Research article

Philip Z. Maymin*

Smart kills and worthless deaths: eSports analytics for League of Legends

https://doi.org/10.1515/jqas-2019-0096 Received September 10, 2019; accepted September 5, 2020; published online September 21, 2020

Abstract: Vast data on eSports should be easily accessible but often is not. League of Legends (LoL) only has rudimentary statistics such as levels, items, gold, and deaths. We present a new way to capture more useful data. We track every champion's location multiple times every second. We track every ability cast and attack made, all damages caused and avoided, vision, health, mana, and cooldowns. We track continuously, invisibly, remotely, and live. Using a combination of computer vision, dynamic client hooks, machine learning, visualization, logistic regression, large-scale cloud computing, and fast and frugal trees, we generate this new high-frequency data on millions of ranked LoL games, calibrate an in-game win probability model, develop enhanced definitions for standard metrics, introduce dozens more advanced metrics, automate player improvement analysis, and apply a new player-evaluation framework on the basic and advanced stats. How much does an individual contribute to a team's performance? We find that individual actions conditioned on changes to estimated win probability correlate almost perfectly to team performance: regular kills and deaths do not nearly explain as much as smart kills and worthless deaths. Our approach offers applications for other eSports and traditional sports. All the code is open-sourced.

Keywords: analytics; eSports; League of Legends; machine learning.

1 Introduction

How much does an individual's performance contribute to a team's likelihood of winning?

This is a central question of sports analytics for traditional sports as well as eSports. Like points scored or defended in traditional team sports, eSports measure kills and deaths. But, again like traditional team sports, these standard metrics do not do a great job of predicting team performance. There is some correlation, but high scoring players on bad teams are so common as to be a cliché—in all sports. Unfortunately, there are not that many games to analyze in traditional sports. eSports, however, is different; millions of games are played every day.

Being digital, eSports should have vast and easily accessible quantities of data, but sometimes they do not. League of Legends (LoL), the most-played multiplayer online battle arena (MOBA) game, only has boxscore-equivalent statistics provided by the game's publisher (Riot Games) through their free application programming interface (API, c.f. Riot Games 2018): e.g., when each champion died, or got a kill or an assist, or similar. All current LoL analytics from amateurs to pros rely on this rudimentary API data.

1.1 Description of League of Legends

LoL, often referred to as LoL or simply League, is a five versus five computer video game reminiscent of capture the flag. The two teams start in their respective bases on opposite corners, bottom left and top right, of a largely jungle-like environment with three main lanes connecting the two bases, referred to as top, mid, and bot. Each team's own nexus is placed near its base, with two turrets protecting it. There are also three other turrets on each half of the map along each lane, as well as one inhibitor per team for each lane. You can't attack the enemy nexus unless at least one of the enemy inhibitors are down, and you can't attack an inhibitor unless all of the turrets leading to it have been destroyed. Turrets, once destroyed, never respawn, but inhibitors do. The first team to destroy the enemy's nexus wins the game. Nothing else ultimately matters: not the number of kills or deaths of each player, or how much gold was earned, or even how many total turrets or inhibitors were destroyed. This makes ends of games

^{*}Corresponding author: Philip Z. Maymin, Fairfield University, Fairfield, USA, e-mail: philip@maymin.com. https://orcid.org/0000-0002-3926-8720

especially dramatic, including back-door plays and sneaky attacks and last-second races.

Each of the 10 champions can go anywhere they want on the map, and there are essentially no enforced or special positions such as a goalie for whom different rules apply. Much like basketball and the evolution of point guard, shooting guard, etc., positions, even though each of the five players on a LoL team could in principle do anything, a standard form of specialization has evolved. Typically, one player on each team plays the top lane, one plays the mid lane, and two play the bot lane. In the bot lane, one is called the "carry" or the primary damage dealer, and one is the support. The fifth player is called the jungler and tends to roam all over the map and kill the neutral minions found in the jungle.

Minions are non-player characters in the game, and come in two types: neutral ones found in the jungle that any player on either team could kill, and team ones who are like robotic soldiers for each team. Killing a minion grants gold. If it takes multiple hits to kill a minion, only the champion whose damage actually killed the minion gets the gold. Gold can be exchanged at each team's base for items that upgrade the champion with stronger or new abilities. Thus, in the bot lane, the carry tends to be the champion who kills the minions while the support often heals or offers shields or tries to damage the enemy. That way, the carry earns more gold and gets better items faster. Some items are primarily offensive in nature in that they increase the champion's combat abilities. Some are primarily defensive in nature in that they make it more difficult for enemies to kill you. Some are a mixture, or provide utility such as vision or shields for allies. Champions that have built primarily defensive items are called tanks and champions that have built primarily for damage are called squishies, because they tend to be easy to kill, so long as you can get past their ally tanks and other teammates looking to protect or "peel" for their carry.

Champions also gain experience or "XP" and level up at certain benchmarks if they are near things that die that are not on their team, for example enemy champions or minions or neutral minions. Leveling up allows you to upgrade one of your champion's base abilities. For example, you might have an ability that lets you spin over some distance; when you upgrade it to a higher level, perhaps the cooldown is reduced, or the damage it deals is increased, or both. All champions begin the game on level one and can reach a maximum level of 18. Most champions have an "ultimate" ability that is unlocked only when they reach level six. This ultimate ability or "ult" is usually the strongest ability of the champion, causing, for example, a vast amount of damage.

See Appendix 1 for further details about the game.

As is evident from the above description and the Appendix, League is a complex game. It is also constantly evolving, with new patches changing or modifying rules or champions approximately every few weeks. The standard data available to analyze is woefully insufficient to properly analyze gameplay.

1.2 League of Legends data

Using some techniques outlined in this paper, we can collect more comprehensive and granular data. We track every champion's location multiple times every second. We also track every ability cast, every attack made, and all damages caused and avoided. We track actual vision granted and denied in the fog of war. We track every player's health and mana and cooldowns. We track everything, continuously. And we do it invisibly, remotely, and live, all without any impact on the user, or even their knowledge. (There are no privacy concerns here: we only track games that allow public spectating.)

From this comprehensive data, we are also able to define much better stats and analytics than what is possible from the API. A myriad of things critical to understanding and evaluating the game cannot be determined from the API data. When you died, how many allies were around you? How many enemies? Why did you die? Was it for lack of vision? A deficiency in health or gold spent? Outnumbered? Ganked? Or simply mechanically outplayed? Was your death a worthless one, or did your team come out ahead overall? If you killed someone, was it a smart kill, meaning one that increased your team's likelihood of winning, or was it merely a kill that padded your stats but did not help your team? How much time did you spend doing productive activity versus how much time was wasted? In team fights, were your formations correct, tanks in the front, squishies in the back? Did you peel? Did you engage and disengage correctly? How effectively did you use your summoner spells, and did you use them offensively or defensively?

