure 3: LoLKing.net player profile

Similarly to LoLKing, op.gg[8] is another website that allows players to look up a summoner by name and presents data about that summoner. Also shown on op.gg are the names and ranks of players in the game that the searched summoner is currently playing in, allowing research into the other team's skill as well as their skill with the champions they have chosen to play in the current game. Another difference between the two sites is that op.gg shows the summoner's match history as far back as the API will allow (usually several months back). In the League of Legends client itself, only the last 20 games are shown. While it is a useful resource for data about individual players, op.gg does not analyze any data about

[8] http://op.gg/

champions, patches, or the game's balance as a whole. Below in Figure 4 is the same player as displayed in Figure 3, but on op.gg.

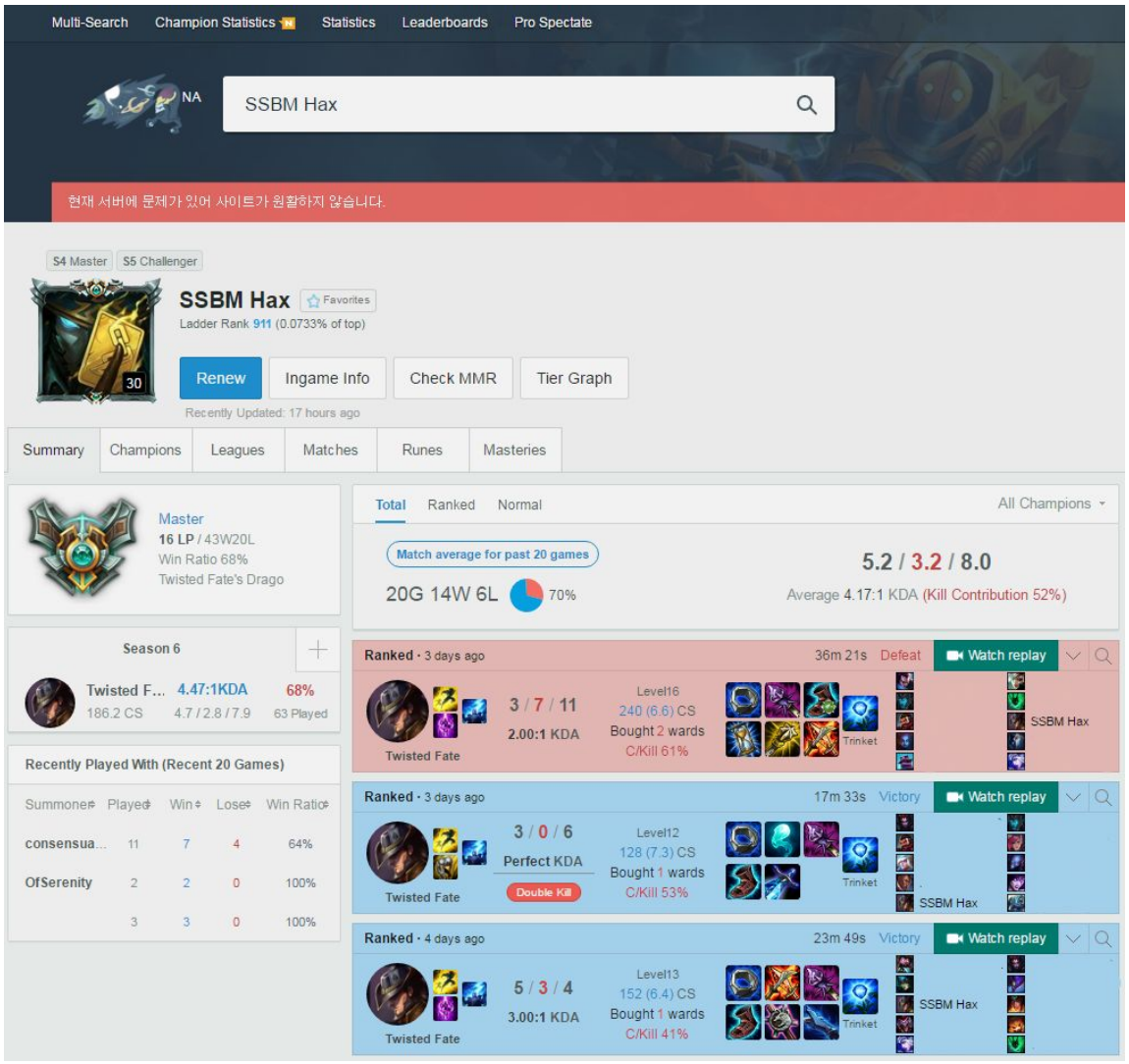

Figure 4: op.gg player profile

Champion.gg[9] presents data about each champion, which includes current win rate, play rate, and ban rate (the percentage of games in which that champion was banned). It provides basic performance statistics such as average kills, deaths, and assists earned per match, and compares these with other champions which fulfill the same role in the game. Champion.gg

[9] http://champion.gg/

shows the win rate for the past 5 patches, as well as win rates based on game length and player experience with that champion. Finally, it shows the most common and most winning runes, masteries, and item builds for that champion. Champion.gg does not show the changes, known as patch notes, for each champion, nor do the win rates go back further than 5 previous patches. Below in Figures 5 and 6 are screenshots of Champion.gg. Figure 5 displays the home page and statistical overviews. Figure 6 displays in depth information about the champion Ashe.

Figure 5: Champion.gg homepage

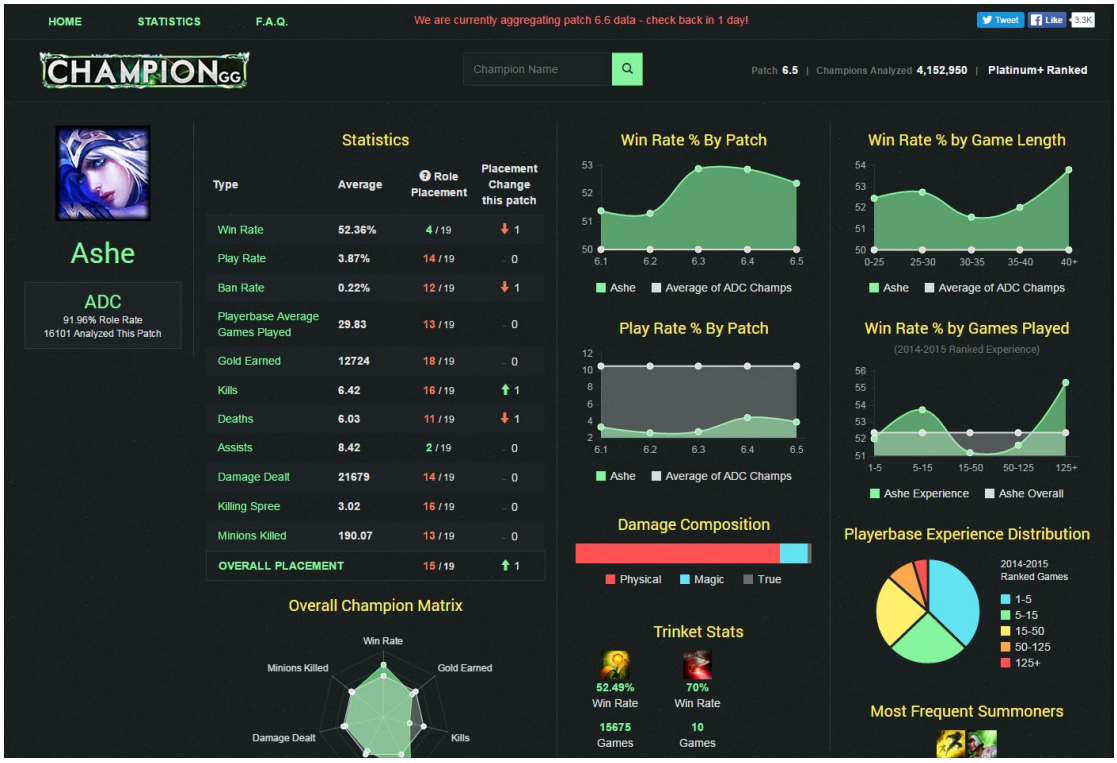

Figure 6: Champion.gg Champion Profile

2.3 Game Balance

The paper “Character Balance in MOBA Games”[10] investigates the balance of LoL as well as Dota 2, another popular MOBA game, using an algorithm called Metagame Bounds. This calculates the optimal play rates of various characters given their win rates against all of the other characters in the game. The authors gathered data by hand from a website that reported champion win rates and then entered the data into an algorithm to determine the results. The algorithm was developed for one versus one player competitive games such as fighting games, so after analyzing the data the authors commented that because MOBAs are team based games, a character could win their lane and still lose the game, but this would be

[10] Palm, Emanuel and Norén,Teodor. “Character Balance in MOBA Games” June, 2015. http://www.diva-portal.org/smash/get/diva2:818292/FULLTEXT01.pdf

counted as a win for the other character in the matchup. Overall the paper brings up some interesting results concerning balancing for high level and low level play. It does not comment on how patches specifically affect the game balance, or analyze past balance issues.

"Level 16: Game Balance"[11] is an article written as part of a educational website about game design. It explains that asymmetrical games are often the most difficult to balance because the players are starting with very different abilities, resources, and/or positions. It comments that in games with items, a certain item always being bought early by many players may indicate that the item is too strong. LoL has some items that are rarely bought, and some that are almost always bought first on a certain champion because of synergies with that champion's abilities. Although we do not plan to directly look at every item, looking at certain item buffs and nerfs in relation to the champions those items are very popular on may help explain win rate changes for that champion. The article also covers some ideas to help game designers balance their games, including statistical analysis and playtesting. For items, often the cost is in some way proportional to the benefits the item gives, so math can be used to calculate the correct cost and benefit ratio so that different items all benefit the player equally. However, the article mentions that some items may have unique effects not seen elsewhere in the game, so figuring out how to balance the cost of these items appropriately requires playtesting. Riot Games playtests using their Public Beta Environment, a server that receives daily, often experimental, changes to the game. This allows them to see how various changes may affect particular items and champions.

[11] https://gamedesignconcepts.wordpress.com/2009/08/20/level-16-game-balance

2.4 The Riot Games API

Riot's API is a tool for the public to access game data in a secure and reliable way. There is a large amount of data that can be gathered using the API, but there are also limitations. This section contains a brief overview of what the API can retrieve. For a more extensive list of exactly what can be retrieved and how that is done, see Appendix A.

Through the API, we can access an assortment of data within any ranked game a given summoner has played from Season 4 onwards. Prior to Season 4, data about the games is incomplete and not able to be used to calculate win rates. Games played in Season 4 or later include general data such as the game type, the players involved, the champions played, the team that won, the patch the game was played on, and other similar information. It also offers more detailed information such as the scores of each player, the damage a player dealt, the damage a player received, the items purchased, and much more. All of this information can also be retrieved for normal games, but only for the last 10 games of a given summoner. This limits what claims we can make about data of normal games in the past.

While we can still access all ranked games a summoner has played, we mentioned above that data that can be gathered from games prior to Season 4 is incomplete. At the start of Season 4 in 2014, Riot Games changed what the game kept track of and the Riot Games API received an overhaul. This allowed the public to gather more advanced data about games and players. Prior to this point, information that can be retrieved is limited. Specifically, win rates for a given champion cannot be calculated before this point because wins and losses by player were not recorded. This explains why the previously mentioned websites that use the API change what kind of information they display when observing statistics before Season 4.

While gathering information about games is the core of this project, we cannot simply retrieve all games played. In order to retrieve a game, we must first retrieve a summoner and that summoner is linked to their ranked and recent games. Unfortunately, we also do not have access to all summoners, so we developed a method to gather large samples of summoners. This is described further in section 3.1.

3 Methodology

In order to collect data and build our website to display that data, we researched relevant technologies and softwares and weighed the pros and cons for each. In this section, we outline our decisions in approaching these tasks and describe the final result.

3.1 Sampling with the Riot API

The most important aspect of statistical analysis is ensuring a sample is as representative of the population as possible. This can prove difficult in our case because the player-base of this game varies extremely in skill level and this variance may or may not be normally distributed. For our purposes, we are not analyzing the entire population as a single group. Instead, we are collecting a sample of all players across ranks and using their ranked games to gather champion statistics.

The Riot API does not have the built in functionality to retrieve an arbitrary amount of randomly chosen games. We started with “seed” players of various ranks, randomly selected using a section of Lolking.net that lists all players on each server by rank. One player of each rank, excluding Master and Challenger, was chosen as a seed to create a list of 25 players. We then used each player’s ranked match history to collect a number of players around that ranking, eliminating any duplicate players. This process was then repeated until a desired number of players was found; approximately 11,000 players total. Then, we combined a random sample of games in the histories of all of the players found using this method. The end result was a set of 465,000 ranked games from Season 4 to the present. Once the list of games was compiled, each champion’s win/pick/ban rate in that set of games was calculated. To determine

when our sample was large enough, we continued to collect games until the win rate leveled off and no longer changed significantly with the number of games. This process was completed using Python to query the API for data as well as store the data and use it to calculate the win, pick, and ban rates.

3.2 Categorization of Patch Notes

One of the unique features we offered in this project was the categorization of changes for every patch. While a human should be able to read a change in a patch-log and fairly easily assess if it is beneficial or detrimental to the champion, we attempted to automate this process to the best of our ability. Below we have described the guidelines we use to automate the process. While these guidelines are not all encompassing and there may be false negatives, they are accurate to the point where there will be few false positives and plentiful true positives. False negatives occur when a change is not detected and no categorization is made. False positives occur when a change is detected but is incorrectly categorized. True positives occur when a change is detected and correctly categorized.

The majority of balance changes in the game can be categorized as number changes. This means the quantifiable strength of a spell or stat is being changed. Luckily, this is also the easiest type of change to automatically detect. The patch notes are consistent in the language they use in relation to this type of change, using the terms “increased” to signify a buff, “reduced” to signify a nerf, and “modified” to signify a neutral change. For some cases, the logic for these keywords is reversed as reducing the value is a buff and increasing it is a nerf (for example, the cost or cooldown of a spell). This exception can be accounted for by detecting what value is being changed and deciding whether increasing it would be a buff or a nerf.

As an example of a number change, the patch note, “Base armor increased to 23 from 19” would be categorized as a number buff, as the quantity of this champion’s armor is being increased. Had the values been decreased, it would be categorized as a number nerf. Sometimes, values are changed such that it is neither a buff nor a nerf, making it a neutral change. For example, “Base damage changed to 150 / 300 / 450 from 100 / 300 / 500” would be categorized as a neutral number change. This is because the ability is being buffed at level 1, going from 100 to 150 damage, but also nerfed at level 3, going from 500 to 450 damage.

The next most frequent changes are bug fixes and visual updates. These are also fairly easy to automatically categorize because Riot has developed a fairly consistent language when introducing them. The terms “fix(ed)” and “bug” always signify a bug fix change. The terms “visual”, “animation”, and “update(d)” always signify a visual update.

For example, the patch note, “Fixed a bug where interrupting Vault Breaker would sometimes render Vi unable to cast spells” would be categorized as a bug fix because it a fix for an intended feature which is not working properly. The patch note “Flamespitter: new particle animations” would be categorized as a visual update because it is only changing what the ability looks like.

The least frequent changes are quality of life changes and utility changes. Quality of life changes affect intuitiveness and ease of use of a champion while utility changes affect what a spell does or how it interacts with other components of the game. Quality of life changes are so rare and unique that there does not seem to be a consistent language that could be used to identify them accurately. Utility changes occur most often in champion reworks, which can help narrow down where they are more likely to appear. However, utility changes suffer from the same issue as quality of life changes, meaning they are fairly unique and specific to individual champions, so there are not any consistent keywords connecting the category of changes.

As an example of a utility change, the patch note, “Added a 20% slow on hit” would be categorized as a utility buff, as something about how the ability functions has changed and benefits the ability. Likewise, the removal of a slow would be categorized as a utility nerf. As an example of a quality of life change, the patch note, “Added a visual indicator to better determine where the ability will hit” would be classified as a quality of life buff, as it makes an ability easier to use without making it more powerful or changing how the ability functions.

Because the categorization process is very difficult to automate with complete precision, any remaining changes after the automation were manually categorized by the team. This means that maintenance of the website through manual patch note categorization will be consistently needed if it is to stay up to date. Because the team categorized the entire history of patch notes up to the present time, any future maintenance would take little effort. When manually categorizing the patch notes, our team also took down statistics of the true positive, false positive, and false negative rates of the automation system. Without human intervention, our automatic system classifies approximately 67% of all changes on average with a true positive rate of 63.18%, a false negative rate of 33%, and a false positive rate of 3.82%. From these statistics, the precision and recall can be calculated to be 94.3% and 65.7% respectively. These results are expected given the complexity of the task and are further proof that human intervention is needed to reach as close to 100% precision and recall as possible.

Patch notes are not available for retrieval through the Riot API, so the retrieval process has to be done through the League of Legends wiki, which keeps up to date patch histories of each champion. A combination of AutoIt[12] and Perl scripts are used to pull the patch history text off of each champion page on the wiki and parse it into a readable format with clear sections. This process produces a patch note file for each champion, which are in turn used as input files

[12] https://www.autoitscript.com/site/

for the patch note categorizer, another Perl script. The categorizer looks for keywords or combinations of words in each line to determine which category to apply the patch note to.

It attempts to categorize number changes, bug fixes, and visual updates as accurately as possible. It does not attempt quality of life changes or utility changes, as their inclusion greatly increases the rate of false positives. Of course, without including them, the rate of false negatives would increase in response, however, a high false negative rate is less detrimental than a high false positive rate for our purposes. It is easier for human categorizers to notice where the automation system is unable to classify a change than it is for them to notice when an existing classification is incorrect. After the categorizer script runs, it produces categorized patch note files, which are then looked over by the team to fill in the gaps it missed, correct any mistakes it made, and reformat the files to prepare them to be entered into our database. This process is shown below in Figure 7.

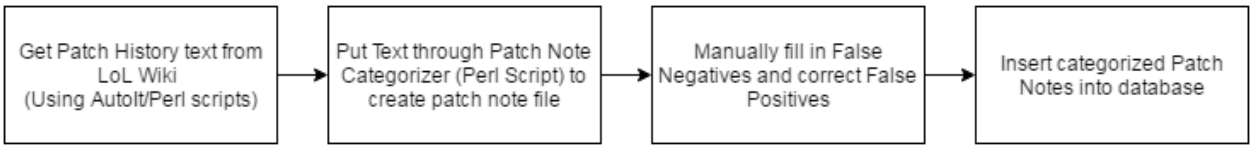

Figure 7: Patch Note Categorization Process

3.3 Database Implementation

The data described in the above sections needs to be stored if we are to have any use for it so we decided to store it in a database. We had considered Oracle, MySQL, SQL Server, and Flat Files for fulfilling our database needs. We decided on MySQL because it was said to be more intuitive and a better fit for smaller projects rather than long term projects. Additionally, it

was open source, supported PHP, and supported our virtual machine's OS. The pros and cons of the aforementioned database systems is included in Appendix B.

Below is the relational model we used to store the data. It is also displayed in the form of a diagram in Figure 8. The text outside the parenthesis is the name of that table. The text inside the parenthesis represents each of the columns in that table. Any underlined columns are the primary keys and any italicized columns are foreign keys.

- champion(<u>id</u>, picname, cname)
- patch(<u>id</u>, pname)
- patch_note_type(<u>id</u>, type)
- patch_note(<u>id</u>, note, *champion_id*, *patch_id*, *type_id*)
- rate_type(<u>id</u>, type)
- rate(<u>id</u>, percentage, *champion_id*, *patch_id*, *type_id*)

The champion and patch tables store the names and IDs of the champions and patches respectively. The names and champion IDs are provided by Riot Games. The "picname" attribute in the champion table is used to display a champion's name properly on the website. Riot stores champion names without spaces, so champions like "Twisted Fate" would display as "TwistedFate". Storing a custom picname with a space fixes this issue while still providing us with other conveniences afforded by keeping the cname.

The patch_note_type table contains all of the custom categorizations we described in Section 3.2. These are needed in order to categorize patch notes in the patch_note table. The patch_note table stores all the changes of each patch. Each tuple in this table is a single change made to a single champion on a specific patch.

The rate_type table stores the types of rates, namely win, pick, and ban. While these are the only rates we calculated in this project, this implementation allows for more types of rates to

be easily added. The rates table stores these rates as calculated by our sampling methods described in Section 3.1.

Figure 8 below is the same relational model shown above, but in the form of a diagram. Rectangles represent entities, or tables. Ovals represent attributes, or table columns. Diamonds represent relationships, or foreign key attributes.

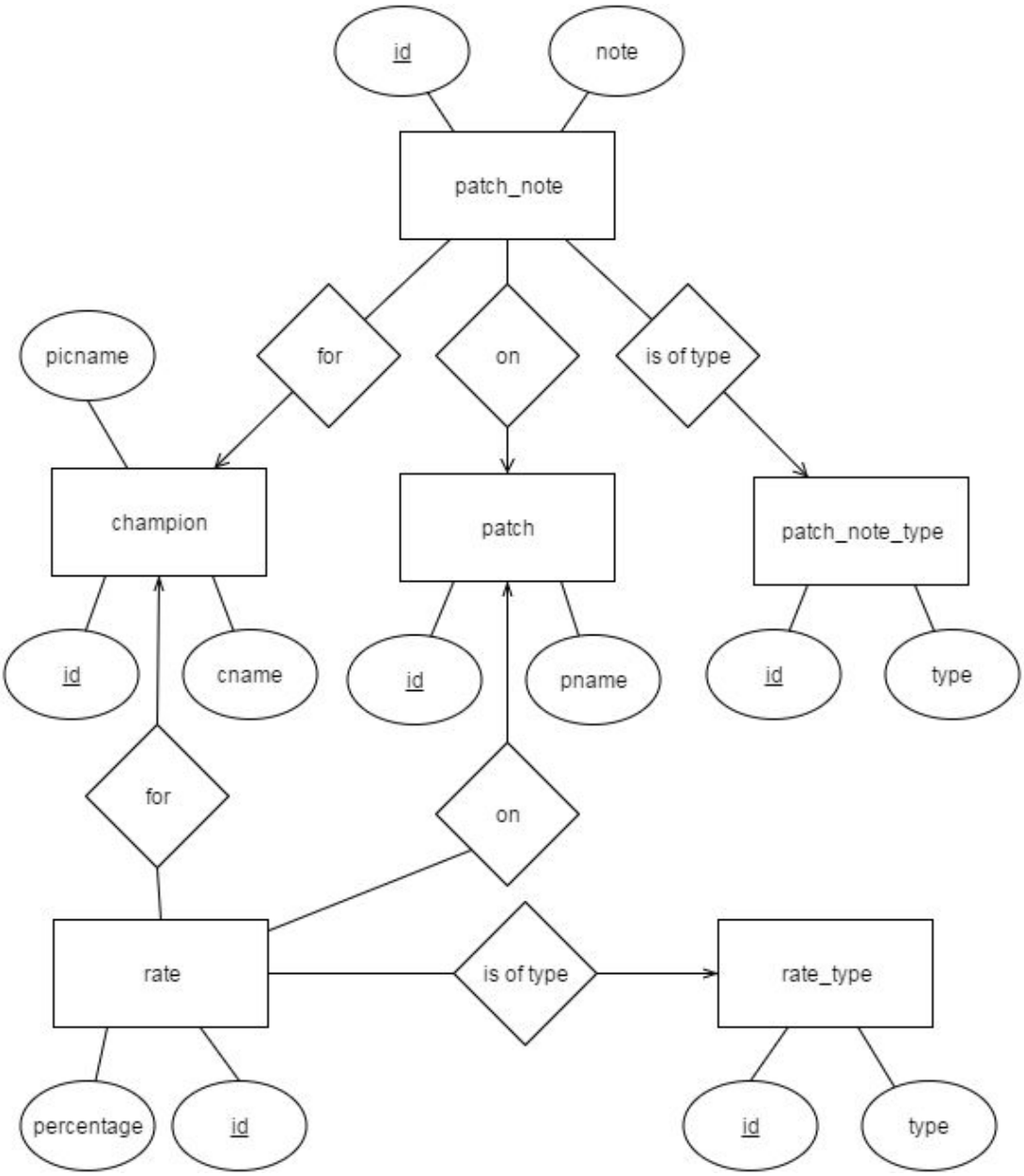

Figure 8: Database Relational Model

3.4 LoLCrawler Website

There were many decisions that needed to be made for the design and implementation of the website. First, we needed to decide the primary language for the server side code. The options consisted of PHP, nodeJS and C#. We decided to use PHP because it was the simplest to set up with the Linux virtual machine we were provided with. Team knowledge of the language was also considered but ease of implementation was a priority.

The website was designed in order to be intuitive while displaying a large amount of data. Being able to understand how to use the website just by looking at the layout was an important consideration when determining the design. We decided that having a page to select a champion and a page to display the data was the best approach to balancing intuitiveness and the amount of information displayed. Additionally, this is the same format in which the champions are displayed when a player is prompted to select a champion to play in the League of Legends client, making the layout familiar to the user. Included in Appendix C are original mockups of the website done in Photoshop.

When designing the home page, it was most important that users would know what to do as soon as the user saw the page. A list of champions and their images makes it more apparent to the user that the list can be interacted with in some way. Because of this, we decided to display clickable names and images in rows rather than other selection methods such as a drop down menu. In addition, users who are familiar with the game itself will be used to this method of display, as it is used similarly within the game when a champion is to be selected. Upon clicking one of the images or names, the user is directed to the data page for that champion. A screenshot of the homepage is displayed below in Figure 9.

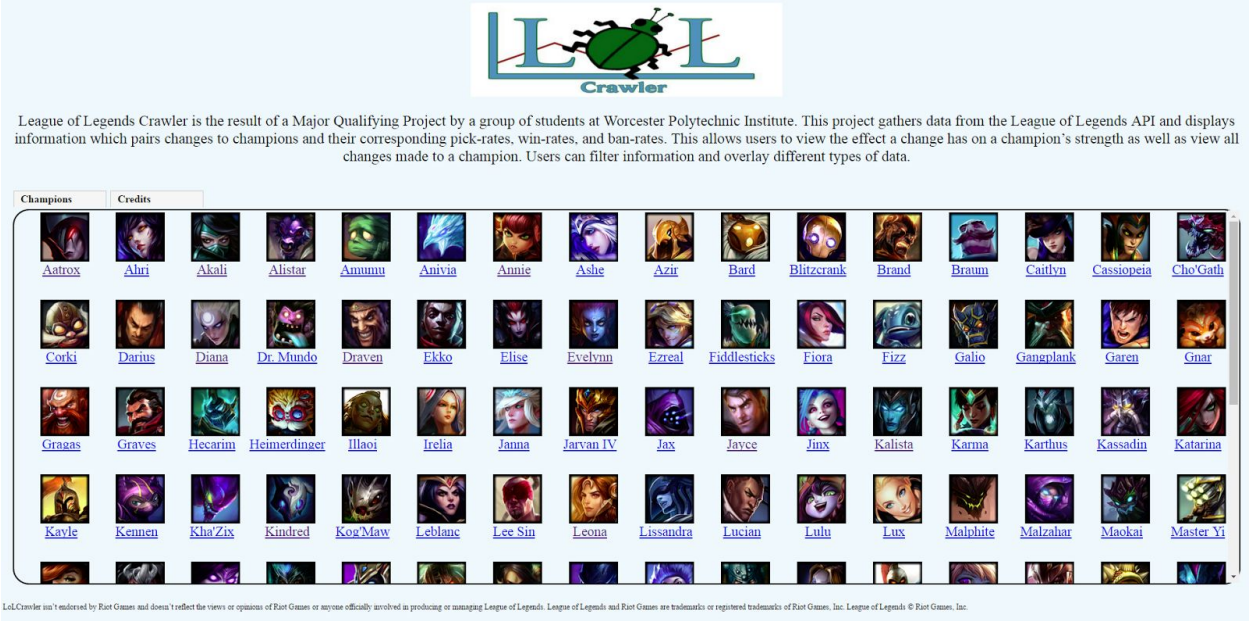

Figure 9: LoL Crawler Homepage

Because a key point of our website is the ability to display data related to changes over time, a graph would display this best with patch version on the x-axis and rate data on the y-axis. We wanted to be able to display any or all of the following: pick rates, ban rates, and win rates. The ability to display these statistics simultaneously can offer potentially interesting insight, so this functionality was important when deciding how to present the data; a graph seemed intuitive. Additionally, being able to interact with each data point was necessary in order to see what changes were affiliated with each time period. We decided to use a Javascript library called HighCharts[13] because it had all of the above functionality and was compatible with our server configuration. A screenshot of this page is displayed below in Figure 10.

[13] http://www.highcharts.com/

Figure 10: LoL Crawler Graph of Draven

A user can click on the legend to the right of the graph to enable or disable any of the rates and the graph will be changed dynamically. Additionally, hovering over a data point provides more information about the changes in that time period by showing a tooltip above that point. There are many patches on the x-axis, which would display awkwardly compacted all at once, so a scrollbar is implemented to allow the user to view smaller portions at a time. Patch notes are displayed in detail below the graph if the user wants to know more about the changes made. These changes also include our custom categorization of each change as well as the original patch notes text provided by Riot Games. If the user clicks on a point on the graph, the text below the graph scrolls to the location of the corresponding changes, if any occurred at the selected patch.

3.5 Website User Testing

Our website underwent informal user testing to determine what parts of the website were unclear, unintuitive, or difficult to use. We selected three friends who currently play or have played League of Legends to do this testing. We did not feel it would be helpful to test with subjects who have not played League of Legends as the information displayed will not have meaning to them. Because of this, these subjects would be unlikely to have the insight needed to think of meaningful interactions they would want to perform with the data.

We performed this testing by allowing the subjects to use the website however they wanted, encouraging them to attempt to utilize all features that they could find. We observed them while they tested and took note of any features they used and how they were used. Once they were done, we received their feedback on the features they used and informed them of the features they did not use. We then also received feedback on how we could make the features they did not use more obvious or clear.

All of the problems encountered by the users were related to the graph of champion rates. Most of the interaction done on the website is done on the graph so it was expected that it would be a source of usability problems. When more than one rate is displayed on the graph, they are distinguished by different color lines. Initially, one of the colors we used for the rates was black. Because many champions have ban rates which are close to 0% and the line color for ban rates was black, it was difficult to tell that the line for the ban rate was not the x-axis. It seemed as though the line was missing, rather than at the bottom of the graph. Because of this, we changed the line colors so that none of them were black.

When using the graph, a user can click on one of the rates in the legend to the right of the graph in order to toggle whether it is displayed. This feature was not apparent to subjects in

testing. In order to fix this, we display only win rates initially on the graph. Because the other two rates in the legend were greyed out due to being disabled, it was clear to the user that they could be enabled somehow.

Because the graph shows changes over a very large amount of patches, we thought it was best to extend the graph horizontally and allow the user to navigate through the patches with a scroll bar. This allowed the user to see meaningful changes within the graph. However, the scroll bar was initially grey and blended in with the background, making it difficult to notice. Because there were no other indicators that the graph continued to the right, subjects that did not notice the scroll bar did not think of the possibility that there was more to see. To fix this, we made the scroll bar cobalt blue in order to contrast the background.

3.6 Summary

Our code implementation that pulled data from the Riot API produced a sample size large enough to represent the player population accurately. Our scripts were able to pull the patch note information for each champion, but had trouble exhaustively categorizing those patch notes. Although it could not categorize all patch notes, the ones that were categorized automatically had a precision of 94.3%. The remaining patch notes were categorized by the team, meaning upkeep would be necessary for maintaining it in the future. This data was stored in a mySQL server. The website was implemented utilizing a javascript library called Highcharts that displayed champion rate information while we displayed the patch note categorizations underneath in HTML. The website was then put through informal usability tests and improved with feedback.

4 Analysis

In order to come to meaningful conclusions about the data we have collected, we have analyzed and graphed the data in the following sections. We analyze the number of champion releases, the rates of champions with regards to how they influence one another, the rates of champions with regards to roles, the distribution of patch note categorizations, and the effect of patches on rates. We also perform an in-depth case study looking at a champion in order to put other analysis into perspective.

4.1 Champion Releases

Figure 11 shows the number of champions released during each season. Early on, many champions were released quickly, usually one or two each patch. More recently Riot began scaling back releases and reworking more old champions that were underplayed and underpowered. Now, one champion is released around every two to three months, which means about five to six releases per season. Figure 11 was created using the patch note data collected during the project.

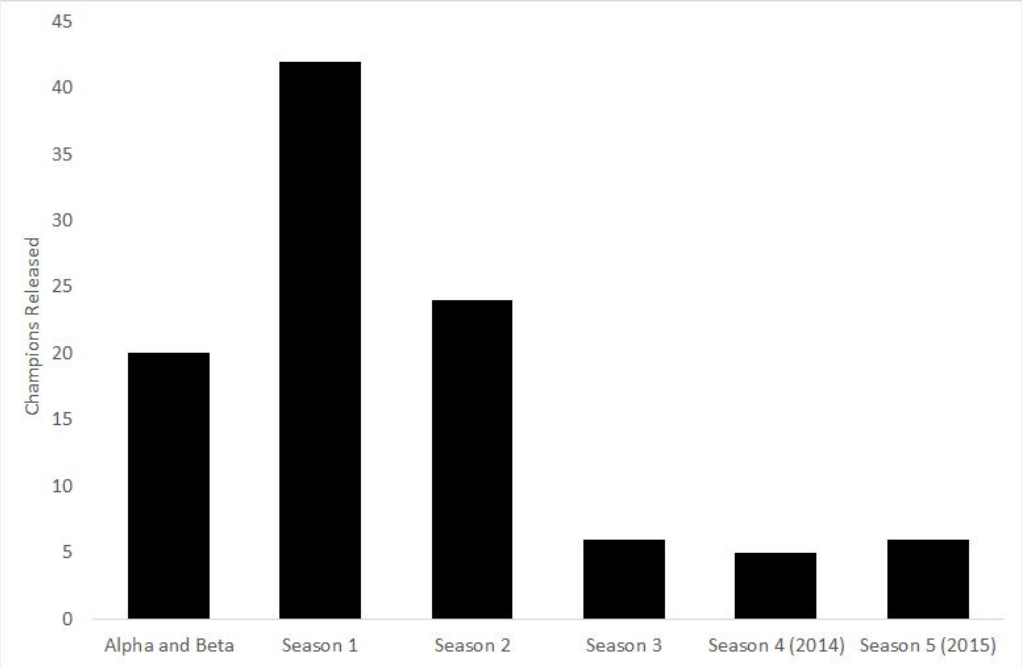

Figure 11: Champion Releases per Season

4.2 Analysis of Rates

The three categories of data we collected for each champion through the Riot API are win rate, pick rate, and ban rate. These three rates are widely considered to be measurements of success, both in the eyes of the playerbase and in those of the game's developers. As a result, the analysis of these rates can also serve as analysis of the success of champions. Win rate typically associates the measurement of success with the overall performance of the champion in terms of strength. Pick and ban rates are also related to champion strength, but are more influenced by the popular opinion of the playerbase.

After sampling enough games to reach a statistically appropriate sample size, we found the data gathered to fairly accurately represent our expected outcomes. Figures 12 and 13

below display the 5 point summary plots (displaying the min, median, max, and quartiles) and cumulative distribution function charts for the 3 rate types.