Without this new data, there is no systematic way to answer any of these crucial questions, and many others. Yet these are exactly the kinds of questions that LoL players, casual and professional, ask themselves when they review game film.

1.3 Our contribution

We began with LoL because it has by far the largest player base (Paul 2017) but our general methodology is applicable to virtually all eSports. Other MOBA sports either do provide more comprehensive data directly from the game publishers, such as Dota2, or have other tools that have extracted data, such as Matz (2017) for Overwatch, so that particular portion of our methodology would not even be necessary there. We hypothesize as a direction for future research that our findings about individual contributions to team performance would, however, carry over. Furthermore, we argue that insights from eSports analytics can in principle inform analytics in traditional sports as well, because of the vastly larger data that eSports can provide.

We analyzed millions of games of LoL to extract the high-frequency and high-fidelity eSports data. Next, we developed and calibrated a live, in-game win probability model based on the conditions of the game at each moment (see Figure 1). Finally, while we tracked all deaths and kills, we improved the standard metrics to focus on worthless deaths and smart kills. Worthless deaths are ones that do not increase your team's win probability. Smart kills are ones that do increase your team's win probability. With these two new metrics, suddenly the relation between individual performance and team performance is virtually a straight line. Figure 2 shows the difference between the upward sloping but relatively flat relationship of the standard stats on the left-hand side, and the practically

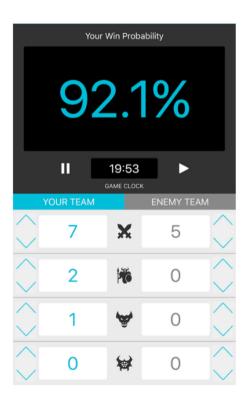


Figure 1: Win probability.

straight one-to-one relationship of our new advanced stats on the right.

Not only do these results suggest specific new ways forward for traditional sports, they also provide a possible novel method for generating brand-new insights: try things on eSports, where the data is far larger and cleaner, and carry those lessons back to the non-e-sports. Finally, with eSports itself poised to take over traditional sports in the future—LoL tournament viewership, for example, exceeds FIFA World Cup viewership (Keiper et al. 2017)—these insights will be valuable for their own sake as well, in addition to improving all other sports analytics.

2 Literature review

There are three classes of relevant existing literature.

First is research on LoL. The game itself is only about a decade old and there has been comparatively little research on it, relative to traditional sports. Eaton et al. (2017) use publicly available sources of data to show that the amount of time a team's best player is incapacitated relates directly to changes in standard kills as well as team performance, and they call out a need in their conclusion for the development of big data tools, techniques, and metrics in this space, which is exactly the need this current paper is aiming to fill. Deja and Myślak (2015) build a predictive model of team outcomes based on difficult-to-perceive topological features such as the areas of polygons connecting the players. Other papers use the public Riot API data for team style analysis or other aspects not directly related to our purpose here. For example, Lee and Ramler (2017) analyze whether typical team composition choices of players are in fact optimal, and do Nascimento et al. (2017) use machine learning and cluster analysis to identify seven profiles of varying team success.

Second is research on other MOBA games and how individual performance relates to team outcome. Xia, Wang, and Zhou (2017) conclude that the multiplayer killing indicator effectively predicts game outcomes. But they use the standard measure of kills (and deaths), rather than the win-probability-filtered ones we introduce in this paper. Drachen et al. (2014) describe a way of extracting positional data for Dota2, but the challenges of data extraction differ substantially between Dota2 and LoL. Schubert, Drachen, and Mahlmann (2016) describe a process for determining encounters for Dota2 that has some similarities to one of the ways in which we define team fights. However, our primary focus here is not on the definitions themselves but on the next step of determining

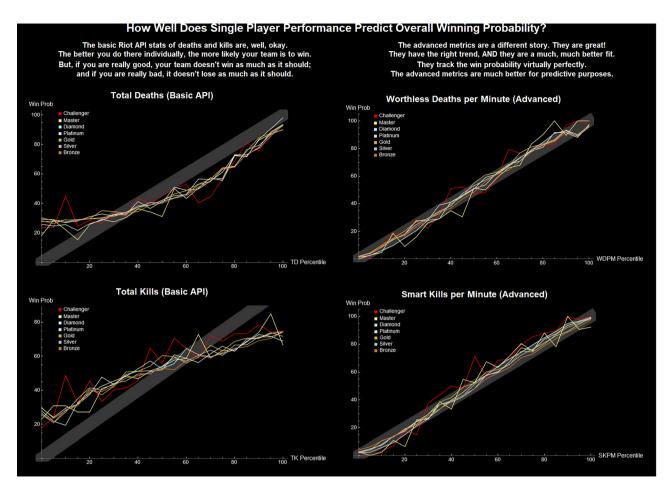


Figure 2: Regular deaths and worthless deaths versus team win probability.

individual contributions to team outcomes. Hodge et al. (2017) use machine learning on in-game Dota2 features to craft win probability models. They train a separate model for each minute of in-game play, whereas we opt to create a single model across all minutes of play. Our in-game metrics are similar to, though much simpler than, theirs because our model also doubled as an in-game tool, so our metrics needed to be the simplest possible. We also found in exploratory work that adding additional features did not substantively improve the win probability model. Yang, Qin, and Lei (2016) incorporate both pre-match and in-match features to predict game outcomes for Dota2. We rely only on in-match features. Cavadenti et al. (2016) take an interesting approach to player improvement by looking at deviations from relatively "normal" behavior to evaluate their relative gameplay. Here, we instead look at absolute gameplay on novel metrics, and compare them to peer performance. Yang, Harrison, and Roberts (2014) model combat through features of graph networks to attempt to predict overall team victory. By contrast, we are focused on individual play patterns here rather than teamwork or team fights. Rioult et al. (2014), similar to the

contribution for LoL by Deja and Myślak (2015), look at topological features of Dota2 such as polygonal areas described by the players. As above, we focus more on connecting an individual's performance with the team's outcome.

Third is research on traditional sports analytics, which is too numerous to list here. Examples of similar applications in traditional sports include the use of spatiotemporal data in the NBA by Cervone et al. (2014), in-game win probability models for the NFL by Lock and Nettleton (2015), and quantifying individual player performance in soccer by Duch, Waitzman, and Amaral (2010).

3 Methodology

There are two parts to our methodology. The first is the data extraction and collection specific to LoL and applicable only to other eSports whose publishers also do not provide sufficient data. The second is the more general analytics methodology that applies in principle to all eSports as well as traditional sports.

3.1 Data methodology

We have developed three independent methods for extracting detailed gameplay data. This section explains the basic principles behind our approaches in a high-level overview. The full details and all source code for all methods, both front-end and back-end, are open sourced and available online.

When 10 humans (referred to as "summoners" to distinguish them from the specific "champions" or avatars they control) play a round of LoL, and allow public spectating of the game, anybody can choose to watch the game live as it unfolds. The normal use cases for this are friends watching each other play, professionals streaming for their coaching staff, and commentators discussing a live professional game. When you spectate someone else's game, you do it through the same game client as if you were playing a game yourself. Rather than being a static video, you have various controls, including camera controls, time controls, and pop up information. You can see both teams if you choose. To avoid cheating, Riot broadcasts spectated games on a 3 min delay. That way, you can't have a friend spectate your game and tell you where the enemy is.

The network architecture is such that a player's actions are sent to Riot's servers, which respond with updates of the game. This happens quickly so that players experience

neither a lag between their issuance of a command and the ensuing movement, nor erratic enemy activity. This bidirectional network traffic is highly encoded, difficult to decipher, and subject to change with every patch. Patches happen regularly in LoL, approximately every few weeks.

For our methods of data collection, we rely on the fact that the LoL game client is running and decoding the network traffic appropriately. The results are displayed on the monitor exactly as they would if you were playing the game. Figure 3 shows a snapshot of a spectated game.

Note the plethora of information. While the center of the screen shows the action, and is the primary focus of a live gamer, the top shows the overall progress of each team in terms of gold, turrets, and kills, the bottom shows gold earned, items purchased, and the basic kills/deaths/assists stats, and the left and right champion bubbles show health, mana, level, and important cooldowns, but the most important information is in the bottom right minimap. The minimap shows the locations of all 10 champions, turrets, inhibitors, wards, traps, minions, and monsters.

The approach our eSports optical data tracking software takes is to visually process that minimap and infer all the locations it implicitly shows. We also track the other statistics and information from the other areas, but it is the minimap that is the most challenging and most rewarding. Technically, this approach relies on OpenCV, the industry



Figure 3: A single-frame image snapshot of a league of legends game in spectator mode.

standard open source computer vision library. Much of the difficulty in processing is due to champion overlaps and various other visually confusing information.

Here are some of the technical details for using computer vision. First, notice the white rectangle in the middle right of the minimap at the bottom right of Figure 3. Detecting this white rectangle is relatively straightforward, with a few edge cases: for example, it could be missing one of the sides. The white rectangle indicates what portion of the minimap is shown in the full view. The center of this rectangle indicates the position of the champion being tracked.

On the bottom left, note the various icons and numbers for the particular champion selected, in this case the blonde female mage called Lux. Each of those icons represents the current state of one of her abilities or summoner spells, including cooldown or additional effects. Her health, attack damage, ability power, and other stats are displayed in a standard font. In the bottom middle, a summary table of information about each champion lists the items they have purchased so far and their associated cooldowns if any, their kills, deaths, and assists, expressed as k/d/a, and the total number of minions killed. That screen can be toggled to also display each champion's gold earned in a standard font and text.

Two of our approaches to data collection rely on such computer vision live tracking using the spectator mode of the game client, one written in Python and one written in Go: the Python version was quicker to develop but the Go version was more robust. However, both approaches had a major downside: to get all 10 champion positions throughout the game, the entire game had to be spectated 10 times, so that the appropriate white rectangle could be detected. Therefore, we implemented a third approach, using direct hooks into the game client to determine actions, in a manner similar to scripting software such as Quipko (2017) that had often been used by players to attempt to cheat at the game (though note that our approach to data collection made any form of cheating impossible as the hooks were into the observer client, not any player's client).

In each case, cloud servers running Windows and game clients are automatically managed, starting and stopping in response to the volume of games requiring analysis. Each new instance launches and prepares the game client, activates spectator mode for the next game in the queue, and begins the optical tracking, shutting down when complete. The cloud instance needs an expensive graphics processing unit (GPU) for the game client to function; hence the motivation for the third approach to reduce the cost by 90%.

Table 1: Extractable optical tracking data for League of Legends, grouped by category.

Game	Minutes played; alive/dead status and health of turrets,
state	inhibitors, and monsters
Resources	Champion health and alive/dead status
	Mana or other secondary resource bar
	Cooldowns and status of all four abilities and both summoner spells
	Items owned and cooldowns if active
	Ward cooldowns
	Champion pings
	Gold earned and spent
Statistics	Attack damage; ability power; movement speed; attack speed
	Cooldown reduction; lethality and armor penetration;
	magic penetration
	Life steal; spell vamp; health regeneration; mana/
	resource regeneration
	Critical chance; critical damage; attack range;
	tenacity; armor; magic resist
	Current buffs such as baron buff, dragon buffs, or neutral monster buffs
	Current kills, deaths, and assists
Damage	Damage dealt and to whom and type
	Damage received and from whom and type
	Damage mitigated or shielded
	Neutral monster or minion damage
	Combos (the order in which abilities and attacks were executed)
Location	X location and Y location; velocity and acceleration; stealth status

Table 1 lists some the data we are now able to extract from games for each champion, each second.

One analytical advantage of eSports over regular sports is the number of games played. With our data methodology in hand from the Section 2, we can let it loose on an ever-expanding set of players, following each of their matches, and adding all that data to our universe. We start with a list of the top LoL players. Players in LoL fall into one of these ranks: Unranked, Bronze 5 through Bronze 1, Silver 5 through Silver 1, Gold 5 through Gold 1, Platinum 5 through Platinum 1, Diamond 5 through Diamond 1, Master, Challenger. Ranks are determined essentially *via* an ELO process calculated from your performance, much like a chess rating. By definition, Challenger players are the best 200 players in the world.

Starting with this list of 200, we can spectate all their games. Sometimes their games will include nine other challengers, in which case we simply analyze the game and move on. Sometimes their games will include at least one other non-challenger, so we add that person to our list. And when that person plays games, he likely also plays either with or against other people not already in our corpus, so it

Table 2: Number of players by tier and role in the corpus containing millions of games.

Tier	Тор	Jungle	Mid	ADC	Support
Bronze	15,984	11,734	15,615	15,871	15,771
Silver	8011	6172	8182	8043	6825
Gold	3186	2512	3065	3118	2582
Platinum	1995	1689	2120	2071	1758
Diamond	1322	1178	1272	1333	1228
Master	196	169	171	155	136
Challenger	209	200	236	215	191

continues to grow. Only future played games are spectated, so a player's history is not imported. By this iterative search process, we scraped several million games.

Note that our corpus of millions of games spanned several LoL patches, each of which can significantly affect gameplay. Hence, we focused on the ~150,000 games from the latest patch for the purposes of this section, which is still the equivalent of more than a full century of NBA seasons. Of those, about half were Bronze, a quarter Silver, 10% Gold, 7% Platinum, 5% Diamond, and less than 1% in each of Master and Challenger. Because all but about a thousand players had only a handful of games in our database, the breakdown of distinct players by tier and role shown in Table 2 follows approximately the same distribution.

3.2 Analytics methodology

In-Game Win Probability. Our first important analytics contribution is in calculating in-game win probability using a machine learning model.