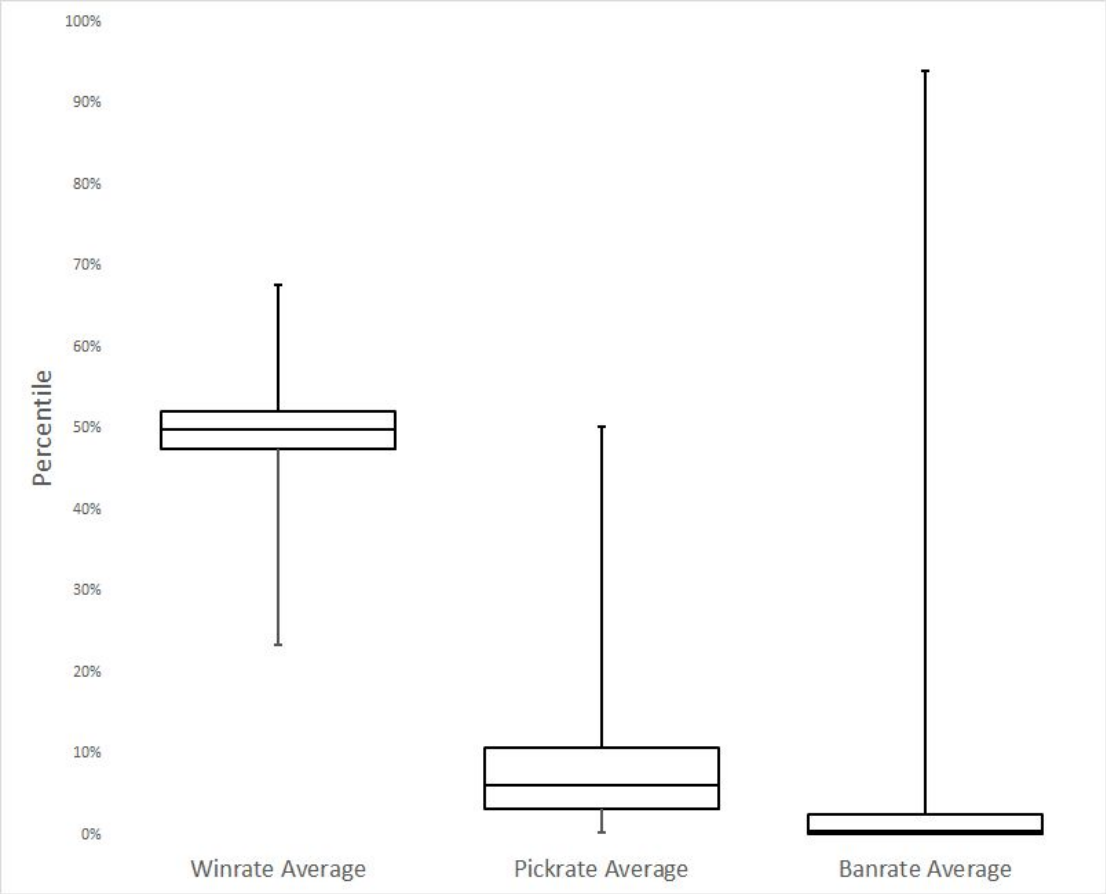

Figure 12: Box Plot for Rates of all Champions

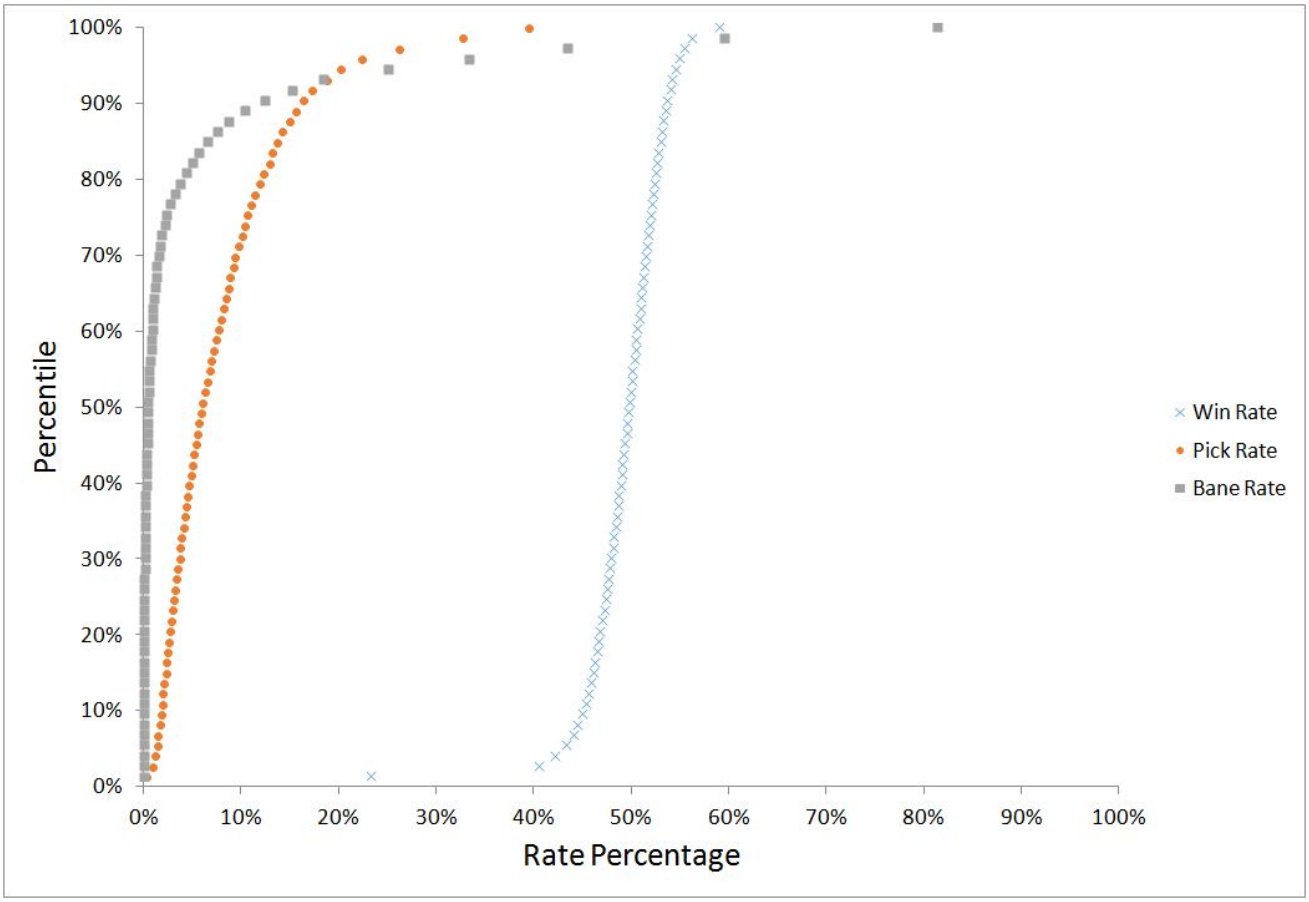

Figure 13: CDF for Rates of all Champions

	Win Rate	Pick Rate	Ban Rate
Min	23.3%	0.3%	0%
Quartile 1	47.5%	3.2%	0.1%
Median	49.8%	6.2%	0.5%
Quartile 3	52%	10.7%	2.5%
Max	67.7%	50.2%	94%
Range (Max - Min)	44.4%	49.9%	94%
IQR (Q3 - Q1)	4.5%	7.5%	2.4%
Mean	50%	7.8%	4.7%

Figure 14: Statistics for Rates

The spread and central tendency are two aspects that we thought would be very distinguishable between each of the three rate types. We calculated the median as a measurement of a central tendency and measured spread as a five point summary, recording the values from 0% to 100% of our sample at 25% increments. We then used the expected statistical average for each of the rates to predict central tendency. Win rates were predicted to have a central tendency of 50% and the lowest spread. Pick rates were predicted to have a central tendency of 7.8% with the next lowest spread. Ban rates were predicted to have a central tendency of 4.7% and the highest spread by far, as seen in Figure 14.

Win rate was predicted to have the lowest spread because of the nature of game balance. Riot Games attempts to keep champion win rates as close to 50%. Any win rate higher than 50% implies that the champion is giving the player an advantage while any win rate lower than 50% implies the champion is giving the player a disadvantage. In order to maintain balance, Riot decides to respectively nerf and buff these champions. Because this is the only rate that has been guided towards a specific, unchanging goal, it was likely to have a fairly small variance centered around its goal. This hypothesis is supported by Figure 15 below. The graph plots the win rate of a champion against the number of changes on a given patch for that champion. It should be noted that the graph only includes champions which had at least one change on that patch, as we are analyzing that each patch changes problem champions and not that problem champions are changed on every patch. We use quadratic regression to model a trendline to the data. The trendline shows that as the win rate moves farther from 50%, the number of changes on that patch for that champion increases. However, the R-squared value is only 0.0082. This means that only 0.82% of the data can be attributed to a relationship between the number of changes in the patch and the win rate. While this makes the correlation weak, the

graph still seems to have a trend which shows that changes made in patches are more often associated with problem champions.

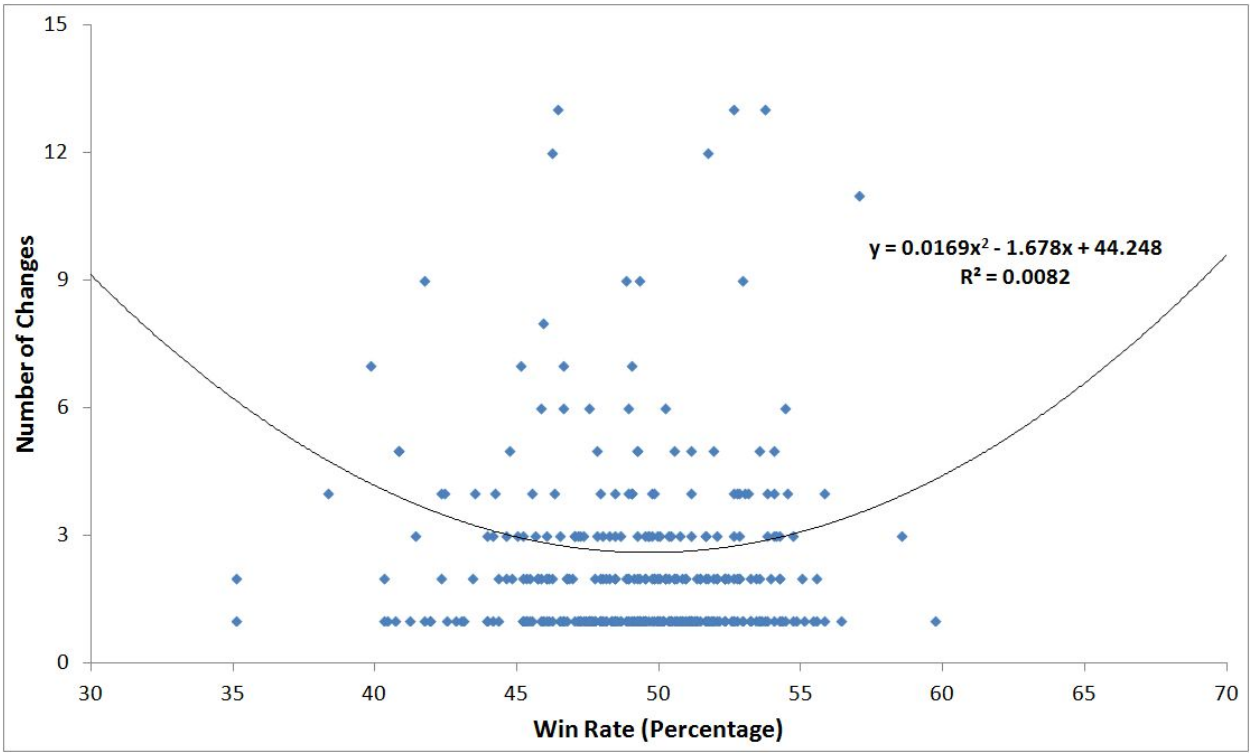

Figure 15: Number of Changes vs Win Rate of each Champion by Patch

Additionally, whenever a match ends, one team must win and one team must lose; so for any given match, 50% of the champions experience a win and 50% experience a loss. However, only 10 champions can be picked in a match and only 6 can be banned. Because there are 128 available champions, the chance of a champion being picked or banned in any given game is naturally lower than the chance they win a game given that they have already been picked. The results gathered show this hypothesis to be supported, with a median of 49.8% and a mean of 50%, as the distribution was very close to normal, with a similar median and mean. A normality test was conducted to show how well the win rate data fit a normal distribution. A normal probability plot, seen in Figure 16 below, was made with the win rates and their corresponding calculated z values, the result was a fairly straight line with a correlation

coefficient of 0.991, meaning the line of best fit was almost perfect. Despite the line looking fairly curved, so many data points were condensed along the line that the outliers towards either end of the curve didn't effect the line much. Win rate also had the lowest range at 44.4% and the second lowest IQR (Interquartile Range) at 5.5%, as seen in Figure 16. Each point represents a champion-patch pairing.

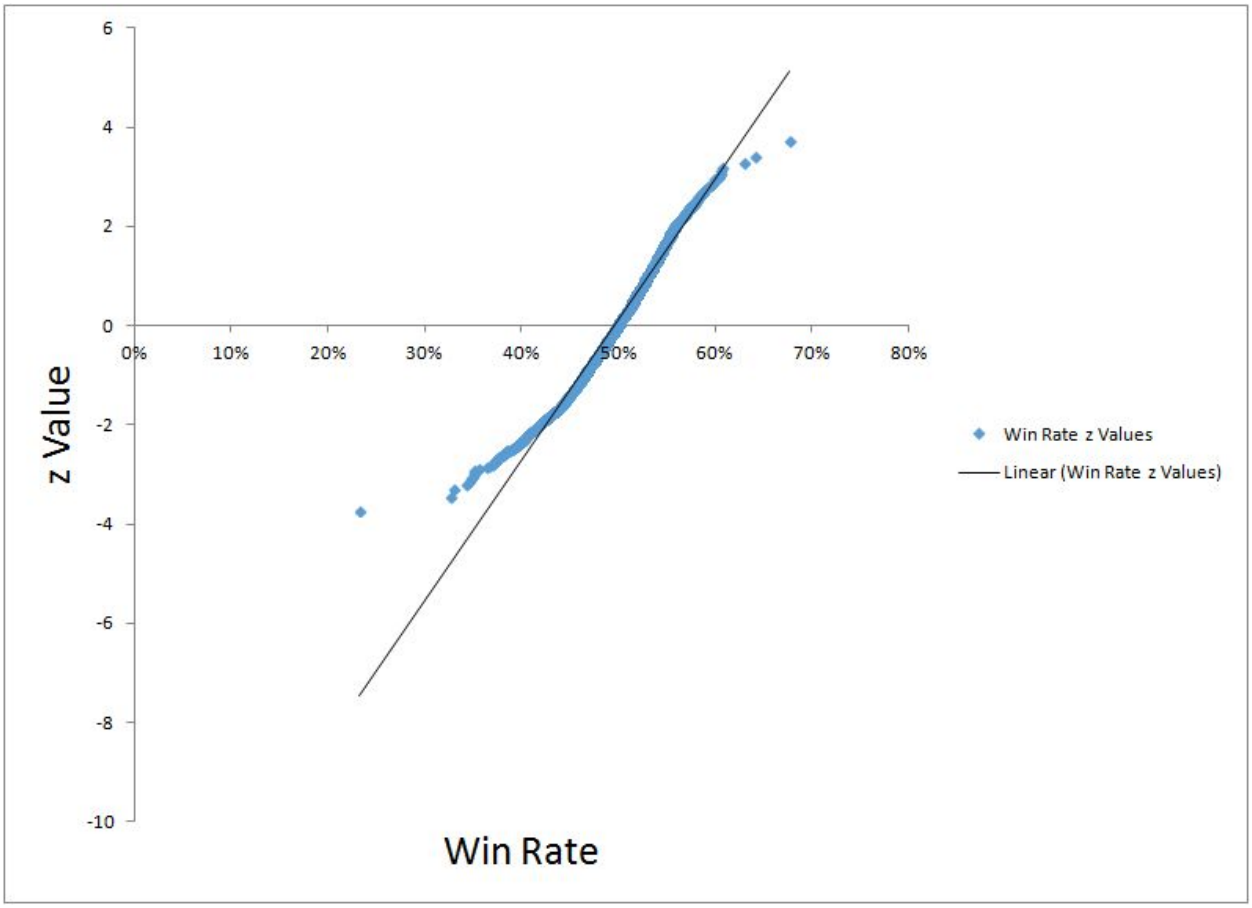

Figure 16: Normal Probability Plot for Win Rate

Pick rate was predicted to have a spread and central tendency between those of win rates and ban rates. It is only possible for 10 champions to be picked in a game. This means, just going off of chance as a basis, each champion has about a 10/128 or 7.8% chance to be

selected per game, a much lower chance than they would have of winning a game they have already been selected for. This is the basis for our prediction of 7.8% as a central tendency for pick rate. We also predicted the spread to be larger than with win rate because pick rate is heavily influenced by public opinion, meaning champions that are perceived as strong will be picked more often than those perceived as weak. This perception is often somewhat disconnected with the actual strength of champions, meaning the best champions are perceived to be better than they are, and the worst champions are perceived to be worse than they are. This notion can be seen the comparison of win rates vs pick rates in Figure 17 below. We thought this could lead to a larger spread, as players would pick the more popular champions in number disproportionate to how effective they actually are. The results gathered showed our central tendency prediction of 7.8% to be a small overestimate, measuring 6.2% for the median, which makes sense as the pick rates were not as normally distributed as the win rates. Mean would be an appropriate measurement of central tendency if the distribution was normal, but because the data is skewed, a median is more appropriate. The mean was calculated to be 7.8%, exactly representing the expected value. The spread was measured as 49.9% for range and 7.5% for IQR, both larger than the win rate spreads, as seen in Figure 14.

Ban rate was predicted to have the lowest central tendency and the highest spread as very few champions per game are banned. In every game, only 6 champions are banned, meaning each champion has a 6/128 or 4.7% chance of being banned. This is what led our hypothesis of central tendency to 4.7%. Similarly to champions picked, the champions banned are generally the ones perceived to be the strongest or the ones that many players personally find frustrating. This leads to the majority of the champions which are considered average or below to almost never be banned, while a select few champions are almost always banned. This skews the data considerably, making the median significantly lower than the expected

mean. The results gathered support this notion over our predicted value, as 0.5% was calculated for the median; far off from our mean-supported prediction of 4.7%. The calculated mean was 4.7%, exactly representing the expected value. This skew in the data also shows in the spread. Ban rates had the largest range, at 94%. However, they had the smallest IQR, at 2.4% as seen in Figure 14. This shows that on a large scale that includes the extreme outliers, ban rate has the widest spread, but when ignoring most of the extreme values its spread is very small because most champions see few or no bans.

Another interesting analysis of the rates is to observe whether they have an effect on one another. Since players play in order to win, players generally pick a champion that would help them win the most. Are the champions that get picked the most actually the champions that win the most? Are bans actually influencing the champions that are picked? Below are some graphs used to determine whether the rates are influencing one another in some way. All of the graphs compare rates such that each dot represents a pair of rates for a specific champion on a specific patch.

In Figure 17, we can see a graph comparing win rates and pick rates. While the graph does seem to be trending upwards, the R-squared value is very low at 0.0155. This means that only 1.55% of all the data in the graph can be attributed to a relationship between win rate and pick rate and did not happen by pure chance. This does not support a relationship between the data and implies that there are other factors which influence the rates.

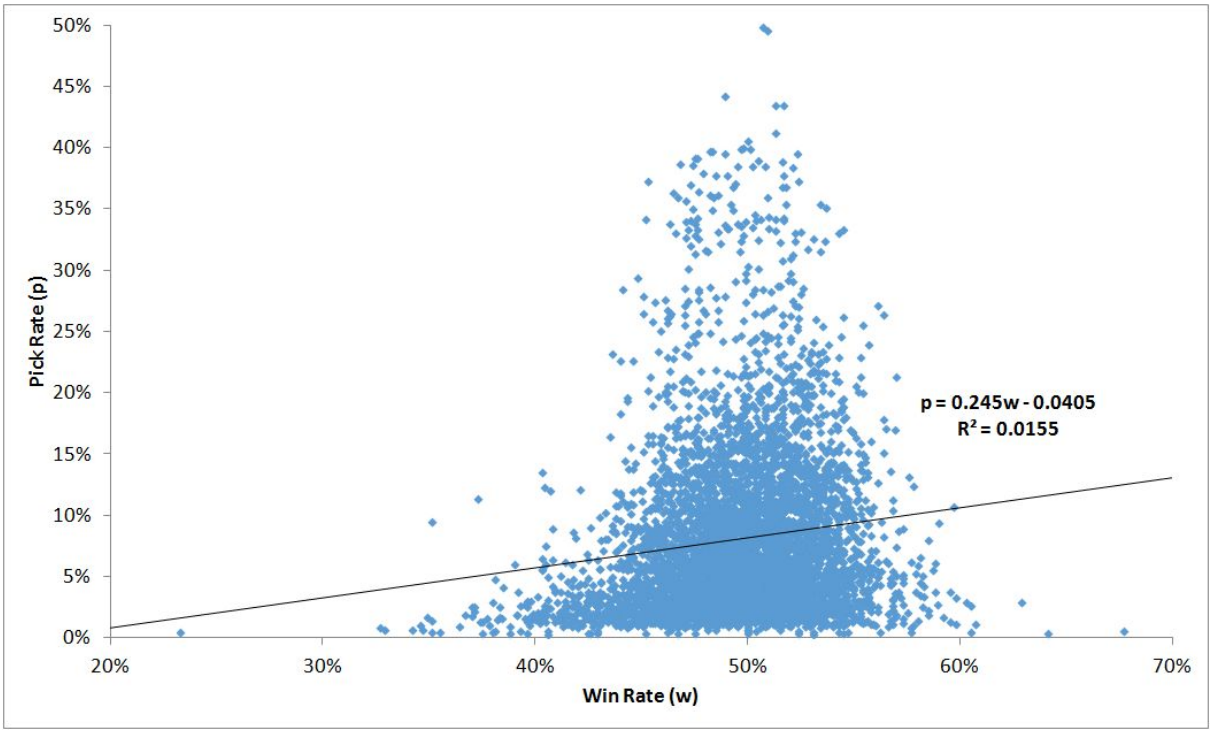

Figure 17: Win Rate vs Pick Rate

In Figure 18, we can see a graph comparing win rates and ban rates. This graph seems to have even weaker support of a relationship with an R-squared value of only 0.0023. This means that only 0.23% of all the data in the graph can be attributed to a relationship between win rate and pick rate. Similarly to the previous graph and despite the r-squared value, the trendline still seems to show a small positive correlation.

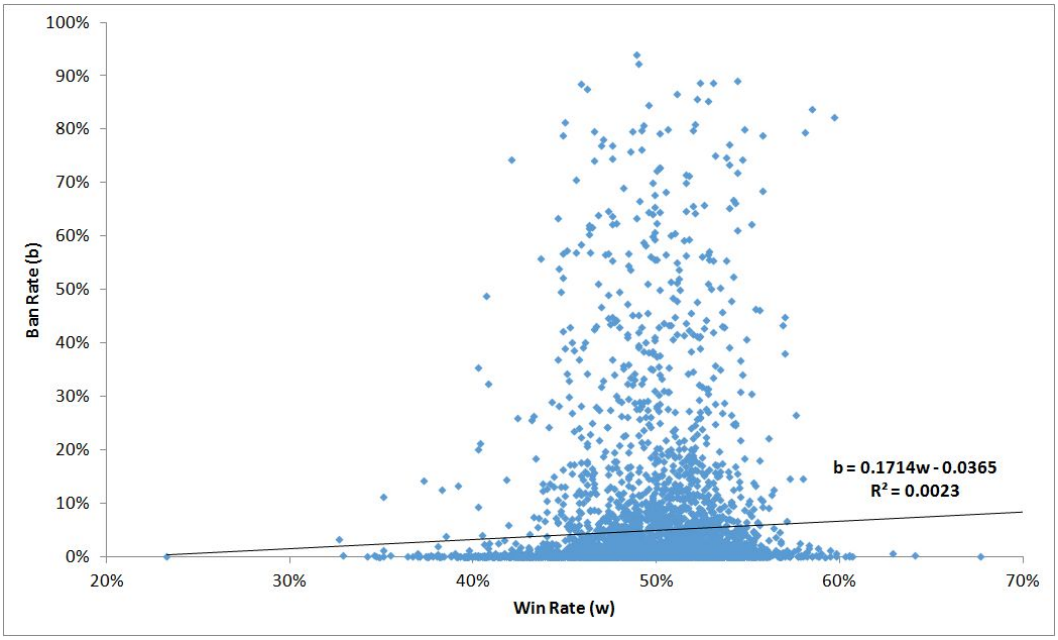

Figure 18: Win Rate vs Ban Rate

In Figure 19, we can see a graph comparing pick rates and ban rates. This is expected to have the largest correlation because picking and banning are disjoint, unlike the other combinations of rates. If a champion is banned, that champion cannot be picked. Likewise, if a champion is picked, that champion was not banned. Because of this, it would be expected that ban rate and pick rate would have a negative correlation. However, the trendline seems to imply a weak positive correlation with an R-squared value of 0.0598. While the points themselves seem to imply a negative correlation, it seems that the large density of champions which are uncommonly picked and banned is affecting the trendline significantly, causing it to appear more like a positive correlation. We attempt to mitigate this effect by removing any points which have ban rates under 10% in Figure 20. Because of the removal of these points, the trendline now suggests a negative correlation. However, it this line still has a weak R-squared value of 0.0652.

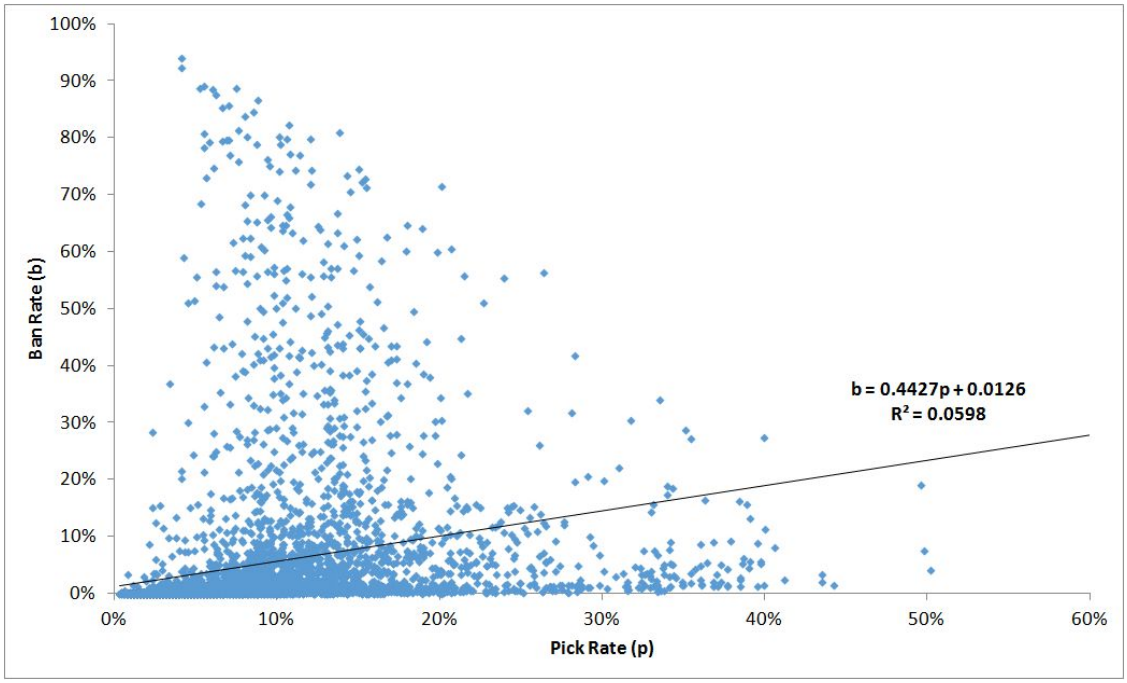

Figure 19: Pick Rate vs Ban Rate

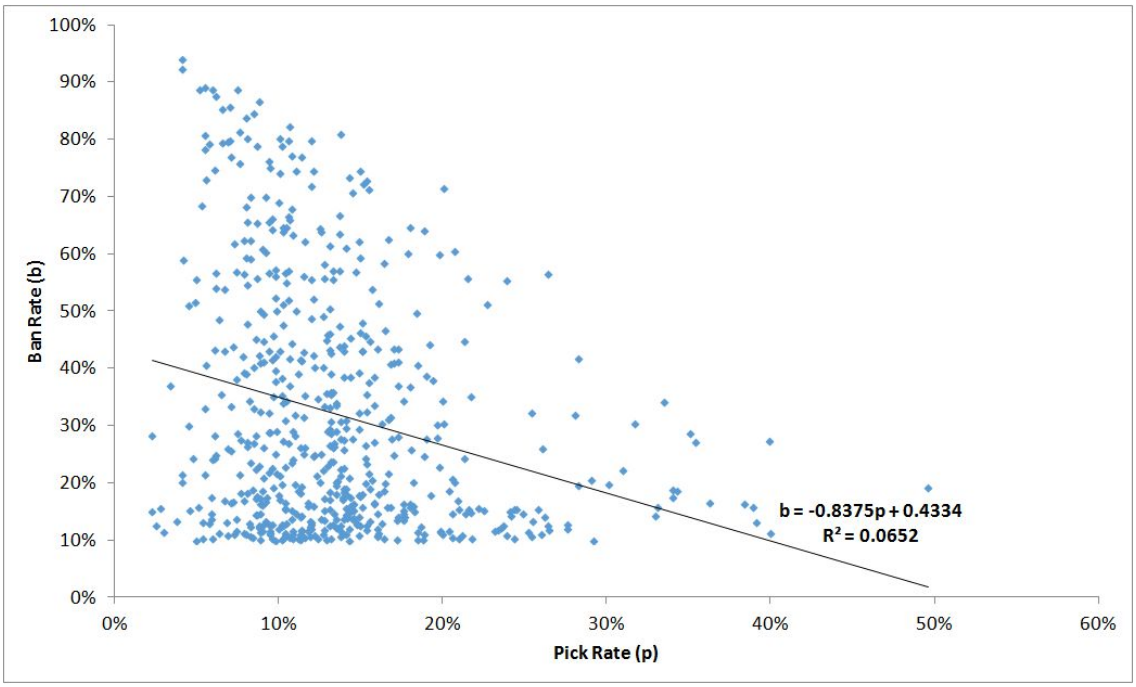

Figure 20: Pick Rate vs Ban Rate (>10% ban rate only)

This high density of points below 10% ban rate is likely due to the strength and popularity of champions. Top champions are often the ones being picked and banned while unpopular champions are almost never picked or banned. This is because a team's goal is victory and top champions on one's team help achieve victory while top champion's on the opposing team hurt victory. Additionally, since there are only three bans per team, not all the top champions can be banned. This leaves room for them to still be picked.

4.3 Analysis of Rates with Respect to Champion Roles

There are six roles that Riot assigns to their champions to characterize the function they are supposed to fulfill: marksman, mage, support, tank, fighter, and assassin. In addition to observing the rates of all champions as a whole, much can be learned from viewing the rate data stratified by champion role. The roles are very distinct in playstyles and objectives. Marksmen are designed to deal consistent damage with basic attacks from a safe range. Mages are designed to deal high damage with their abilities in short bursts from a safe range. Supports are designed to aid the damage dealers by buffing them or debuffing the enemy. Tanks are designed to take a lot of damage and protect their damage dealers. Fighters are designed to be durable and deal decent damage over time in close range. Assassins are designed to have high damaging, close range abilities, but poor defense. The rate data for the six roles differs, reflecting the contrast of characteristics.

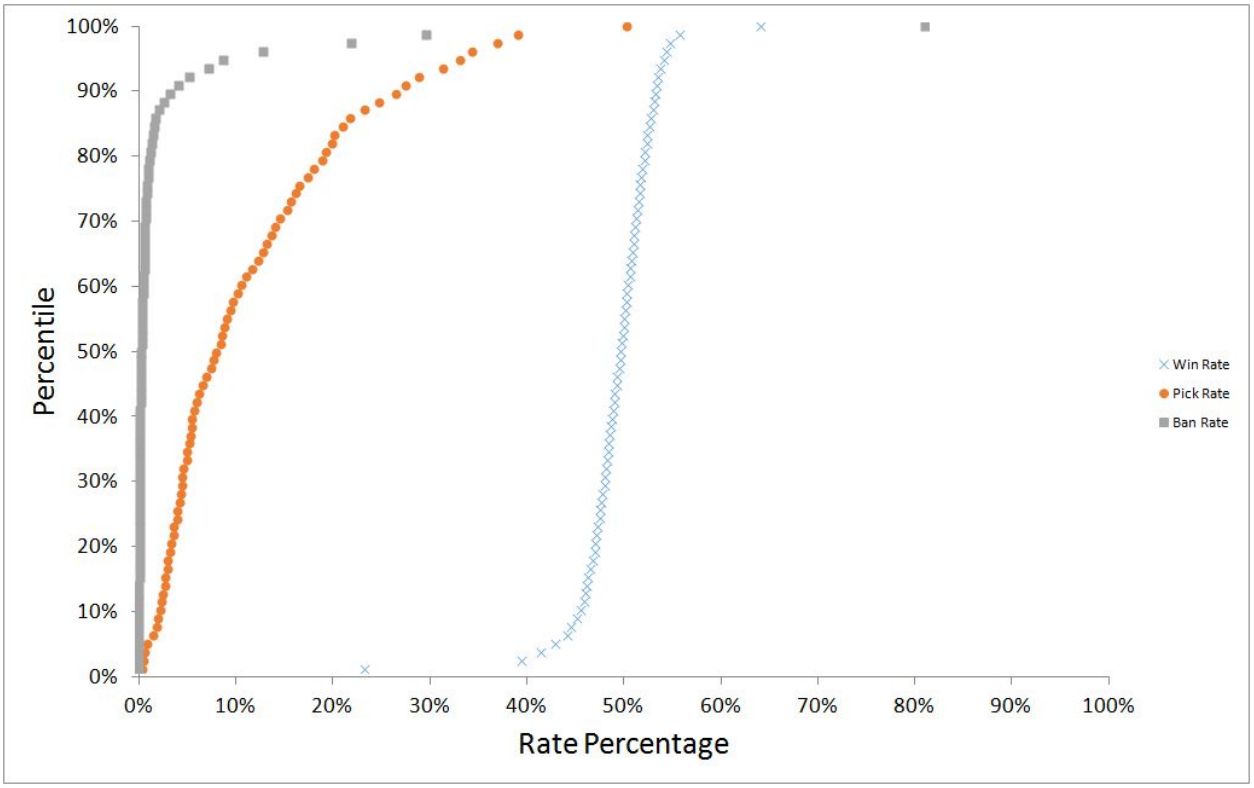

Figure 21: CDF for Rates of Marksmen

For marksmen, we found them to have a lower ban rate, higher pick rate, and had an average win rate in comparison to the other roles, as seen in Figure 21 above. The marksman role is unique among the other six in that it is considered the only “necessary” role for an effective team. Certain combinations of champion roles are seen as more beneficial than others. A team might not need a mage and replace him/her with an assassin. The team might choose to add a second mage instead of a support. The team might double up on fighters instead of having a tank. However, the one constant in almost all team compositions is the necessity for a marksman. The reason for this is simply that marksmen become highest source of consistent damage by the end of the game. Other roles may outperform them at different points earlier in the game, but a marksman is an investment most teams must have. If the game runs too long

and a team decided not to pick a marksman, their chances of winning drop significantly against a team that did.

The pick rates of the other roles, shown in Figures 23, 25, 27, 29, and 31 further down, typically see a maximum between 30% to 40%. However, marksmen pick rate is noticeably higher, reaching up to 50%, showing they are the most represented role. This result supports the notion that marksman are considered the most necessary role. Marksmen also are considered to have less variety between them than other roles in terms of abilities and characteristics. This leads to fewer bans, as only one or two significantly strong champions tend to stand out from the rest on a given patch. Win rate was found to be a fairly standard range centered at 50% and normally distributed, which makes sense as there will almost always be a marksman on each team, so for every game one should win and one should lose. The normal probability plot of the win rates and their corresponding z values was created to show how well the data fit a normal distribution, seen in Figure 22 below. The correlation coefficient for the line of best fit was 0.966, showing a very strong linear relationship, which in turn implies the original data is a good fit for a normal distribution. Each point represents a champion-patch pairing.

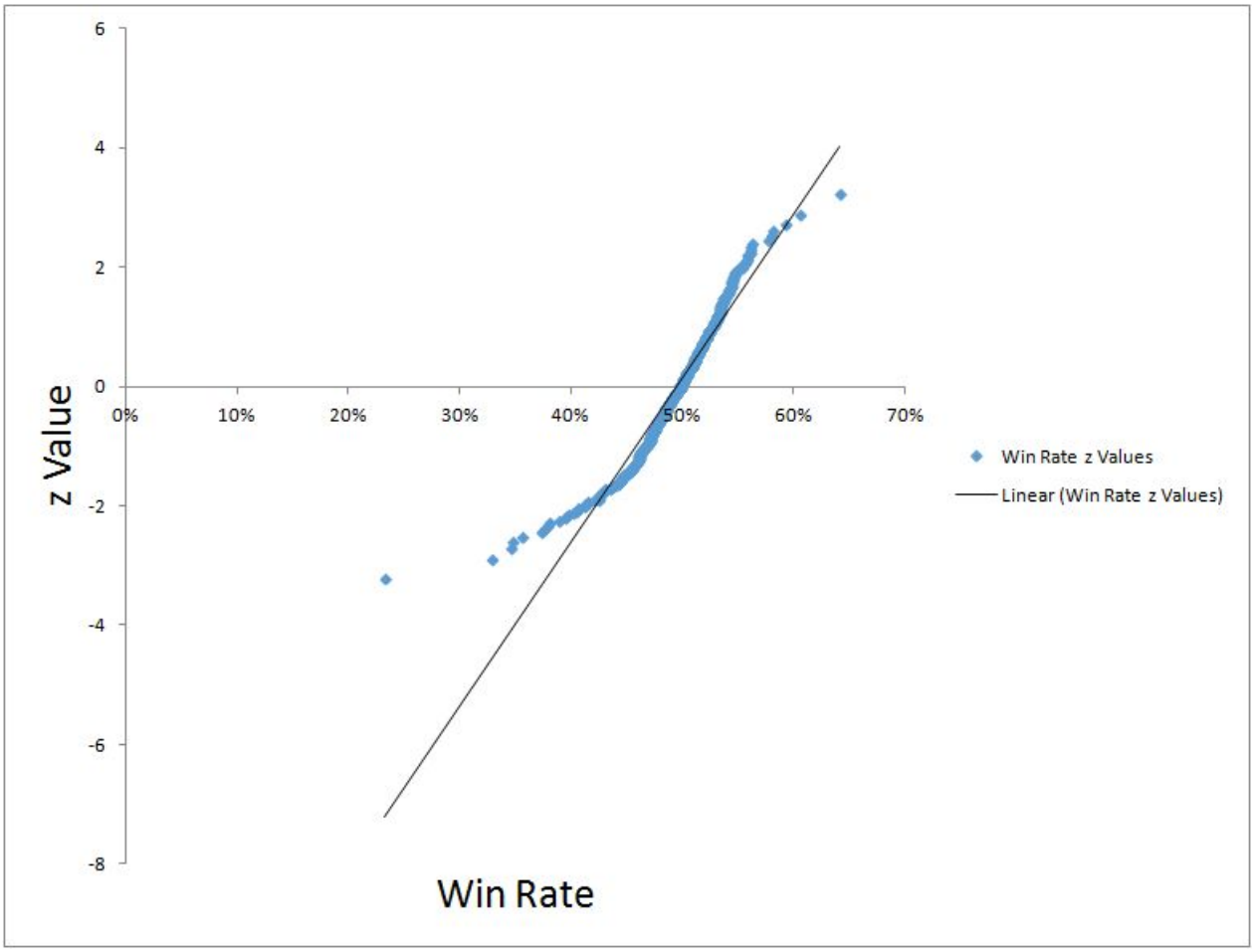

Figure 22: Normal Probability Plot for Marksman Win Rate

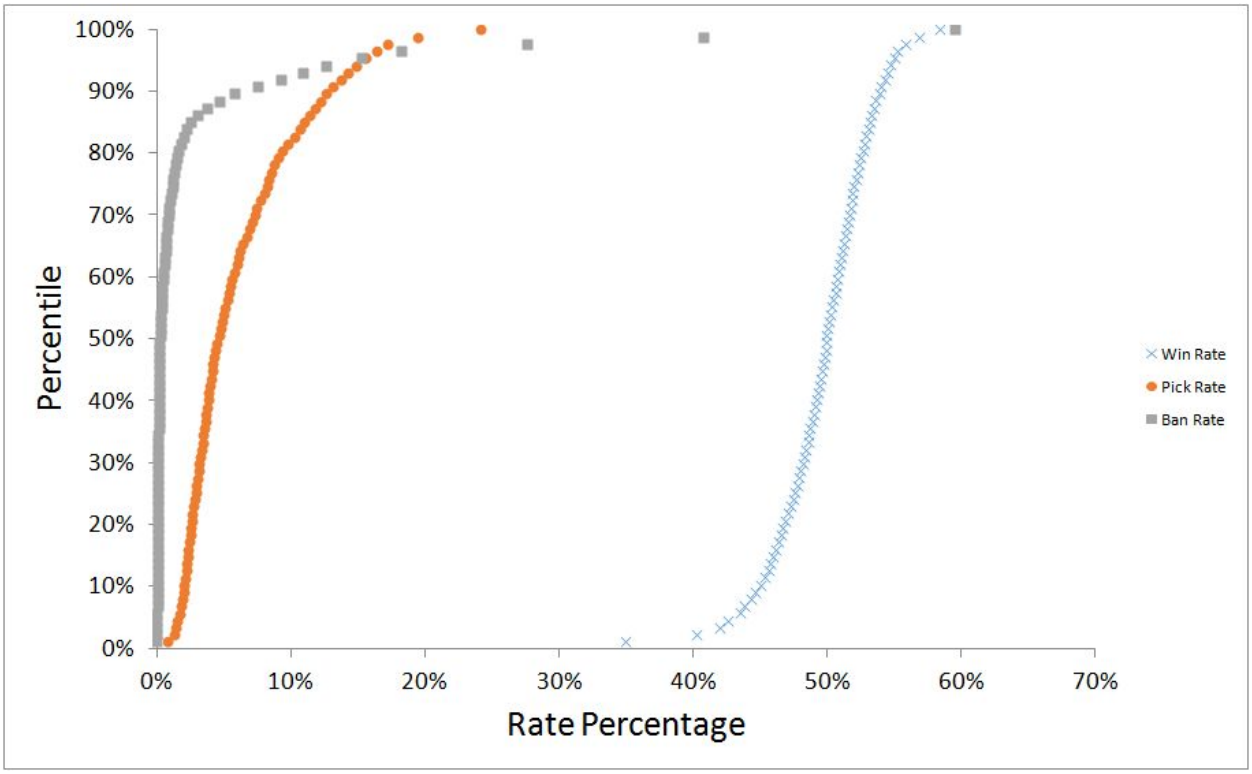

Figure 23: CDF for Rates of Mages

Mages were fairly standard across the board in comparison to the other roles, as seen in Figure 23 above. They had average pick rates, ban rates, and win rates. Mages are frequently picked, but it is not uncommon for team compositions to replace them with one of the other roles, so their pick rate is centered at around 7% with maximums at 35%. Another contributing factor is that mages have the largest number of champions fitting the role. So for every mage that was picked for a game, a larger proportion was being left out than for all other roles. Ban rates are average, at just slightly more frequent than those of marksmen. Win rates also seem somewhat normally distributed and centered around 50%. The normal probability plot of the win rates and their corresponding z values was created to show how well the data fit a normal distribution, seen in Figure 24 below. The correlation coefficient for the line of best fit was 0.993,

showing a very strong linear relationship, which in turn implies the original data is a good fit for a normal distribution. Each point represents a champion-patch pairing.