While there are hundreds of potential metrics each second that could contribute to the win probability (including all the metrics in Table 1), we decided to limit it to the few most salient features so that the win probability could be estimated quickly by players whenever they have a brief few seconds of slow action in-game using a simple interface on a mobile app (see Figure 1).

These features are: the red and the blue team's kills, towers, and large monsters killed.

We also chose to look at each incremental minute rather than every second.

Our training set thus includes rows like this:

Minutes Elapsed, Red Kills, Blue Kills, Red Towers, Blue Towers, Red Monsters, Blue Monsters → Win /Loss

where the Win/Loss indicator output variable is based on the outcome of the complete game, and each of the values on the left is a whole number indicating the number of occurrences.

One immediate issue is that consecutive rows from the same game are highly correlated. To counteract this issue, and because we had the luxury of millions of games, we chose a random minute from each game, leaving us with millions of independent rows. Games typically last between 15 and 60 min, so this is sufficiently granular.

The machine learning model we used was logistic regression:

logit (Win/Loss) =
$$\beta_0 + \beta_1$$
 Minutes Elapsed + β_2 Red Kills
+ β_3 Blue Kills + β_4 Red Towers
+ β_5 Blue Towers + β_6 Red Monsters
+ β_7 Blue Monsters + ε

Following the standard definition and implementation of Wolfram (2017), this models the log probabilities of the indicator variable $Y_i = \begin{cases} 1 & \text{win} \\ 0 & \text{loss} \end{cases}$ with a linear combination of numerical features $x_i = \{\text{Minutes Elapsed}_i, \dots, \text{Blue} \}$ Monsters_i} weighted by the coefficients $\beta^{(i)} = \{\beta_0, ..., \beta_7\}$. The estimation of the parameters by stochastic gradient descent minimizes the loss function:

$$\sum_{i=1}^{n} \begin{cases} -\log (\beta^{(i)} x_i^T) & Y_i = 1 \\ -\log (1 - \beta^{(i)} x_i^T) & Y_i = 0 \end{cases}$$

Notably absent from our model is the total gold for each team. Both casual observers and professional commentators often use the net gold difference between teams as an indicator of win probability. However, this is an inherently flawed approach, especially as the game progresses, as the team with less gold can often win a team fight, ace the enemies, and march down to destroy the nexus. Furthermore, total enemy gold is not revealed to players in game and must be estimated.

Full-Game Win Probability. How much does an individual's total performance within a game contribute to his overall team's win probability? A version of this question appears across all sports (c.f. Cervone et al. 2014; Duch, Waitzman, and Amaral 2010).

Consider a dataset with rows of the following type, for each individual in each game:

Tier kills (or Deaths or an other individual metric across the entire game)→Win/Loss

where Tier refers to the major portion of the rank described in Section 3, i.e., Bronze, Silver, Gold, Platinum, Diamond, Master, and Challenger, ignoring the 5, 4, 3, 2, or 1 division within a tier, Kills or Deaths or other individual metric is a whole number indicating the number of such occurrences within the game, and Win/Loss is a boolean value indicating whether the team the individual belonged to eventually won the game.

Within each tier, we can compute the percentiles for the metric in question. For specificity, consider a player's raw number of kills. This kills number will be replaced by its percentile, meaning, he did better than what portion of all players in his tier across all games in our dataset. Thus we now have:

$$logit(Win/Loss) = \beta_0 + \beta_1 \quad Tier + \beta_2 \quad Kills$$
 (or other metric) Percentile + ϵ

Now we accumulate average win/loss percentages within each integer percentile (e.g. 3rd percentile, 27th percentile, and so on) to create a graph of the relationship between individual performance of a single full game metric measured as a percentile $\hat{\pi}$ relative to that individual's tier, and the team's win/loss probability. This results in graphs such as the left-hand side of Figure 2.

Worthless Deaths and Smart Kills. Given our in-game win probability model, we can refine the standard metrics of kills and deaths depending on if the team's estimated probability of winning $\hat{\pi}$ increased or decreased following the event.

Specifically, let us call a *worthless death* one that did not increase your team's estimated win probability and a *smart kill* one that did. If you died but took out an enemy inhibitor, that is virtually always "worth." ("Worth," as opposed to "worth it," is a standard term in LoL. Players often type it either in ally chat or all chat to indicate that the value of the objectives achieved outweighed the death itself.) But if you chased a weak enemy for 2 min to kill them, that is almost surely not a smart kill, because it was a waste of time and you likely missed out on an objective.

With these new metrics, we can repeat the process of Section 3.2 to generate graphs such as the right-hand side of Figure 2.

Basic and Advanced Stats. Beyond worthless deaths and smart kills, we can define dozens of new advanced stats. Table 3 lists the basic API stats and the new stats. For space considerations, we discuss only the most important. The bottom panel of Table 3 lists convenient bundles, chosen to be as orthogonal to each other as possible and to correspond to important higher level concepts.

Some of the stats measure activity. This includes the number of attacks, the number of each of the four abilities used, and the damage per minute caused from combinations of attacks and abilities. Others measure your team

Table 3: Basic and advanced league of legends stats (top panel) and bundles of stats (bottom panel).

	· · · · · · · · · · · · · · · · · · ·
Basic stats	Advanced stats
Kills/deaths/assists	Damage taken percent per death
CS/Minute early/mid/ late/full	Carry focus efficiency
Jungle minions killed/ minute	Attacks/minute
Enemy jungle minions killed/minute	Ability counts Q/W/E/R
Gold and gold diff early/ mid/late	Combo damage/minute
Level 6 seconds and diff seconds	Map coverages early/mid/late/full
Wards placed early/mid/ late/full	Useful percent early/mid/late/all
Wards killed early/mid/ late/full	Favorable team fight percent
Total damage dealt/ minute	Favorable team fights count/kills/deaths/ net
Total damage to champs/ minute	Balanced team fights count/kills/deaths/ net
Total damage taken/ minute	Unfavorable team fights count/kills/ deaths/net
Total heal/minute	Reveals per ward average
Average death time percentage	Live wards average yellow/pink/blue
Won	Smart kills/worthless deaths total
	Smart kills/worthless deaths numbers difference
	Smart kills/worthless deaths lacking vision
Early: 0-10 min	Smart kills/worthless deaths health difference
Mid: 10-20 min	Smart kills/worthless deaths gold spent difference
Late: 20 min till end	Smart kills/worthless deaths neutral damage difference
Full/all: entire game	Smart kills/worthless deaths other
Bundle	Component stats
Abilities	Attacks/minute
Gold	Gold early, gold diff early
Laning	Level 6 seconds, CS/minute early, CS/ minute diff early
Survivability	Damage taken percent per death
Time management	Map coverages full, useful percent all
Carry focus	Carry focus efficiency
Vision denial	Wards killed per minute full
Vision	Live wards average yellow/pink/blue, re-
Deaths	veals per ward average Average death time percentage, worthless deaths total per minute
Favorable fights	Favorable team fight percent, smart kills

total per minute

fights. A favorable team fight is one where you and your allies outnumber the enemies. How often did such fights happen, and how many kills and deaths resulted? Similarly for unfavorable team fights where you were outnumbered, and balanced fights.