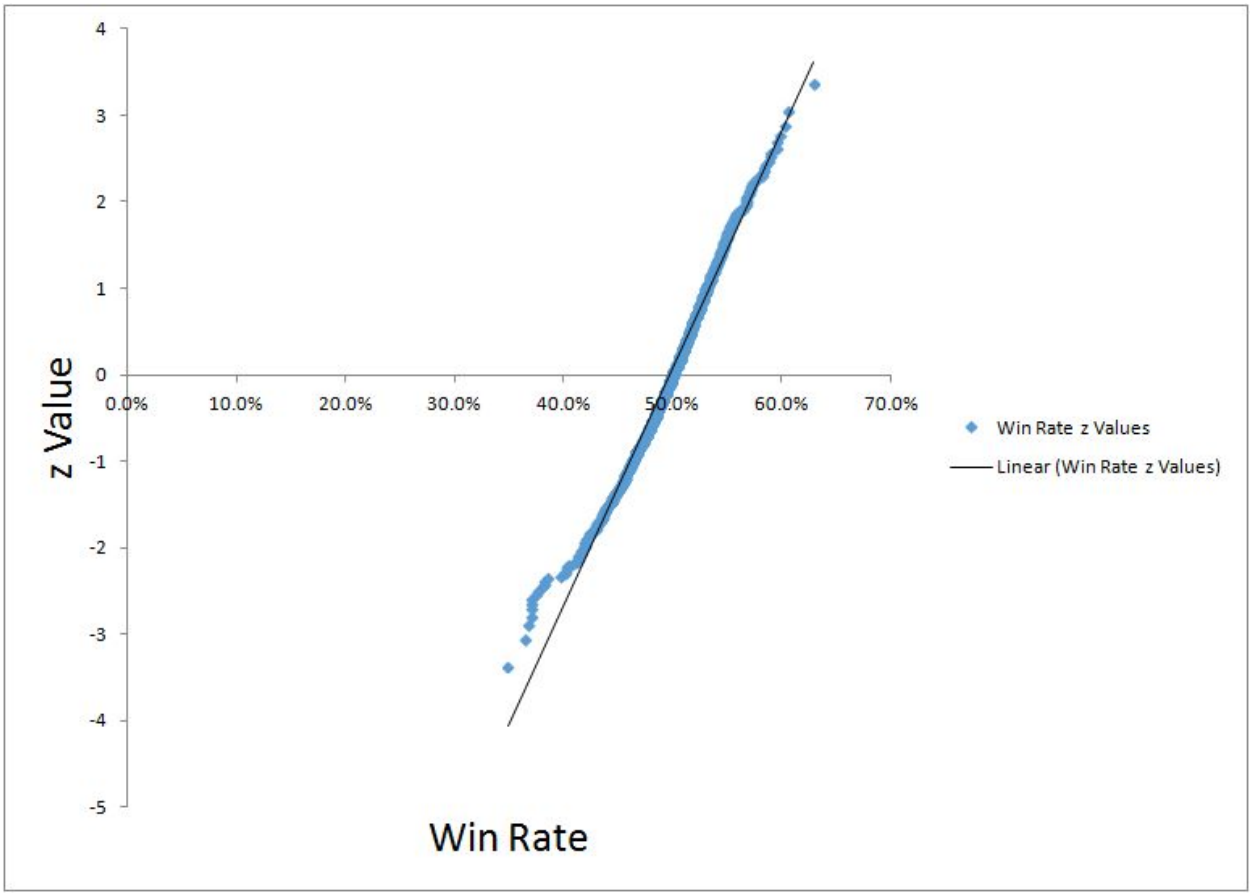

Figure 24: Normal Probability Plot for Mage Win Rate

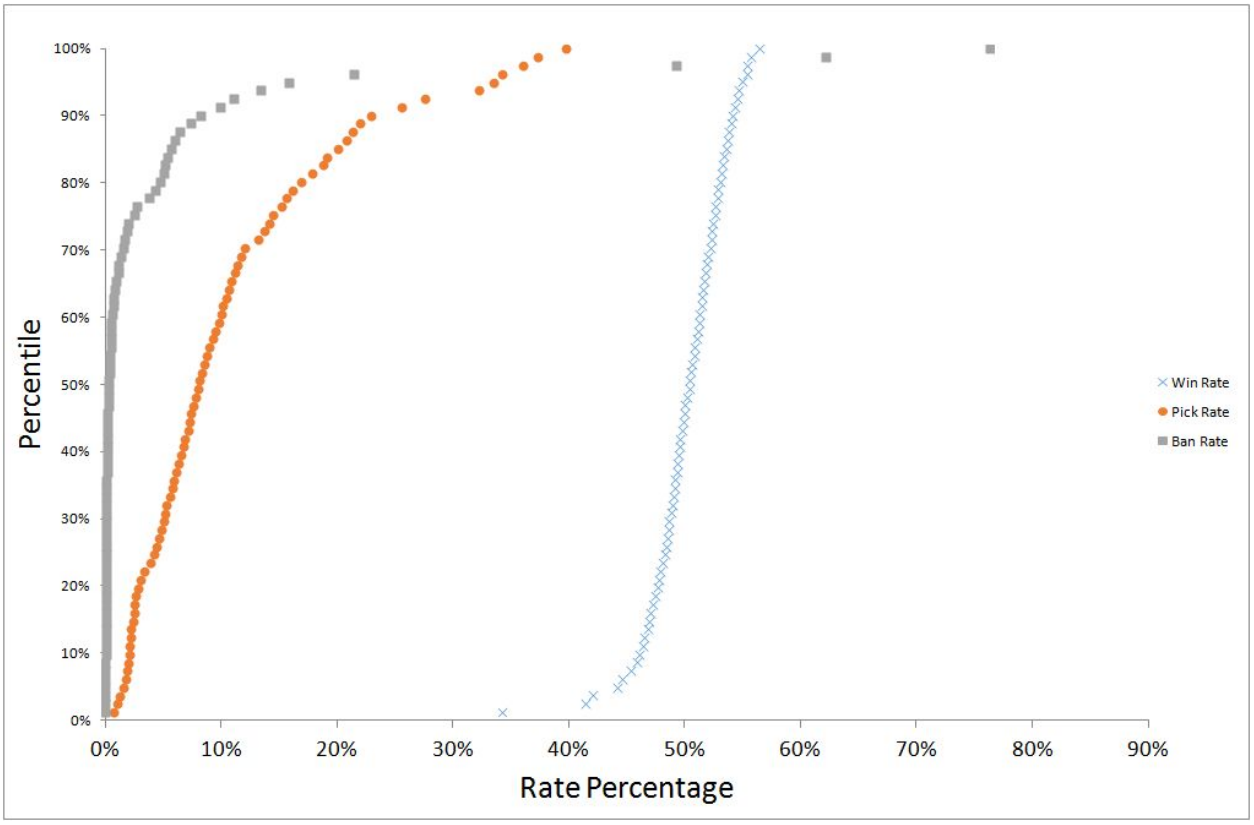

Figure 25: CDF for Rates of Supports

Supports were found to have above average pick rates and ban rates and average win rates as seen in Figure 25 above. After marksmen, supports are one of the most often roles to be picked, as their objective is to protect the marksman, and are often seen as a pair. This trend supports why their pick rate is so high, reaching past 40% for some champions. Their ban rates were among the highest 3 roles. In addition to containing some of the more frequently banned champions, the support role also contains the fewest champions of all roles. This would explain a high pick and ban rate, as any time a single support is picked or banned, it is proportionally more significant than for any other role. The win rate distribution was once again average and seemed normally distributed. The normal probability plot of the win rates and their corresponding z values was created to show how well the data fit a normal distribution, seen in

Figure 26 below. The correlation coefficient for the line of best fit was 0.984, showing a very strong linear relationship, which in turn implies the original data is a good fit for a normal distribution. Each point represents a champion-patch pairing. Each point represents a champion-patch pairing.

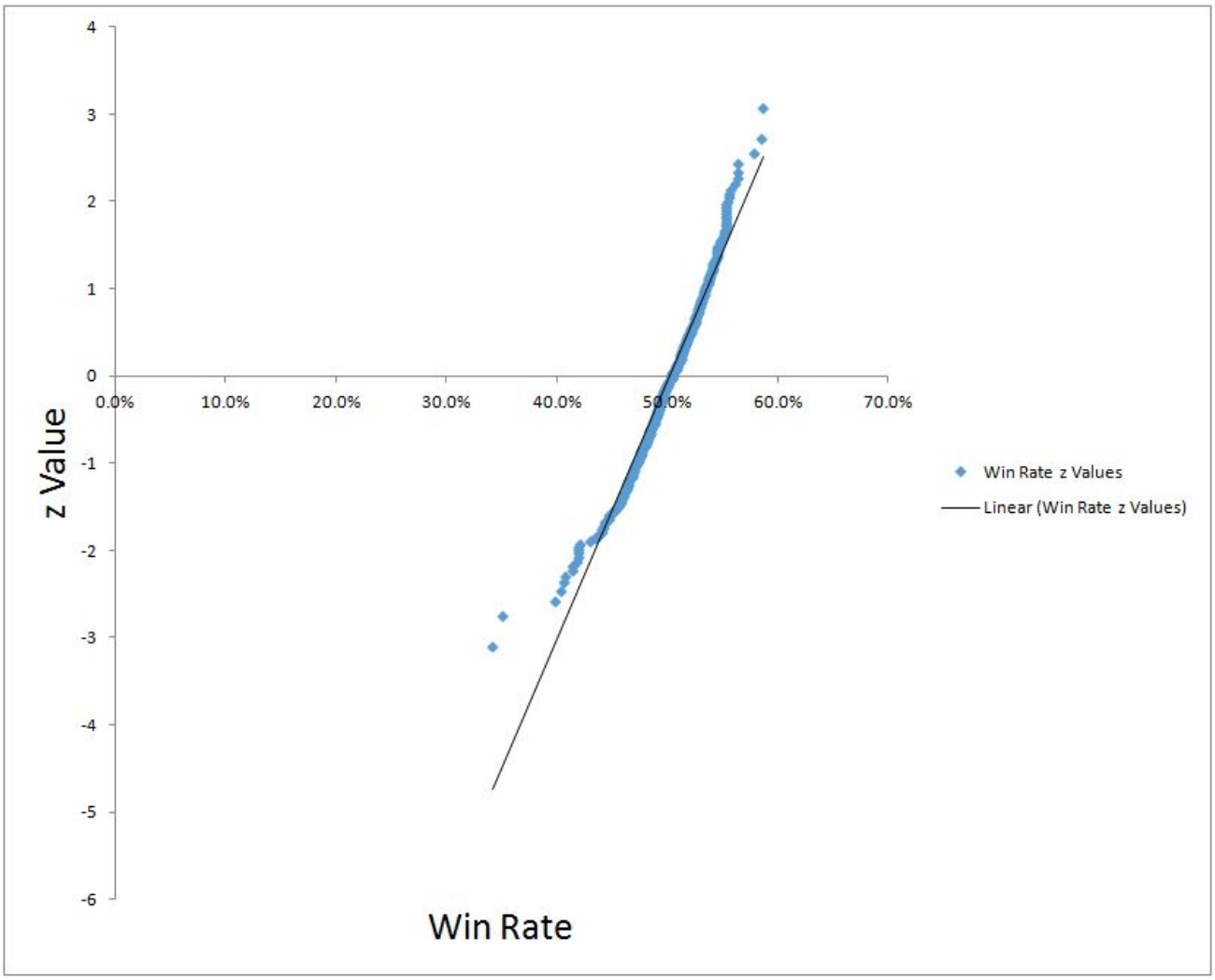

Figure 26: Normal Probability Plot for Support Win Rate

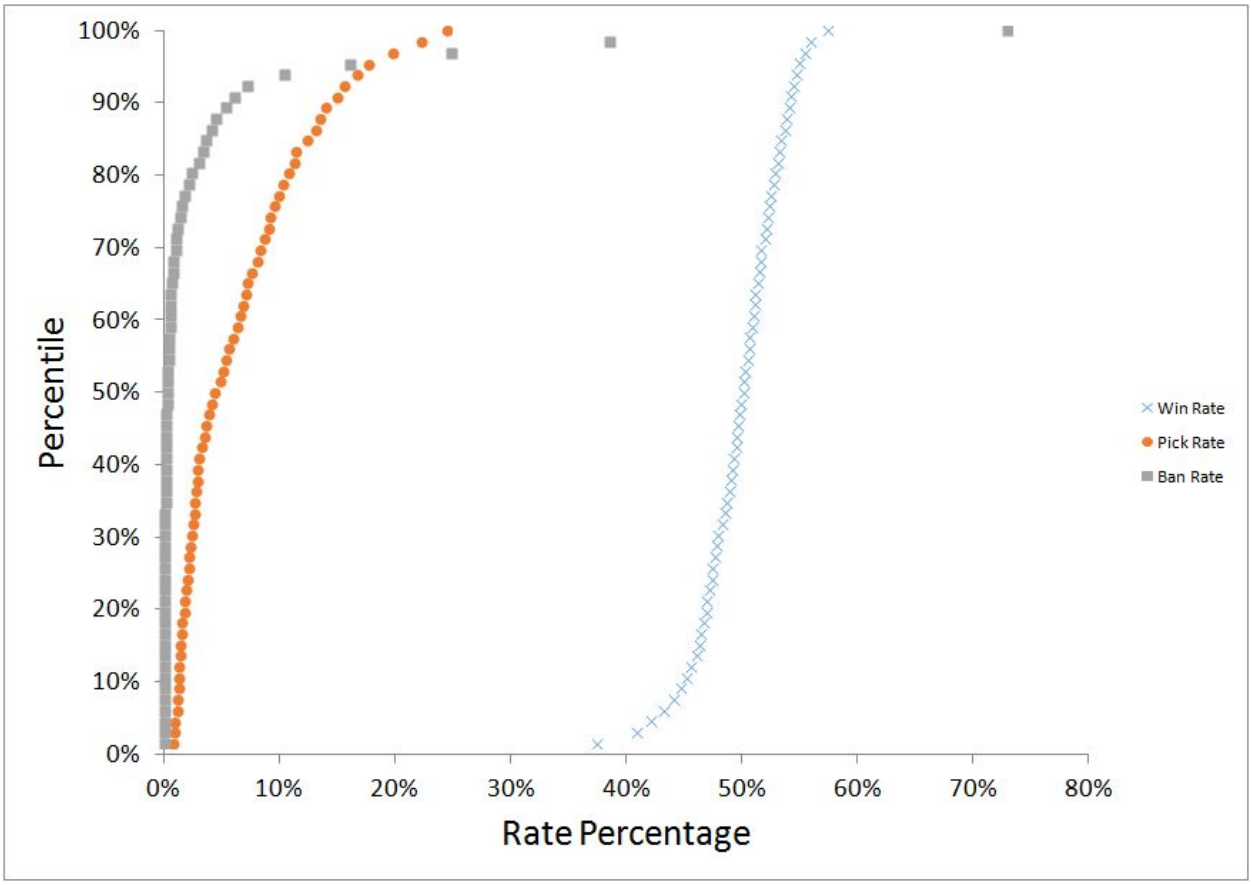

Figure 27: CDF for Rates of Tanks

Tanks were average in win rate and ban rate but had one of lower pick rates, as seen in Figure 27 above. Tanks take a lot of damage, and deal little in return, as their main objective is to keep their damage dealers safe. Something that might affect pick rate in this case is how fun the general public finds the playstyle. If most people enjoy fighting and killing their opponents, tanks might be seen as boring and unappealing. Despite the role being just as effective as the others in terms of win rate, the pick rate is found to be the lowest, not even reaching 30% at a maximum, likely due to popularity of playstyle. The tank win rate was normally distributed similarly to the other roles. The normal probability plot of the win rates and their corresponding z values was created to show how well the data fit a normal distribution, seen in Figure 28 below.

The correlation coefficient for the line of best fit was 0.994, showing a very strong linear relationship, which in turn implies the original data is a good fit for a normal distribution. Each point represents a champion-patch pairing.

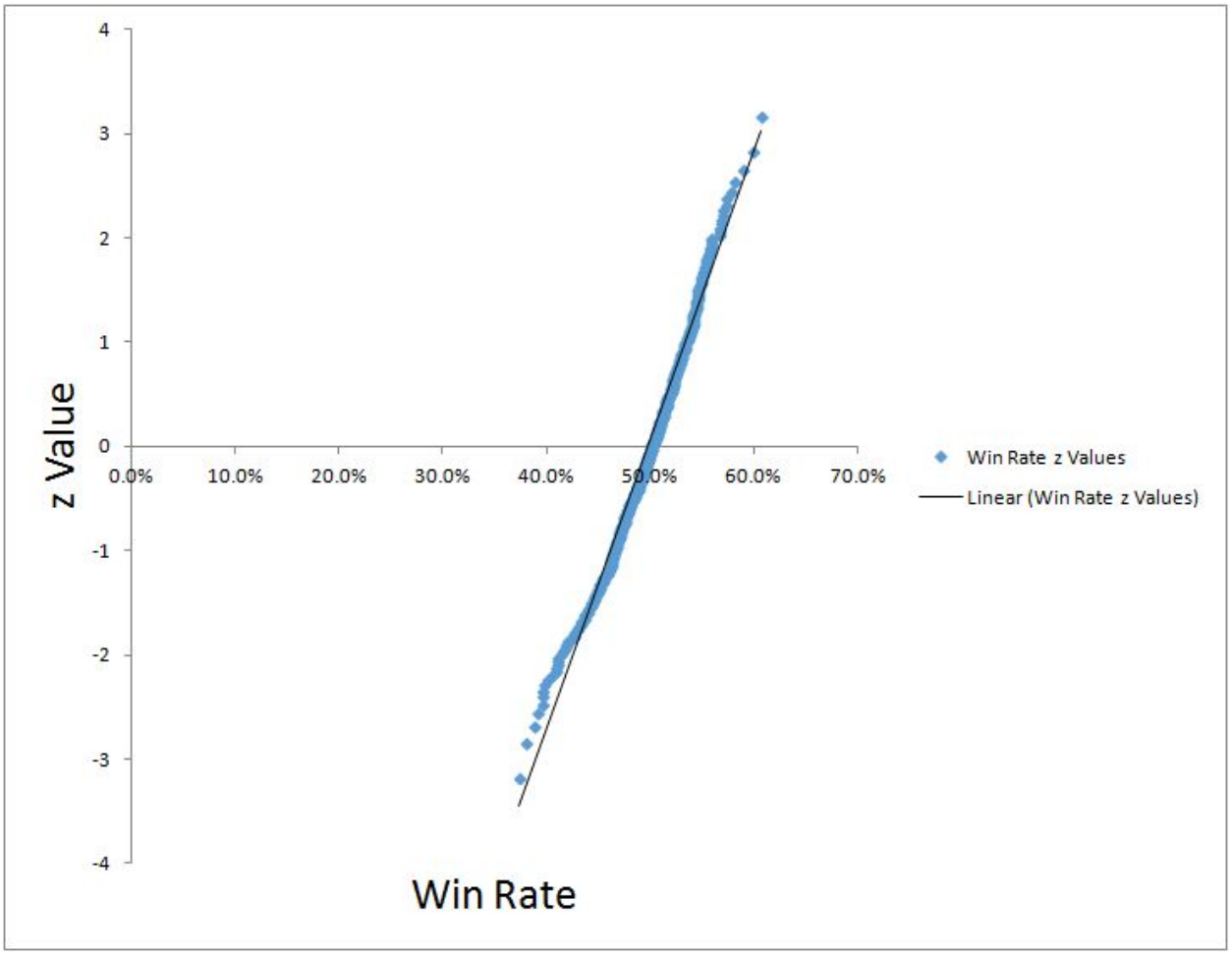

Figure 28: Normal Probability Plot for Tank Win Rate

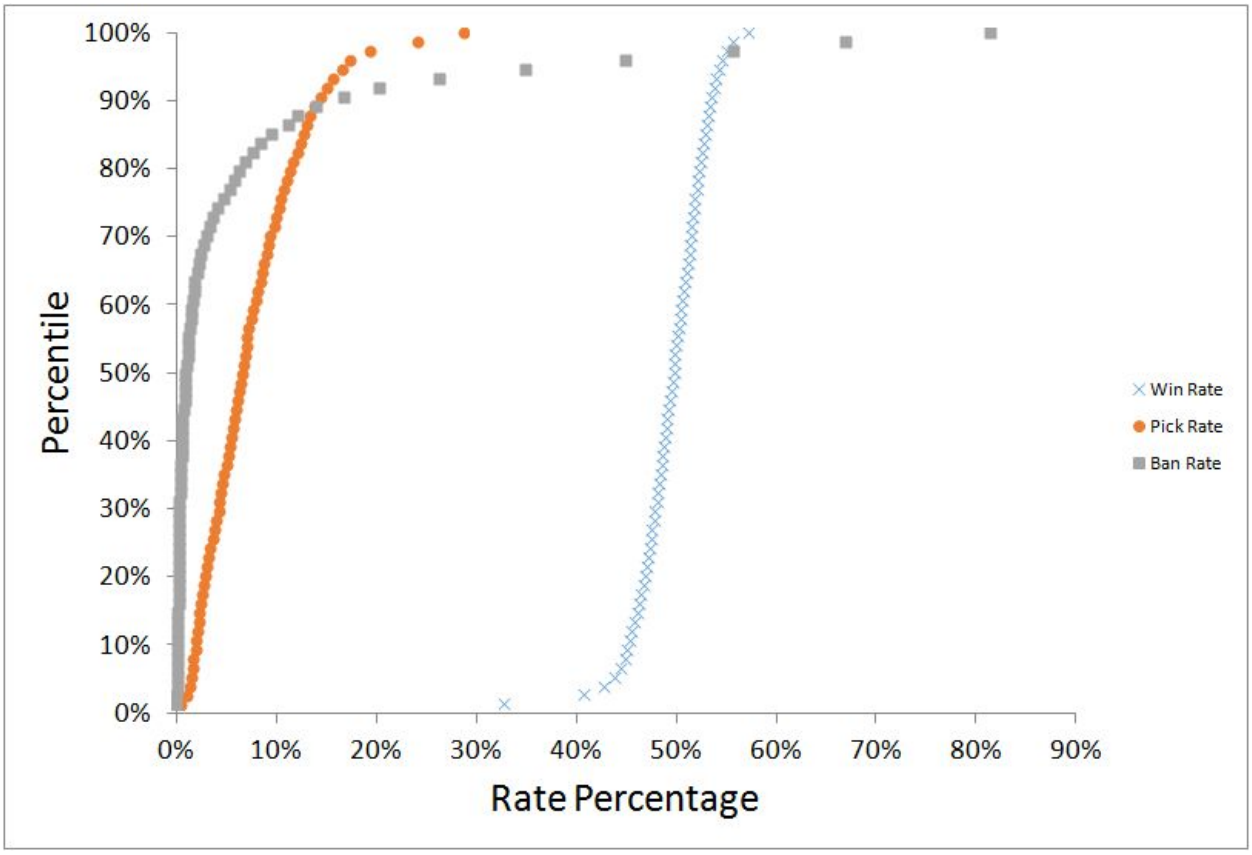

Figure 29: CDF for Rates of Fighters

Fighters were found to have an above average pick rate, and an above average ban rate, and a slightly above average win rate, as seen in Figure 29 above. It is impressive to see pick and ban rates so high in this role, as fighters contain the second largest number of champions out of all roles, meaning a single pick or ban is proportionally much less significant in comparison to most of the other roles. This means popular opinion of fighters must be very high or their playstyles must be seen as very enjoyable. Unlike the other roles, which have near identical win rate cdf curves, fighters are just slightly above average in win rate. Their overall average is 53%, have many points reaching close to 60%, and have a few approaching 70%. With even just a slightly distinguishable difference in win rate, it follows that the fighter role would be the most popular. The normal probability plot of the win rates and their corresponding

z values was created to show how well the data fit a normal distribution, seen in Figure 30 below. The correlation coefficient for the line of best fit was 0.992, showing a very strong linear relationship, which in turn implies the original data is a good fit for a normal distribution. Each point represents a champion-patch pairing.

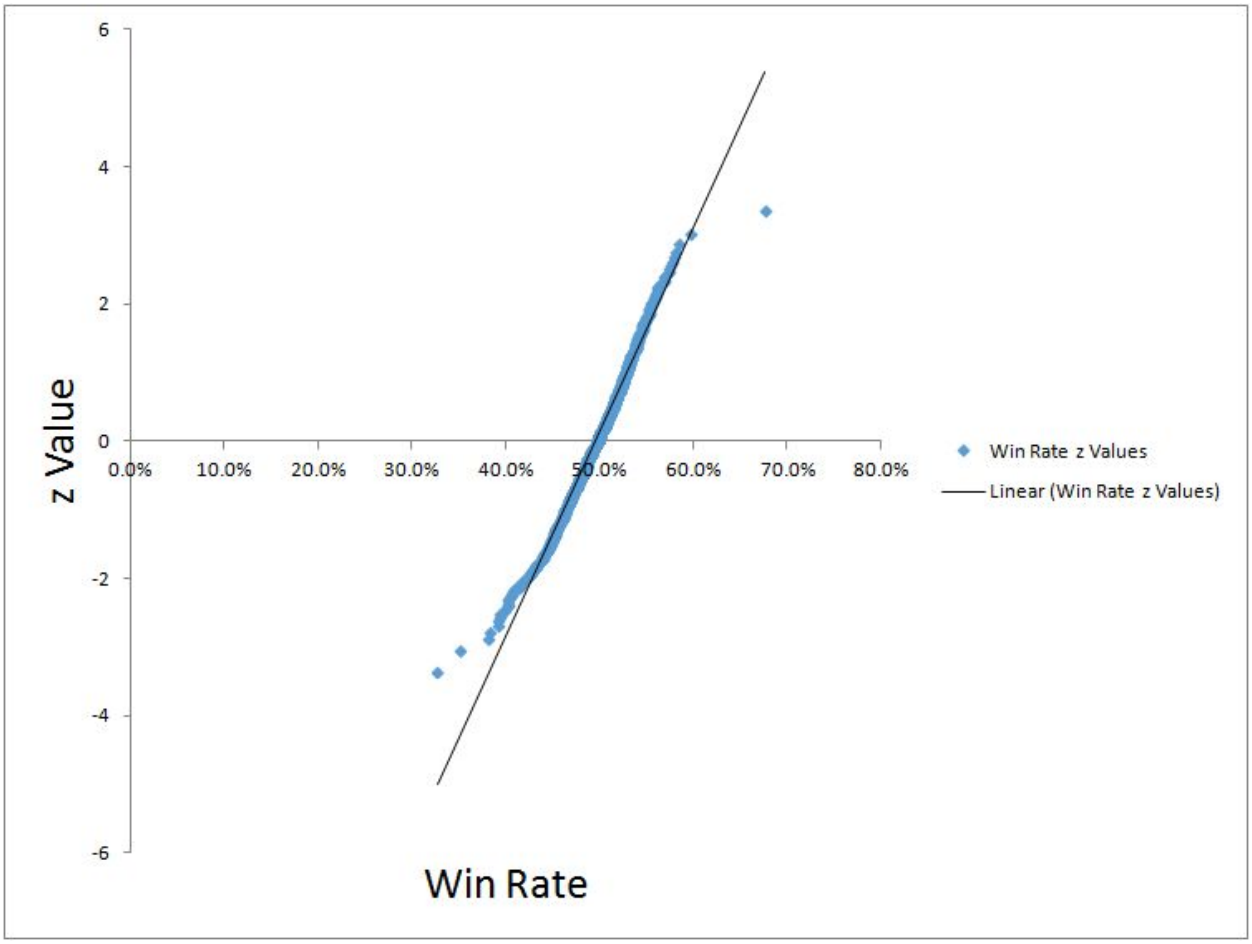

Figure 30: Normal Probability Plot for Fighter Win Rate

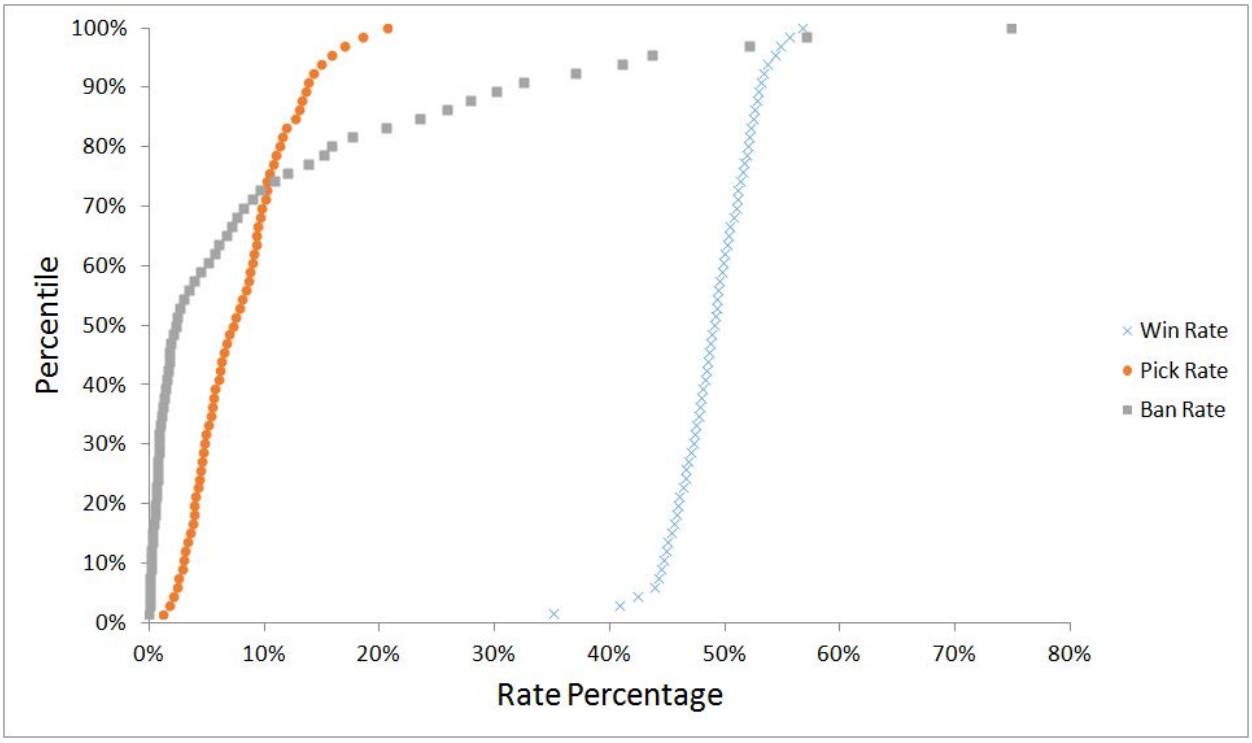

Figure 31: CDF for Rates of Assassins

Assassins were found to have a below average pick rate and win rate, but the highest ban rates of all roles, as seen in Figure 31 above. These results are odd because pick rate and ban rate have been seen as measurements of popular opinion, and generally are not inversely related. However, in this case, the assassin role sees very little play, with a maximum pick rate of 30% despite having incredibly high ban rates. The assassin role contains the second fewest number of champions after supports which would help explain such a high ban rate, as each ban would be proportionally more significant than most roles. However, the same should be true of pick rates, inflating that statistic as well, yet it is below average. This disproportionate ban rate could be the product of public opinion being against the assassin playstyle. Similarly to how tanks might be picked less because players do not want to play as them, assassins might be banned more because players do not want to play against them. If players found playing against

assassins frustrating, ban rate would likely increase regardless of the actual effectiveness of assassins. The slightly below average win rates support this possibility, as they would be higher if they were causally connected to such high ban rates. The normal probability plot of the win rates and their corresponding z values was created to show how well the data fit a normal distribution, seen in Figure 32 below. The correlation coefficient for the line of best fit was 0.9999, showing by far the strongest linear relationship of all roles, which in turn implies the original data is a good fit for a normal distribution. Each point represents a champion-patch pairing.

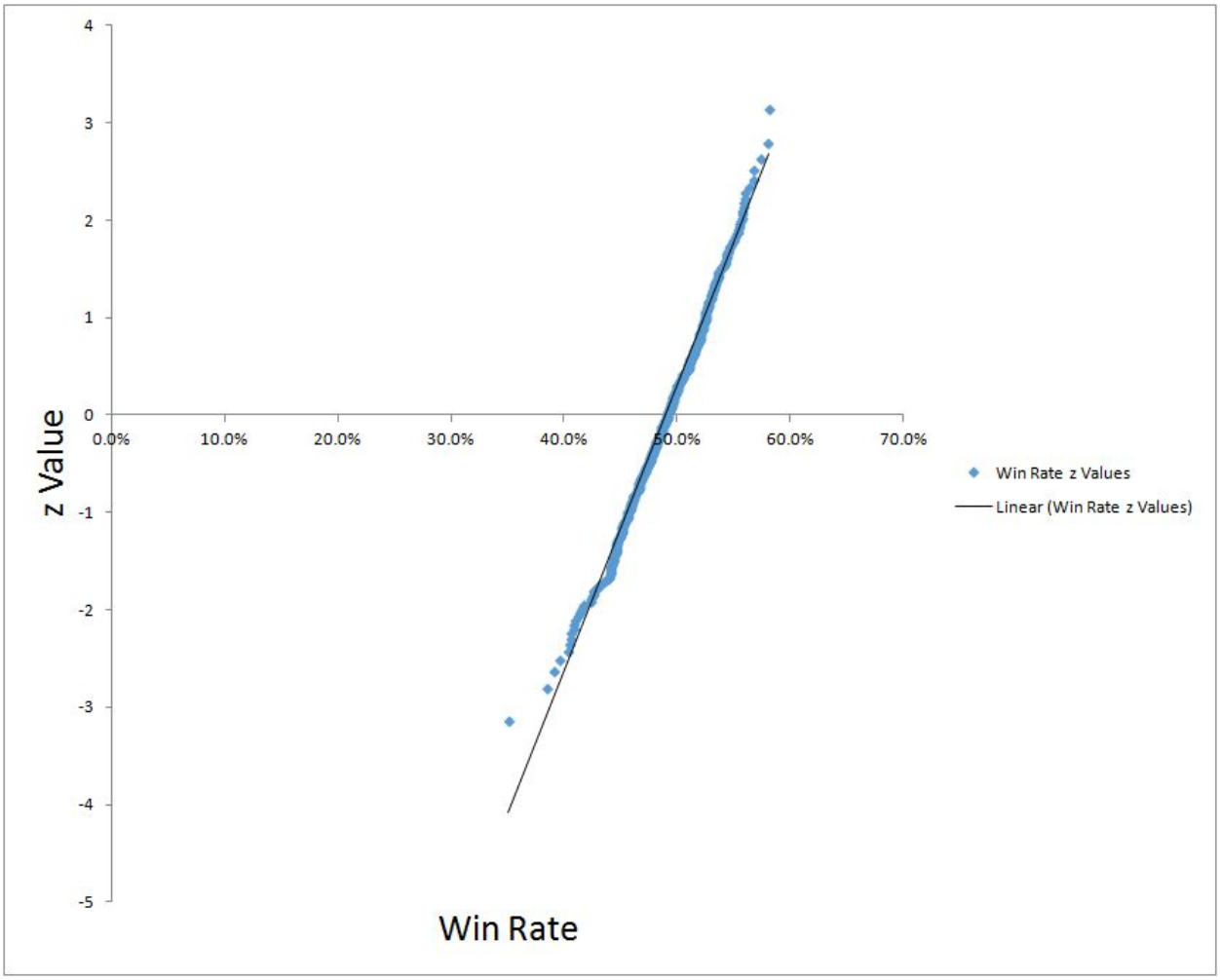

Figure 32: Normal Probability Plot for Assassin Win Rate

Figure 33 shows the average win, pick, and ban rate for each role and the overall averages.Figure 34 displays all of the roles and their rates relative to the average rate for all champion roles in a table. In Figure 34, a dash represents being similar to the average rate. This means within 0.5% for win rate, 1% for pick rate, and 1% for ban rate. An upwards arrow represents a rate that is higher than the average rate by the amounts mentioned prior. A downwards arrow represents a rate that is lower than the average rate by the amounts mentioned prior.