Smart kills and worthless deaths (both always computed on a per-minute basis) are further broken down depending on why the death occurred. Was it due to being in an unfavorable team fight? Or from a lack of vision? Or a large difference in health or gold spent between the champions in the fight? And so on.

Determining something as fundamental as team fights is a non-trivial problem. First, of course, it requires our new optical tracking data to know the locations of all champions every second. Second, it further requires some judgment and game knowledge to disambiguate team fights. We do this by starting with engagements: an engagement is a continuous sequence of champion-onchampion damage until 5 s elapse with no damage.

Five seconds is consistent with the concept of "incombat" used within the game by certain powerups: for example, Boots of Mobility add a burst of speed after 5 s out of combat.

During engagements, we track the damaged champion and the one who dealt the damage, and include in the team fight any champions within experience range of any champion already deemed to be in the engagement. Thus, engagements get concatenated together as champions join or leave a team fight, either by dying or retreating. Recall that champions earn XP or experience points whenever they are within a small range of an enemy's death, even if they did not participate in the death. Thus, an otherwise uninvolved ally is deemed part of the engagement if and only if they are within XP range.

The distributions of the basic and advanced stats typically exhibit positive skewness. Almost all players fall along a bell curve but the few that had great results had particularly great results.

4 Results and discussion

In comparison to other sports for which optical tracking has become available in recent years, the journey from data to actionable results is shorter and better in eSports, largely because there is more open data and code. In line with that advantage, one of the major contributions of this paper is the open source resources provided.

In basketball, for example, optical tracking data consists of player and ball locations only; even basic actions such as dribbles, passes, and shots need to be inferred,

which is sometimes a straightforward process, and sometimes not. But the more useful actions, such as picks and screens or cuts or post-up position, etc., are notoriously difficult to determine from the raw location data. Progress has certainly been made, but over many years, requiring many contributions, with complicated machine learning models (c.f. Cervone et al. 2014; Maheswaran et al. 2012).

In eSports, on the contrary, we are blessed not only with raw location information, but the actions as well. Who attacked whom, with what ability, for how much damage; who else was nearby; what buffs, levels, items, and offensive and defensive stats did each champion have; where were wards placed and what vision was available; etc. Thus, with our new methodology, we can get to important and actionable information virtually immediately.

4.1 Examples of advanced analytics

Some of the immediately useful information is simply visual presentation of the raw data, while others also require additional computation, such as the in-game win probability. One immediate benefit of our approach is better detection of which champion played which role, a categorization that the official Riot API often fails at. Human inspection of a game rarely leaves any doubt about roles, but the Riot API uses a heuristic based on early game positioning, so, for example, a top that commits early with the bot lane to attempt to score an early kill might be erroneously deemed to be a support. The Appendix shows pseudocode for our improved detection.

Ward Maps and Heat Maps. As a visualization example, Figure 4 shows a ward and a heat map, neither of which are possible with API data. The ward map shows where wards were placed to provide vision, coded to indicate the type of ward and its fate: whether it expired, was replaced or destroyed, or persisted until the end of the game. The heat map shows the main locations a given player occupied on the map. In the case shown, it was clearly a player in the mid lane, who roamed bottom a few times, and to the baron pit, before finally pushing through to the nexus.

Vision. Beyond simply counting or plotting wards, we can also compute the union of all the regions of vision granted by still-living wards using their specific radius at each moment in time for each team. Figure 5 shows this visualization, with the chosen team's vision as a percentage of the total map on top in blue, the enemy team's vision on the bottom in red, the chosen team's net vision as the vellow line, and the chosen individual's vision



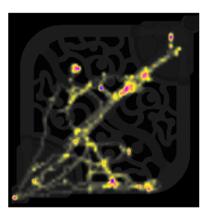


Figure 4: Ward map showing placement and state of wards (left) and heat map showing position (right).

contribution to the chosen team in the middle in green. In this specific game, the chosen team slowly but surely built a vision lead, and the chosen individual's contribution was crucial.

Post-Game Win Probability with Automatic Annotations. It is common in many sports to see graphs of win probability after a game. The large fluctuations especially at ends of games can be dramatic. We further extend this standard type of graph with automatic annotations indicating key states of the game at local minima and maxima.

Figure 6 shows an example. The chosen team got a few early kills (69%), but then had seven net deaths and a lost dragon (35%), a brief mid game bump followed by a long slump of lost towers and lives (17%), before finally rallying with a sequence of good kills and likely an ace. The terminal win probability shown is below 100% because the last partial minute of game play is not included.

Tilt Analysis. A common psychological issue in LoL is the concept of tilt. Borrowing from poker, tilt in LoL refers to situations where a player plays suboptimally because of emotional turmoil. The most extreme form of tilt occurs when a player dies in lane then returns immediately and plays too aggressively and dies again. We can proxy for tilt by calculating the time spent dead by the player over the preceding 3 min and dividing it by the total time spent dead

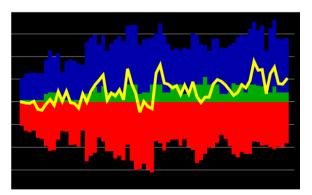


Figure 5: Map vision: team (blue), opponent team (red), net team (yellow), and individual (green).

by anyone on his team over the same period. Three minutes was chosen because death timers range up to around 60 s, and it takes approximately 30 s to walk back to the middle of the map, so "inting" or intentionally dying to the enemy immediately on respawn will quickly show up on this metric. This measure can be calculated from basic Riot API data, and produces the blue chart shown below in Figure 7, starting by convention at 0.2 instead of 0.0 to indicate balance because there were zero individual deaths and zero team deaths. However, this calculation by itself provides little context or explanation. If we further highlight all the player's deaths in red, and note the kinds of team fights they were involved in (which requires the new data), then the context helps to understand and possibly alleviate future examples of tilt. In the example below, notice the player first died in a favorable and then a balanced team fight, before taking three unfavorable fights in a row.

Time Management. Given the location and damage target information, we can partition each player's activity at any given moment into one of several types. An example of a post-game time management analysis is shown in Figure 8. As is common, before minions spawn, most time is essentially wasted, meaning, not engaged in productive activity. Reducing the amount of wasted time during the rest of the game, however, is an important metric. Transition time ought to be minimized. Laning time should also decrease over time, with the possible exception of split pushers.

Individual Performance and Team Win Probability. How does an individual's performance within a game correlate to his team's probability of winning? We previewed earlier in Figure 2 that the relationship between standard kills or deaths with winning is positive but imperfect, similar to the relationship one might expect between, say, an individual's scoring and his team's winning in a traditional sport such as basketball. Filtering individual production for activities that benefit the team, however, has a strong effect.