Role	Win Rate Avg	Pick Rate Avg	Ban Rate Avg
Marksman	49.8%	11.4%	1.8%
Support	50.3%	10%	6.1%
Tank	50.3%	4.6%	3.8%
Assassin	49%	4.3%	6.9%
Mage	49.7%	7.5%	3.8%
Fighter	50.9%	9%	5.8%
All Roles	**50%**	**7.8%**	**4.7%**

Figure 33: Table of Roles and Average Rates

Role	Win Rate	Pick Rate	Ban Rate
Marksman	—	▲	▼
Support	—	▲	▲
Tank	—	▼	—
Assassin	▼	▼	▲
Mage	—	—	—
Fighter	▲	▲	▲

Figure 34: Table of Roles and Relative Rates

Something interesting to take away from this breakdown by champion role is how the win rate for all graphs looked almost identical despite the ban rate and pick rate varying. All roles' win rate data scored over 0.96 for the correlation coefficient of their normal probability plots. In addition, all were centered within 1% or 2% from 50%. This could show that perception of champion strength, represented as pick and a ban rate, might not be as correlated to actual performance, represented as win rate, as the public might think. Another interesting take on this would be to look at the rate data for entire 5 champion team compositions, rather than individual roles, to see how different combinations are affected. Our data gathering system unfortunately does not gather data on entire teams, but just individual champions.

4.4 Analysis of Patch Notes

In this section, the patch notes are analyzed using the categorization system developed and described previously. There are several interesting trends in the types of changes made to the game at different points. The first trend is that of patch cycles, where the number and type of changes made are cyclical in nature, rather than consistent across all patches. The next is the tendency to release more buffs than nerfs. Finally, champion releases also show interesting trends across seasons.

When deciding what changes to make, Riot generally focuses on champions that are too weak or too strong. Patch notes usually contain a general explanation of Riot's goals within that patch as well as the notes themselves. Riot uses the win rate data they collect internally to determine how best to balance the game, as well as several other factors such as how popular champions are, who tends to play them (less experienced players vs dedicated players), the skill required to play them well, how much counterplay the champion has, and player

perceptions of champion power[14]. Often, the Public Beta Environment (PBE) is used to test out more radical changes to champions before making those changes on the live servers. All of this testing and consideration contributes to patches varying in size and goals, leading to patch cycles. The preseason occurs during the two or three patches prior to the start of a season. This occurs after the World Championships and before the start of NA and EU seasons of competitive play. In addition, ranks for all players are reset during this time and rewards are distributed for achieving particular ranks during the past season. Riot uses this time, generally somewhere around November of one year to January of the next, to make larger changes to the game, map, items, and champions. Generally the first preseason patch will contain a large number of changes, and subsequent patches will contain slightly less as Riot works out the issues with the changes they have made. This cycle also occurs during the season, with less radical changes. Some patches are larger and intended to shake up the game's balance, while some are more for fine tuning and fixing smaller issues. Generally one large patch will be followed by several smaller ones in a cyclical manner. In Figure 35 below, a section of data from Season 4[15] and Season 5 illustrates this cyclical nature. In patches 4.10, 4.11, 4.17, 4.21, 5.5, 5.8, and 5.12, there are a larger number of changes than in the patches in between. Also, most of these large patches occurred near the end or beginning of a season, rather than in the middle. The rest of the patches generally follow this pattern, with some anomalies due to certain patches containing changes that affect every champion. Riot occasionally rebalances certain champion stats and how they work alongside items, so occasionally one patch will, for example, raise each champion's armor by a few points.

[14] http://na.leagueoflegends.com/en/news/game-updates/gameplay/data-and-champion-balance-part-1
[15] After Season 3, Riot changed the naming convention for seasons to the year of the season, e.g. Season 2014. To remain consistent with the patch names and avoid confusion, we use Season 1-Season 6 instead.

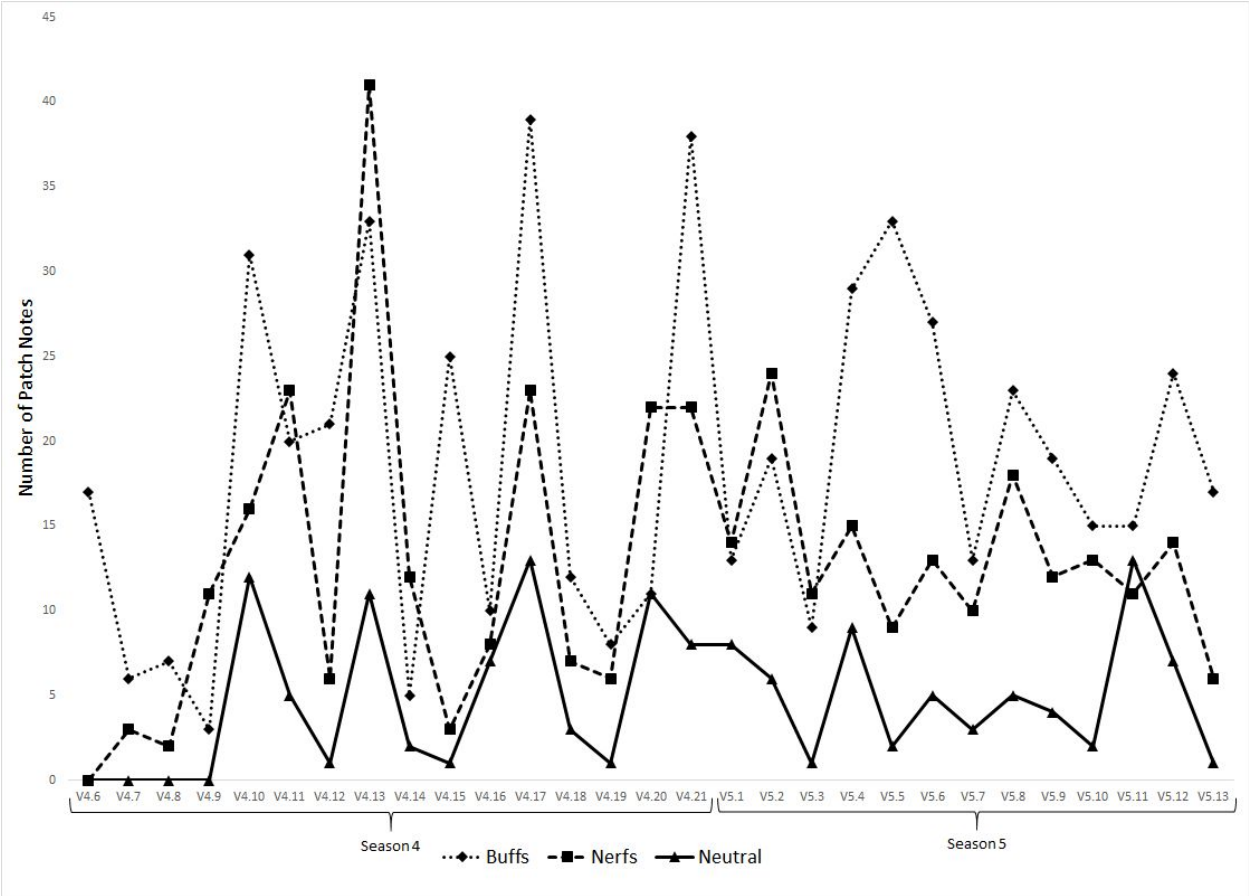

Figure 35: Patch note type totals for patches from Season 4 - Season 5

Figure 35 also illustrates another trend. Generally, buffs are the most numerous type of change, followed by nerfs and then neutral changes. Only 7 out of the 29 patches in the graph had more nerfs than buffs. There are many potential explanations for this, such as the possibility that buffs are smaller in scale but more numerous and nerfs tend to be more severe, or that more champions are underpowered and only a small number are overpowered, leading to this pattern. An overpowered champion will quickly have a destructive effect on the game's balance as players are forced to pick or ban that champion or be at a huge disadvantage. An underpowered champion will simply be picked less often than other champions, but the game can still be balanced because there are many other viable choices for players to make. In addition, players may feel negatively towards Riot if they feel nerfs to their favorite champion are

undeserved, and may even play the game less if the nerfs affected their enjoyability. In contrast, the opposite occurs with buffs so Riot may be more careful to make sure that nerfs are deserved.

4.5 Effects of Patch Contents on Rates

Most important changes can be placed in one of two categories, nerfs and buffs. Simplifying the patch categories in this way allows the analysis of the content of a patch alongside its in-game results. In Figures 36-38, the change in one rate from the previous patch to the current one is graphed with the contents of each patch, represented by the number of buffs totaled (number, utility, and quality of life) minus the number of nerfs. Each point represents one champion on a particular patch. As an example, in Figure 36 one individual point represents the champion Ahri during patch 5.1, where her win rate decreased by 5.9% from the previous patch and she received 2 buffs and 8 nerfs, for a score of -6. Each champion and patch combination is similarly graphed, and the pick rate and ban rate graphs were constructed in the same fashion.

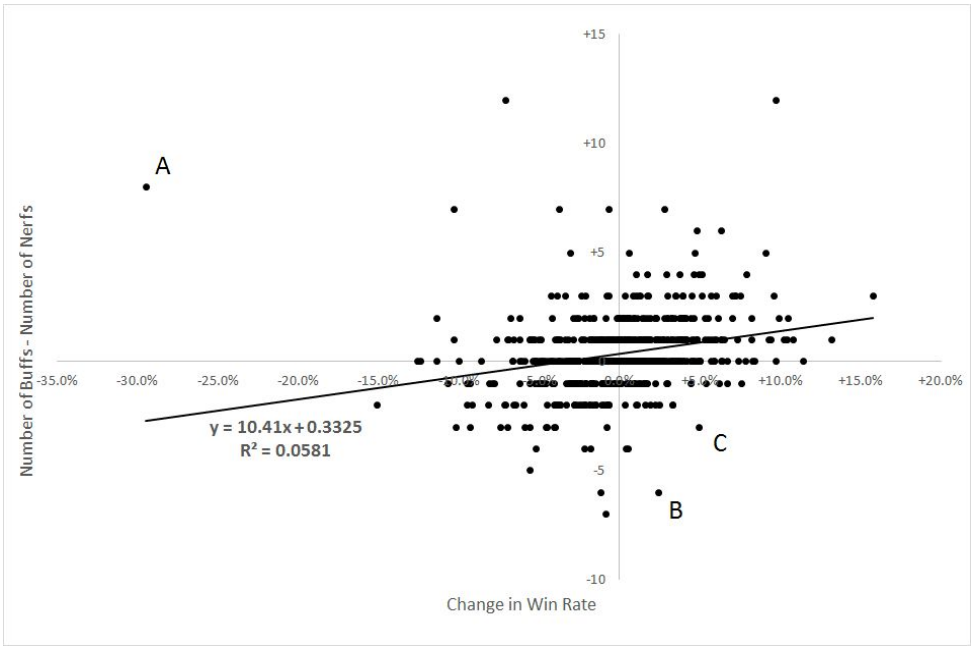

Figure 36: Patch Contents and Win Rate

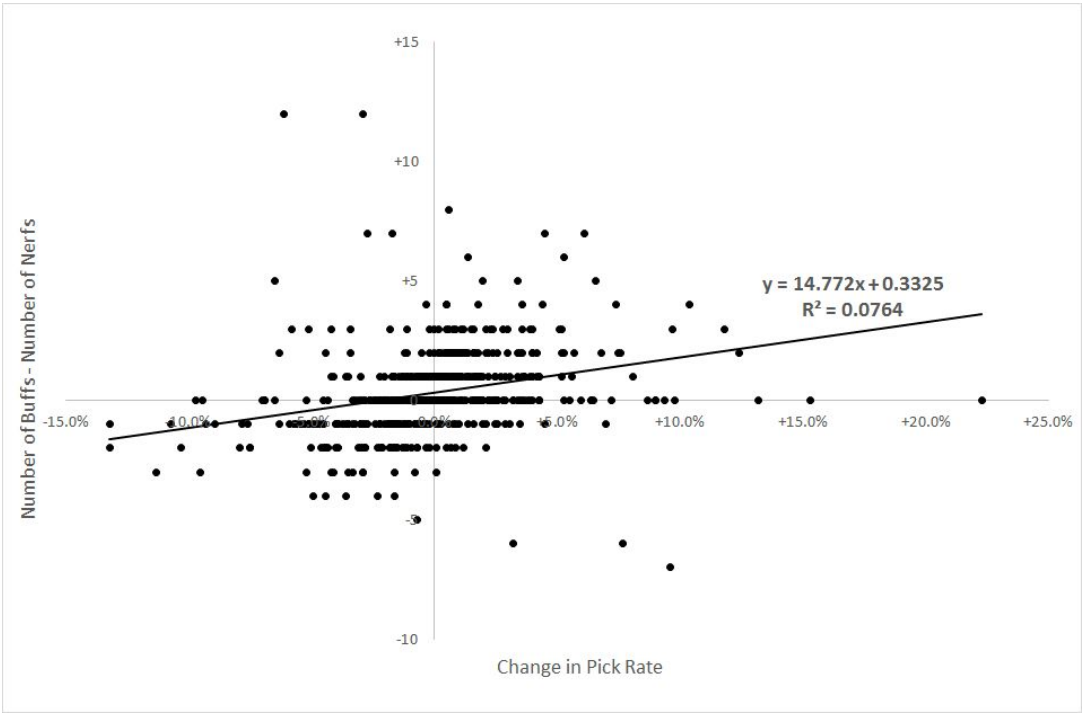

Figure 37: Patch Contents and Pick Rate

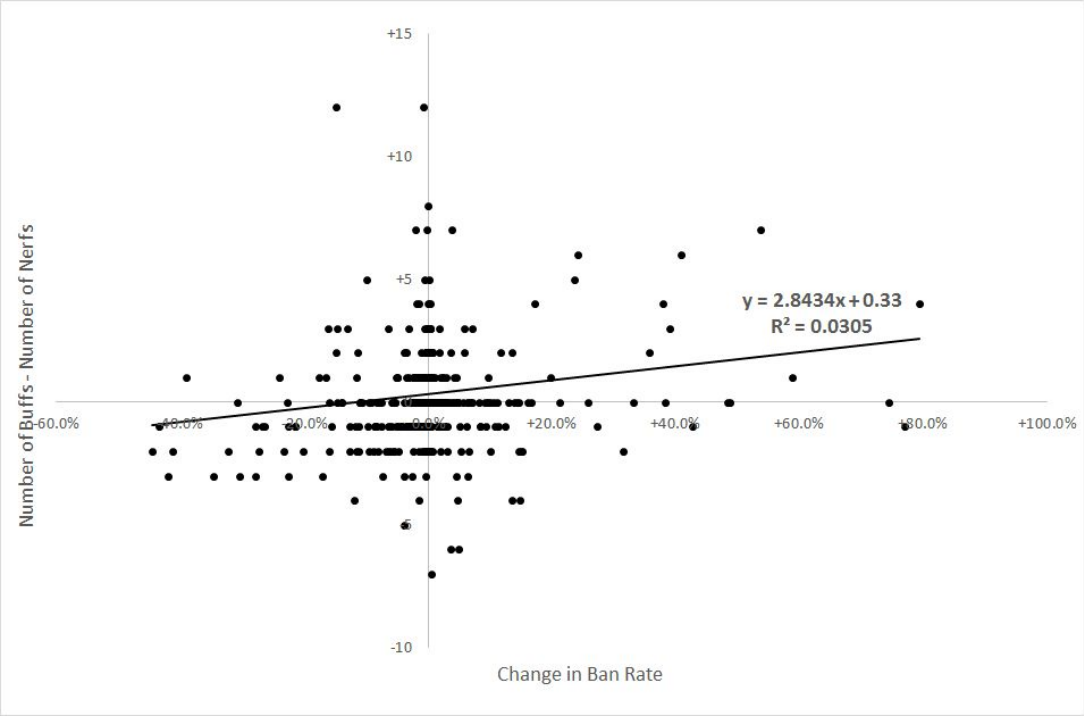

Figure 38: Patch Contents and Ban Rate

There are several outliers in Figure 36 that are interesting to note. The point marked A represents Urgot on patch 4.15, where he had a decrease in win rate of 29.5%, but the patch contents contained 8 buffs. Before this patch, Urgot was a very unpopular champion with a somewhat low win rate and a very low pick rate. His pick rate doubled from 0.5% to 1% during this patch due to a minor rework containing several buffs and a bug fix. This influx of players playing a somewhat odd and difficult champion that they likely hadn't played much in the past was the most likely explanation for why the win rate dropped so suddenly. Also, due to Urgot's unpopularity the sample size is much lower than for most other champions, so there is likely some sampling error.

Two other outliers fall on a different part of the graph. Points B represents Gangplank on patch 5.14, where he received a large rework of his abilities. Overall many of the changes were small statistical nerfs to things like armor or health that likely didn't have a large effect on win

rate. He also received many large neutral changes to his abilities that changed how they fundamentally worked. At the time of the patch, it was unknown whether these changes would be a net positive or negative, which is why they are classified as neutral. However, the new abilities turned out to be much more effective and synergistic with one another, leading to an increase in win rate. Point C represents Kalista on patch 5.17, where she received three nerfs. Her winrate increased by 4.9% with no real explanation, as the nerfs were unambiguously negative. This could possibly be explained by simple sampling error, although Kalista was quite popular and had a large number of games on that patch. The most likely explanation is that changes to other champions or past item changes caused the increase in win rate. In patch 5.16, the previous patch, several fighter champions were buffed and widely considered quite overpowered. These champions excel at killing marksmen, which is Kalista's role, but Kalista has better than average ability to escape from enemies attempting to chase her down. This may explain why her win rate goes up instead of down. Her win rate drops off by the next patch, suggesting that the nerfs were eventually effective.

These graphs show a weak correlation between each of the three rates and the type of changes made in the patch. Pick rate has a higher correlation than win rate, possibly suggesting that player perception following a nerf or buff causes larger changes in the pick rate, while the actual champion power did not change much. Also, this graph and our system in general does not take into account the scale of the change. Most simple number changes are fairly similar in scale, but some major changes, especially in champion reworks, have huge effects on the champion's power and playstyle. The scale of the change is not represented on the graph, only the number of changes, so this effect is not measureable using the current system. Reworks are also not categorized separately, but would likely appear on the graph as patches with a large number of changes.

Additionally, bug fixes and neutral changes are not on these graphs as they do not have a predictable effect on champion success. Bug fixes often benefit the champion, but the bug may have been beneficial to the champion and fixing it may instead benefit the enemy team playing against that champion. Similarly, neutral changes often change how abilities work or when in the game they are more or less powerful, so their effects are unpredictable.

Changes to items as well as game mechanics like jungle monster and tower strength can also affect certain champions disproportionately, causing changes in rates independent of that champion's patch notes. In addition, skins, which do not affect gameplay but merely change the appearance of a champion, can cause a large number of people to play a certain champion to use the skin if it is particularly creative and appealing. This could have no effect on win rate, or could cause a drop because inexperienced players are playing the champion more than before. Unfortunately skin releases are not documented in the patch notes so it is impossible to do any statistical analysis of these effects in this project, but there is potential for future research.

4.6 Champion Case Study

Along with these analyses of the rate data and patch note data, we also conducted a case study of one champion to show how the rate data and patch note data are correlated, if at all. This process required human intuition for accuracy, as the patch notes are categorical in type, and cannot be accurately quantified without background knowledge. For example, one patch might contain three buffs while another patch only contains one buff, however, the impact of the single buff might outweigh the combined impact of the three buffs. Quantifying patch note data on frequency of type does not provide enough information alone. Because human interpretation is needed, it would have been unreasonable to attempt to derive a relation

between rates and patch notes for all champions, so we chose one that would best encapsulate all possible changes.

The champion we observed in this case study is Nidalee. Nidalee was chosen for several reasons. The most important reason is her patch notes contain at least one of every type of the 11 possible categories of changes. This means we will have at least a minimal insight into how each change might affect win rate, pick rate, and ban rate. Another reason Nidalee was selected is that she has gone through a rework. This means she has gone through a major overhaul of her abilities, changing many aspects about how each one works, all within one or two patches. Such a drastic and sudden change would likely affect all three rates significantly. Nidalee is also one of the oldest champions in the game. She was the 42nd champion released and was created when the game was still in closed beta, meaning she had a long history of changes both in balance and stylistically to keep her relevant as the game progressed.

Figure 39, below, shows the changes in win rate, pick rate, and ban rate of Nidalee. The data is organized so that the point at each patch version is the resulting rate in response to that patch's changes. For example, a data point above patch 5.7 means that rate data was taken at the end of patch 5.7 and right before the start of patch 5.8, so all patch note changes in patch 5.7 were active for the collection of that data point.

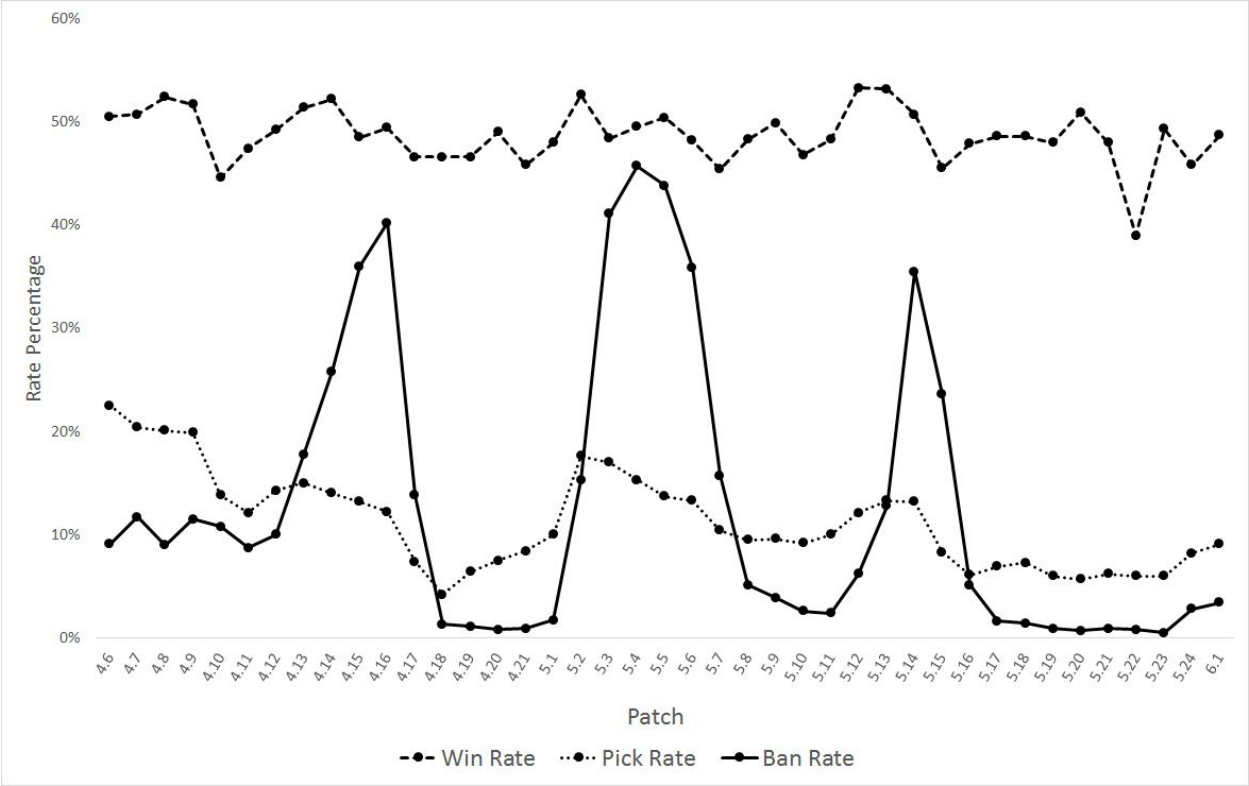

Figure 39: Nidalee Win, Pick, and Ban Rates

The most isolated and controlled patch in terms of types of changes was patch V5.15. This patch had four number nerfs and nothing else. From this, we see a somewhat expected response in the graph's pick and ban rates. The champion is no longer as strong due to the patch, so her ban and pick rates both decrease significantly after patch V5.15, seen in Figure 39 above. Win rate also experiences a noticeable decrease for V5.15, but rebounds by the next patch and stays fairly consistent for the next several patches.

Neutral changes have no objective positive or negative benefit. There are certain cases where a neutral change will be overall more positive or overall more negative, but this depends on the given situation. Nidalee has a great isolated example of this in patch V5.5. One of her abilities lost its bonus damage on low health champions. However, in exchange its base damage was increased and benefitted from ability power. This caused the ability to do less

damage to low health enemies but more damage otherwise. From this point of view, there are situational benefits, so what often decides if the change is a buff or nerf overall is how common each situation is and to what degree each situation is beneficial. For the case of this patch, it was decidedly a nerf in terms of rate change. Win rate immediately plummeted while pick rate and ban rate slowly followed, as seen in Figure 39 above. In cases with neutral changes, the effect on ban and pick rate is usually not as decisive, because there is no objective buff or nerf status that is altering public opinion, which those rates are mainly determined by. Win rate, on the other hand, does a better job of reflecting the change, as it is not as influenced by popular opinion.

Buffs of any kind usually come with increased pick and ban rates for that champion. This is especially true for utility and quality of life changes. When a number change is made, the champion will just feel stronger. Utility changes often bring a new feature to a champion and quality of life changes make a champion more intuitive and easier to use. Both these attributes lead to greater interest in playing the champion in addition to strengthening them. Patch V5.2 is a great controlled environment for this study as it has only two changes, one quality of life buff and one utility buff. These changes added an interesting feature and made Nidalee feel more responsive, this led to a spike in pick rates and ban rates for a few patches, as people wanted to explore the changes. Win rate also responded with an increase, but the gameplay aspect of these changes was not too significant, so reasonably, win rate leveled back out shortly after the patch.

Visual updates and bug fixes are generally not too influential on any of the three rates. Bug fixes can be influential if the bugs being fixed are incredibly detrimental or beneficial to the champion, but it is rare that bugs of this kind occur. Nidalee only has one controlled instance of a visual bug fix, patch V5.3. In this patch, a bug was fixed where Nidalee’s cooldown timers

were sometimes set incorrectly when using one of her abilities. This fix was significant in comparison to most bug fixes, but it was still unlikely to cause any greater change in rate data than general variance would. This is reflected in the rates data, as the changes for patch V5.3 are negligible overall.

In addition to observing how individual change types affect champion rate data, we also wanted to see how rates were affected by a rework. Nidalee underwent a rework in patch V4.10, receiving 45 changes in one patch: 22 buffs, 10 nerfs, 10 neutral changes, 2 visual updates, and a bug fix. Another counterintuitive phenomenon occurs with many reworks. It would make sense for reworks to encourage more players to pick a champion because there will be many new or changed features to experience. This is true for the most part, and champions do tend to see a pick rate increase for certain players. However, many reworks are actually taken very negatively by players who previously played the champion to some degree. While reworks do instill an influx of newer players to try out a champion, they often simultaneously push away other players who were used to the features of the old champion and do not want to see them go or change. This means the deciding aspect of whether net pick rate will increase or decrease is prior popularity. With champion that are very popular, a rework will upset a large portion of the community that dedicated their playstyles to that champion. If a champion is not very popular, a rework will only negatively affect a small minority of dedicated fans. For Nidalee, it seemed to be the first case. She was a very popular pick before the rework, and even though the majority of the rework changes consisted of buffs, many players either gave up on or took a break from playing the champion due to a dislike of the changes. The pick rate data suggests this, as there is quite a large drop in Figure 39 at patch V4.10. Win rate after a rework is generally in a state of flux. Many variables aside from the actual strength of the champion come into play, such as unfamiliarity playing the champion with the new changes, unfamiliarity of the other team playing

against the champion with the new changes, new players who have little experience with the champion trying it out en masse, and the more experienced players halting their play. In Nidalee's case, there was a significant drop in win rate similar to the drop in pick rate. This supports the possibility of several combinations of the previously mentioned confounding variables. Ban rates are usually less complicated. Champions that had relatively high ban rates prior to the rework are often banned a similar amount, as people often fear what they have little experience playing against. For the same reason, champions who were not often banned prior to the rework generally see a rise in ban rate. It is usually rare for a champions ban rate to rise or fall a significant amount unless the changes were very objectively positive or negative. One case of this was for a rework Darius went through, increasing his ban rate from 1.4% to 94%(the highest ban rate we observed) over the course of 4 patches. For Nidalee, there was no significant change in ban rate, as she was a popular pick and ban prior to the rework, and therefore had a decently high ban rate to start with.

4.7 Summary

For game developers, it's important to study and recognize the effects past changes have made on overall game balance. Did past changes accomplish what they were meant to do? Were there any unintended side effects? Are there any other methods that might lead to the same desired effect? Analyzing these cause and effect relationships helps better prepare a balance team for efficiently solving future problems. It also can help educate the player base on the science of game balance to assuage them when the champion they enjoy playing is nerfed or the one they hate playing against is buffed. There is no shortage of player outrage for any and all patch notes, so having statistics defending balance changes can be useful in maintaining

a good developer to playerbase relationship. The analysis we have provided, along with the patch note categorization system, are both additional useful tools available to Riot's development team and the general playerbase.

5 Conclusion & Future Work

Games like League of Legends that are played through an online platform have the unique ability to be able to continually updated and changed. Because of this, Riot Games must decide how and when to make changes if they want to balance their game. Despite there being many existing sites that mine data through the Riot API and release interesting, personalized statistics, none have effectively analyzed the relationship between balance updates and the resulting effects on the game. Our project takes on this task using a classification system to order and quantify patch note changes, and tracks several in game statistics sampled from hundreds of thousands of games. With these proper tools, we are able to the best of our ability find relationships between balance updates and their effect on the game. For this project, our team explored and connected many facets of computer science such as data parsing, automated categorization, API calls, sampling techniques, database management, website design, and human computer interaction to develop a website and create analysis which can help to determine when to make these balance changes.

The website allows users to view changes made in League of Legends and how those changes affected the champions they are associated with. The project also analyzed the collected data by testing hypotheses and observing trends. While there are other websites which provide much more information about the state of League of Legends, our project provided a new perspective by specifically comparing categorized changes to a champion's statistical performance. The website was informally tested with some League of Legends players and feedback was used to improve its features and ease of use. It has also undergone a

formal inspection by a Riot Games employee and approved for the use of a Riot API production code.

In analyzing rates and patch notes, we observed many relationships. After calculating central tendencies and other distribution statistics, we analyzed the relationships between the types of rates in order to determine if there was a relationship between them. All of the relationships had R-squared values which were too small to make significant claims. However, trends could be seen in the data. Pick rate versus ban rate had a weak negative correlation while the other two combinations had weak positive correlations.

We then analyzed the rates with respect to champion roles, determining how the rates differed among them. We found interesting patterns when comparing rates to the average rates among all champions. Marksmen and Supports had higher than average pick rates because they are a staple part of any team composition. Fighters seemed to be above average in all rates, implying they are priority picks and bans. Despite having an under average win rate, Assassins have the highest ban rate of all roles.

The distribution of patch note categorizations was also analyzed. Generally, Riot releases more buffs than other types of changes, followed by nerfs and neutral changes. Other types of changes were made even less frequently. Buffs are likely the most common because of public opinion and effects of overpowered and underpowered champions. Slightly buffing a champion multiple times is less risky than one large buff because if a champion is too powerful, it tends to break the flow of gameplay. However, when a champion is overnerfed, they simply stop being played.

5.1 Future Work

Listed below are some known faults or improvements which could be made to the project if it were to be worked on in the future. Currently, it is difficult to observe shifts in the data when viewing graphs where more than one rate is being displayed. A powerful solution to this is to have multiple y-axes. HighCharts has this functionality but we did not implement it in the time frame for this project. Additionally, it is difficult for a user to tell which points on the chart are associated with patches where the champion was actually changed without hovering over the point. To address this, patches where the champion was not changed could appear as a different color or size. HighCharts likely has this functionality, but it was not looked into. Another issue occurs when a user clicks on a point and the patch notes scrolls to that patch. This scroll is sometimes difficult to notice. Visual feedback when clicking on a point can be helpful to communicate to the user that the page has reacted to the click. This could be done by highlighting the patch notes text area for a brief period of time.

We also have the data to be able to implement many more features if we desired. The ability to look through all the changes in a certain patch or perform customized searches could be added. Additionally, statistics about the data we have collected could be shown to the user. The Riot API also contains much more information which we ignore but could use for many purposes. The website could display the win rate of champions who purchase a certain item or display statistics about spell usage. We could also examine rate data for entire team compositions, rather than individual champion roles, to see how teams composed of various different combinations of roles fare.

Other features that could be implemented are related to interactions with the graph. Being able to view multiple champions on one graph would be very interesting and would

require some changes in how the graph is created. Currently, it uses a URL parameter to determine which champion's information to display. This can be modified to allow for multiple champions to be displayed. Allowing the user to zoom in or out on the graph in order to display a different number of patches at once would also be a helpful feature to have. HighCharts has this functionality and zooming could be implemented.

Additional analysis on how different types of changes affect rates would also be interesting to observe. In our analysis in section 4.4, we simplify number buffs, utility buffs, and QoL buffs and their corresponding nerf types into one number to represent the overall change to the champion during a patch. Then, we observe how those changes in general affect rates. However, we do not capture which types of changes, if any, have more of an effect on rates.

Another point of possible analysis in the future could be the analysis of how champion skins can affect rates. Skins are visual changes that players can purchase outside of the game to customize their champion in game. While skins have no direct effect on the gameplay, they can correlate with rates in some way. For example, a popular skin release may influence players to play a champion, increasing the champion's pick rate without any other gameplay changes. However, this is not able to be gathered in the Riot API.

If this project were to be developed further, an important improvement would be to automate the website as much as possible. There will always be some aspects that must be done manually. Tasks like completing patch categorization and updating images will inevitably be done manually, but much of the website could be automated further. We currently use a local program to gather and create a file which contains game data. We then manually insert the data into the database. This process is very active but could be done passively and routinely on the website, further increasing the reliability and modernity of the data.

6 Appendices

Appendix A: Riot API

During our initial research, we collected a list of data that can be found using the Riot API. Our findings informed our decision on what data to base the site around, and what questions we would answer. Note that this is not an exhaustive list of what the Riot API can provide. This is simply information we believed might be useful in the early stages of planning this project. For a full list, please refer to https://developer.riotgames.com/api/methods.

- (Dynamic) Champion Information
    - The champion's ID
    - Whether the champion is enabled (could be disabled for bug fixes, etc)
    - Whether the champion is enabled in ranked
    - Whether the champion is allowed as a bot in custom games
    - Whether the champion is allowed as a bot in co-op vs AI games.
    - Whether the champion is free-to-play right now

- Information for a currently running game (Can also get all of this for featured game)
    - Banned Champions
        - Their IDs, the turn in which it was banned, and the team that banned it
    - The game's ID
    - The game mode, mapID
    - Queue Type
    - Start time, time so far
    - Participant Info
        - Whether the participant is a bot
        - The champion the participant is playing
        - Summoner Name, Summoner ID
        - Runes, masteries, summoner spells
        - profile icon used

- Information for all of a summoner's recent games
    - Champion played, map (Summoner's Rift), Game Type (custom vs matchmade)
    - Game Mode
        - CLASSIC, ODIN, ARAM, TUTORIAL, ONEFORALL, ASCENSION, FIRSTBLOOD, KINGPORO
    - IP earned, player level

    - Other Summoners in game
        - Champion played, summoner ID, team ID
    - Other Game Information
        - Kills/Deaths/Assists/Damage/Buildings destroyed etc.
            - Essentially all the things you can find in the score screen
        - Who got first blood
        - Items at end of game, total number of items purchased, number of tier 3 items built
        - Number of killing sprees
        - The position the player was playing
            - TOP(1), MIDDLE(2), JUNGLE(3), BOT(4)
        - Role they were playing
            - DUO (1), SUPPORT(2), CARRY(3), SOLO(4)
        - Wards purchased, killed, and placed
            - Distinguishes between pink/green wards only for number purchased
        - Total time of CC dealt
        - Number of times each ability is cast
            - All 4 abilities and summoner spells
        - Level at end of game
        - Season the game was played during (SEASON3, PRESEASON2014, SEASON2014, etc.) (Important note: we later discovered that only games played after patch 4.6 could be accessed)

- League Information
    - You can access the entirety of master and challenger tier
    - The queue type of the league
        - RANKED_SOLO_5x5, RANKED_TEAM_3x3, RANKED_TEAM_5x5
    - Tier
        - CHALLENGER, MASTER, DIAMOND, PLATINUM, GOLD, SILVER, BRONZE
    - Information for every player/team in league:
        - Division Name
        - Whether they are new to the league, a veteran of the league, inactive, or on a hot streak
        - Wins, losses, League Points
        - Player/Team Name, player/Team ID
        - Mini-series data
            - wins/losses in series (and order in which they happened), whether it is a Bo3 or Bo5

Appendix B: Database Software Comparisons

Storage System (DB-Engines rank)	**Supported OS**	**Supported Languages**	**Index Types**	**Interface Access**	**General**
Oracle (1)	AIX HP-UX Linux OS X Solaris Windows z/OS	C C# C++ Clojure Cobol Eiffel Erlang Fortran Groovy Haskell Java JavaScript Lisp Objective C OCaml Perl PHP Python R Ruby Scala Tcl Visual Basic	Full-text Hash R-/R+ Tree Bitmap Expression Partial Reverse	GUI SQL	- Large companies - Large projects - Many features - Very powerful - Very complex - Commercial (closed source)
MySQL (2)	FreeBSD Linux OS X Solaris Windows	Ada C C# C++ D Eiffel Erlang Haskell	Full-text Hash R-/R+ Tree	SQL	- Small companies and startups - Small to mid-sized projects - Fewer features - Prioritizes reliability,

		Java Objective-C OCaml Perl PHP Python Ruby Scheme Tcl			performance, and ease-of-use - Open source
SQL Server (3)	Windows	.Net Java PHP Python Ruby Visual Basic	Full-text Hash Bitmap Clustered Nonclustered Filtered Spatial XML	GUI SQL Various	- Inbetween MySQL and Oracle in terms of complexity - Powerful but requires a lot of overhead - Long installation - Commercial (closed source)
Flat File	Any major OS	Any	Depends on implementation	Depends on implementation	- No prebuilt features - Reinventing the wheel - Complete control - Full knowledge of code's inner workings - Security issues

Appendix C: Website Mockups