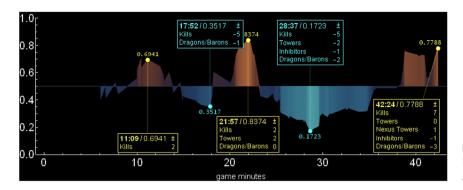


Figure 6: Enhanced post-game win probability analysis showing key turning points and game state.

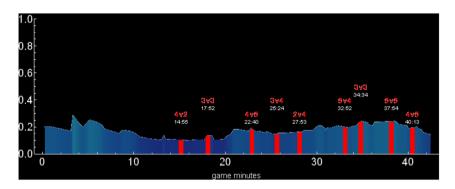


Figure 7: Tilt analysis (portion of recent team deaths responsible) and engagements.

Figure 9 shows the win probability relative to the percentile for various non-overlapping bundles of advanced stats, bundled as shown in the bottom of Table 3, and separated by tier. The OVERALL figure is for the average of all the bundles, as it is a proxy for an individual's overall ingame performance. Averaged across games, it can also represent a player's overall individual rating.

The OVERALL metric does well. It satisfies the main criteria of being a good metric, namely, as the percentile performance on the metric in a game increases, the team's overall win probability also increases, for each tier. However, it is not optimal because it includes several bundles

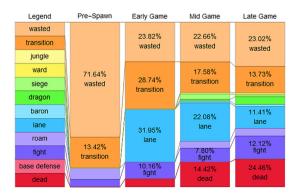


Figure 8: Time management, indicating activities and portion of game phase spent.

that do very poorly at explaining a team's win probability: laning, carry focus, vision denial, and vision. An individual's performance in these four bundles seems to be irrelevant to a team's win probability. An OVERALL metric ignoring these four bundles would do even better.

Some of the bundles that do well are intuitively reasonable—deaths and favorable fights line up perfectly and make sense: don't die, and don't take bad fights. Gold also is increasing but not as much. As we know, gold is good but it is not the whole story. The same is true for abilities and survivability.

Time management is an intriguing one, as it measures an individual's impact across space and time, through map usage and portion of time spent in useful activities. To the extent you had one of the best games in terms of spacetime presence for your rank, you are virtually guaranteed to win.

These graphs, however, only show the isolated effects of each bundle separately. We can further consider the marginal contributions in a complete model with a logistic

Logistic Regression. We fit a standard logit model on the 10 stat bundles percentiles and a constant term for each individual in each game. The estimates $\hat{\beta}_i$ and their corresponding z-statistics in the logistic model $\hat{y} = 1/(1 +$ $e^{-(\beta_0+\beta_1f_1+\beta_2f_2+\ldots+\beta_{10}f_{10})})$ are shown in Table 4 below, for the corresponding f_i that map onto the 10 bundles listed.

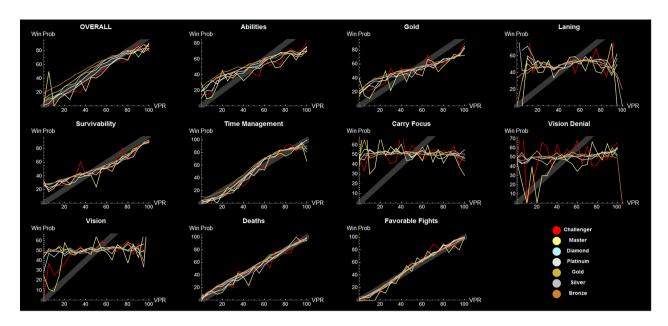


Figure 9: Individual performance in a bundle of stats versus team win probability.

All the estimates are highly significant at the p < 0.0001 level except for Abilities, which has a p-Value of 0.03. As noted in footnote 2, our sample size was approximately 150,000 games. The McFadden (1974) likelihood ratio index or pseudo-R-squared for the regression is 0.42, which represents an excellent fit, as it is even higher than the 0.20 to 0.40 range that McFadden (1978) refers to as "an excellent fit."

The Effect Size column computes the difference in the probability of winning if the respective factor is 55 versus 45% while all the other input factors are fixed at 50%: only Favorable Fights, Deaths, and Time Management are highly sensitive and have a high impact; the others require larger variance to have a strong effect on the win probability.

Table 4: Team win probability regressed on bundles of stats.

	Estimate	z-Statistic	Effect size
(Constant)	0.2132***	4.42	
Favorable fights	0.0378***	99.90	9.42%
Deaths	0.0595***	127.85	-14.78%
Vision	-0.0088***	-23.63	-2.21%
Vision denial	-0.0018***	-8.26	-0.44%
Carry focus	-0.0035***	-13.70	-0.88%
Time management	0.0499***	102.87	12.41%
Survivability	-0.0229***	-55.90	-5.71%
Laning	-0.0023***	-4.68	-0.57%
Gold	0.0058***	19.30	1.46%
Abilities	0.0007*	2.21	0.16%

^{*} significant at the p < 0.05 level.

Surprisingly, several of the bundles have negative marginal contributions, including survivability. Conditional on all the other bundles, it is actually better for you to have less survivability. Similarly with vision, vision denial, carry focus, and laning. There are two legitimate possible explanations for these negative marginal stat bundles. One is that those metrics tend to correspond to a measure of passivity rather than aggressiveness, and aggressiveness tends to win games even if on the margin it makes those stat bundles lower. The other possibility is of multicollinearity among the features.

Table 5 shows the pairwise correlations of each of the features. Very few exhibit substantial multicollinearity. The biggest positive correlation is 0.51 between carry focus and favorable fights. This makes sense because someone who is able in team fights to focus the carry (defined as the enemy's largest damage dealer) is likely also able to avoid fighting in unfavorable conditions. The biggest negative correlation is -0.75 between vision denial and gold. This also makes sense because support players tend to sweep for enemy wards, and they usually earn relatively little gold.

With roles in mind, we can redo the logistic regressions for each of the five standard roles: top lane, jungle, mid lane, attack damage carry ("adc"), and support. Table 6 lists the z-Statistic for the relevant coefficient in a logistic regression of player-games restricted to the given role.

Certain bundles are highly significant across all roles: favorable fights, deaths, vision, time management, and survivability. Carry focus is significant for all roles except the top lane; the top laner often split pushes and does not

^{***} significant at the p < 0.0001 level.

Table 5:	Pairwise	correlations	among the	features	(bundles of stats).

	Fav. fights	Deaths	Vision	Vision denial	Carry focus	Time mgmt	Surviv- ability	Laning	Gold
Deaths	0.28								
Vision	0.12	0.28							
Vision denial	0.23	0.20	0.03						
Carry focus	0.51	0.32	0.15	0.36					
Time management	0.10	0.12	0.04	0.04	0.10				
Survivability	0.06	0.04	0.04	0.09	0.14	0.01			
Laning	0.05	0.09	0.05	0.08	0.20	0.02	0.26		
Gold	-0.21	-0.19	-0.03	-0.75	-0.28	0.04	-0.03	-0.01	
Abilities	0.20	0.29	0.02	0.27	0.45	0.08	0.05	0.06	-0.29

Table 6: Role-specific logistic regression z-statistics across the various bundles of stats.

	Тор	Jungle	Mid	ADC	Support
(Constant)	-2	+4	-2	-6	+16
Favorable Fights	+42	+38	+48	+50	+43
Deaths	+58	+57	+54	+52	+61
Vision	-10	-10	-11	-10	-12
Vision denial	-6	-3	-4	-4	-2
Carry focus	-2	-4	-8	-9	-8
Time management	+47	+52	+42	+46	+45
Survivability	-24	-28	-24	-23	-29
Laning	+2	-5	+1	-1	-6
Gold	+6	+7	+12	+7	+3
Abilities	-2	+4	-2	-6	+16
McFadden R ²	0.39	0.49	0.40	0.43	0.43

join team fights, so he ends up focusing whoever comes to stop him rather than the enemy carry. Laning is important for laners (top, mid, and adc), and not important for junglers and supports, while abilities are the opposite.

For completeness, we can see the separate correlations of each bundle with win probability, by role. Table 7 shows that the top factors are as expected and relatively constant across roles. The implied advice to be a better player is: fight good fights, don't die, and don't waste time.

Fast and Frugal Trees. We compute a fast and frugal tree to see the best way to predict your team's performance given only the most important details about an individual's performance. Fast and frugal trees (described in general as well as the implementation used here in Phillips 2016) are decision rules that are both easy-to-use and make good predictions. We train on all the metrics of Table 3, with 25% of the data, about 35,000 observations, reserved for crossvalidation. Figure 10 shows the best resulting tree. It makes the correct prediction about 80% of the time despite only using three metrics.

Table 7: Role-specific isolated correlations of bundles of stats with win probability.

	Тор	Jungle	Mid	ADC	Support
Favorable Fights	0.4347	0.5067	0.4890	0.5015	0.5007
Deaths	0.4858	0.5474	0.4970	0.4907	0.4976
Vision	0.0499	-0.0082	0.0218	0.0028	0.0177
Vision denial	0.0251	0.0092	0.0224	0.0297	0.0394
Carry focus	0.0254	-0.0178	0.0186	0.0595	-0.0608
Time management	0.4850	0.4929	0.4875	0.5396	0.4565
Survivability	0.3447	0.3428	0.3489	0.3796	0.3360
Laning	0.1264	-0.0514	0.1338	0.1246	-0.0459
Gold	0.2395	0.2615	0.2652	0.2802	0.2396
Abilities	0.2347	0.3011	0.2556	0.3170	0.2744

The final decision rule is simple indeed: as a rule of thumb, if your bad death percentile was below average (meaning you had more worthless deaths than players at your rank typically do in games), then you likely lost. If, on the other hand, you didn't die that much, then, if you had more smart kills than your peers, you likely won. Finally, if you had both above average worthless deaths and below average smart kills, then whether your team won or not depends on how much influence you had over the map. If you had more map coverage then average, you probably won; otherwise, you probably lost. Thus, the fast and frugal tree (FFT) clarifies our story: don't die for no reason; if you do, try for some smart kills; if you can't do that either, at least roam and have decent map coverage.

Note also that the FFT was free to choose any decision tree using any of the metrics in Table 3, including both the basic and advanced stats, and it ultimately chose to use only the advanced stats, and exactly the ones we had identified in the previous section as having a consistent impact on win probability across all roles. Note also that the overall FFT accuracy is 80% with a false positive rate of only 13%.

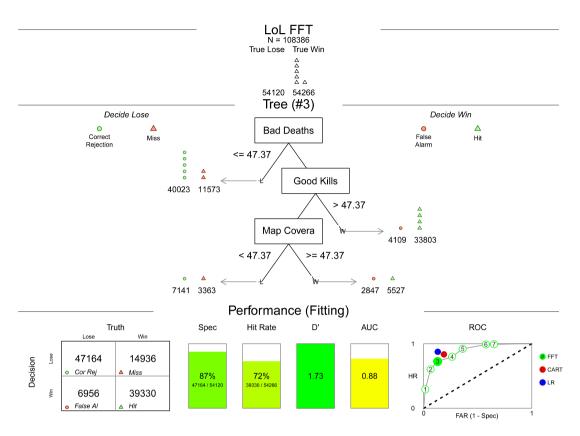


Figure 10: Fast and frugal tree analysis for individual percentile performance to predict team victory.

5 Conclusion

Using a combination of computer vision, dynamic client hooks, machine learning, visualization, logistic regression, large-scale cloud computing, and FFTs, we generated new and unique data on millions of LoL games, calibrated a win probability model, developed enhanced definitions for standard metrics, presented automated improvement analysis, provided a framework for determining an individual's contribution to a team's victory, and applied that framework to show that the advanced stats both better correlate with and explain team outcomes.

How much does an individual contribute to his team's success? This is a touchstone question of sports analytics for all sports. Here, we find that standard metrics of performance are insufficient to explain team success in eSports. We also find that an intelligent adjustment to those standard metrics does indeed explain team success. For LoL, that adjustment required a substantial investment in original data collection, storage, and analysis across millions of games, but the final insight is intuitive: when standard individual actions of performance within a game are filtered for broader impact, the new adjusted summary statistics

correlate almost perfectly to team performance. In other words, one athlete's "points" do not necessarily correlate with victory. Indeed, it can often be negatively correlated. However, if we use the insights from here, we can consider *smart* points, ones that not only padded the individual's statistics, but also significantly contributed in the moment to an increase in the win probability of the team.

All of the code is open sourced. This includes the code for extracting data from spectated games, all of our analytical models, and all of our visualizations shown in this paper.

Future research directions include developing an alternative data collection approach involving interpreting the network traffic directly without the need for the game client itself, porting the analytics approaches to other games, evaluating more micro outcomes rather than team victory, and exploring the dynamic changes that occur as patches from the publisher continue to change the rules of the game.

Acknowledgments: We are grateful to our professional eSports team partners who have used our data and analytics in their own processes and to the members of our

amateur eSports league for their suggestions and feedback. And I thank Brett McDonald, Chase Exon, Cameron Tangney, Kevin Lee, Brandon Sedgwick, Scott Snider, Sheridan Zalewski, Mark Jansen, Nick Kastner, and Lucas Nash for their contributions. I am also grateful for the very useful and helpful comments of participants at the 12th Annual MIT Sloan Sports Analytics Conference and the editor and anonymous referees of the Journal of Quantitative Analysis in Sports.

Author contribution: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: None declared.

Conflict of interest statement: The author declares no conflicts of interest regarding this article.

Appendix 1

Additional description of league of legends

Champions can either perform a basic attack, the equivalent of a punch or a sword swipe or throwing a dart, or they can cast one of their abilities, or both. Doing them in a specific order can often be more effective: for instance, if one of your abilities slows or stuns an enemy, then casting that first will make it easier for you to land a skillshot with greater damage. Abilities and certain items have cooldowns that prevent you from "spamming" them constantly. You typically have to wait on the order of a few seconds before you can cast an ability again. Ultimate abilities tend to have much longer cooldowns, because they are so much more powerful. Abilities can cost "mana," a synonym for magic ability, or "energy," or another resource. If your ability is off cooldown, and hence available to be cast, but you lack the mana it costs to cast it, then you will be unable to cast it.

As a player, you also have two summoner spells that have massive cooldowns on the order of several minutes but can provide massive advantages, such as the ability to teleport to a distant location, flash instantly to a nearby one, or gain an enormous increase in speed. The jungler typically takes the summoner spell called "smite," which instantly causes a large amount of damage to minions. This summoner spell also allows junglers to buy certain jungle-specific items that are otherwise unavailable. In principle, any champion could take smite and also buy those items; however, the game has evolved to make that typically sub-optimal.

When starting a game, you choose one of over a hundred available champions to control in this environment, and your teammates do as well. You also

have the option during the initial champion selection phase to ban certain champions that you do not want the enemy to play against you. Champions are constantly refined, and new ones are introduced, and each champion usually has several other champions who would make very difficult opponents, so most players tend to ban some combination of commonly picked champions and socalled "hard counters." For example, a melee champion with a short attack range may choose to ban a ranged champion who otherwise would be able to simply poke at the melee champion from a safe distance.

Game play usually progresses in two primary phases: laning phase, and post-laning phase. Laning phase means the players stay in their own lanes, for the most part, trying to earn as much gold from killing enemy minions while denying the same to their opponent, or killing the opponent. Killing an enemy champion, or a turret, also grants gold. Laning phase is typically considered over after the first turret falls. In the post-laning phase of the game, a team can choose to group up into a particular lane to siege an enemy turret and gain a numbers advantage. Or a team can have one champion on a side lane, so called "split pushing," while the remaining four distract the enemy's

There are thus two aspects to game play: micro and macro. Micro refers to knowing your specific champion's abilities and skills and combinations and being able to execute them correctly. In basketball terms, this would be like knowing how to dribble, pass, shoot, box out, and defend. The macro part of game play is more strategic and involves positioning both yourself and your teammates through an in-game chat to most effectively control the map and win the game.

Each team can only see what is visible to one of its champions or minions or turrets, so there is a fog of war on most areas of the map where you do not know where your opponent might be. Junglers are especially sensitive to this fog of war, trying to use it to catch an enemy by surprise. Various kinds of wards are available as items that either grant or deny vision, and some champions can become invisible for a time. Various brushes also represent hiding locations that can't be seen unless you are also in them or they are warded. At higher levels of play, vision is often the driving force behind victories.

Specifically, the enemy jungler is often not visible to you. Thus, when you are in laning phase, if you have been aggressively kililng the enemy minions, you may find yourself more than halfway between the safety of your turret and the danger of the enemy's turret, which can attack you if you are within range. If the enemy jungler emerges from the fog of war at this time, you will find yourself in a one-vs-two situation, and probably die. This activity of having an enemy teammate appear out of the fog of war to create a numbers advantage is referred to as a "gank."

Appendix 2

Pseudocode for determining a champion's role

The below presents pseudocode for determining the role of each champion, for each team. LoL is a flexible game and any champion can in principle go anywhere on the map. However, over time, certain kinds of roles have evolved and become relatively fixed in the community: a top laner, a mid laner, a jungler, and two bottom laners, one of whom supports the other (called the attack-damage carry, or ADC, or simply carry). These roles primarily refer to the laning phase of the game, after which champions frequently roam or group in other parts of the map.

The Riot API reports what Riot assumes are the roles of each person. Their results are sometimes incorrect, particularly if a jungler forgot to bring the Smite summoner spell, or if the bottom champions switched with the top laner for the first few minutes, or if players assigned roles during the game setup phase agreed to swap but then roamed the wrong lanes, or other unusual but still occasionally occurring cases. Using our new data, we have found the following algorithm to work in all cases that we have seen, from amateur play to professional play.

Pseudocode: automatic categorization of champion roles

Early Minions: Define early minions killed as those minions killed by a champion who at the time of killing the minion was between level two and level six.

For each team:

- The jungler is the champion who killed the most early minions that were neutral jungle minions.
- Of the remaining four champions: the *support* is the one who killed the fewest overall early minions.
- Of the remaining three champions: the mid is the one who killed the most early minions near the map's center.
- Of the remaining two champions: the ADC is the one who was most frequently near the support during the first 10 minutes of gameplay, where near is defined as being within one-10th of the width or height of the map.
- The remaining champion is the *top*.

References

- Cavadenti, O., V. Codocedo, I-F. Boulicaut, and M. Kaytoue, 2016. "What Did I Do Wrong in My MOBA Game? Mining Patterns Discriminating Deviant Behaviours." In Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on. IEEE, 662-71.
- Cervone, D., A. D'Amour, L. Bornn, and K. Goldsberry. 2014. "POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data." In MIT Sloan Sports Analytics Conference. Cambridge, MA: 42 Analytics.
- Deja, D., and M. Myślak. 2015. "Topological Clues for Predicting Outcomes of Multiplayer Online Battle Arena Games." In IADIS International Conference Interfaces and Human Computer Interaction, July 2015.
- do NascimentoJunior, F. F., A. S. da Costa Melo, I. B. da Costa, and L. B. Marinho. 2017. "Profiling Successful Team Behaviors in League of Legends." In Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web. 261-8.
- Drachen, A., M. Yancey, J. Maguire, D. Chu, I. Y. Wang, T. Mahlmann, M. Schubert, and D. Klabjan. 2014. "Skill-Based Differences in Spatio-Temporal Team Behavior in Defence of the Ancients 2." IEEE Games Media Entertainment.
- Duch, J., J. S. Waitzman, and L. A Amaral. 2010. "Quantifying the Performance of Individual Players in a Team Activity." PloS One 5 (6), e10937.
- Eaton, J. A., M-D. D. Sangster, M. Renaud, D. J. Mendonca, and W. D. Gray. 2017. "Carrying the Team: The Importance of One Player's Survival for Team Success in League of Legends." Proceedings of the Human Factors and Ergonomics Society 2017 Annual Meeting 61 (1): 272-6.
- Hodge, V., S. Devlin, N. Sephton, F. Block, A. Drachen, and R. Cowling. 2017. "Win Prediction in Esports: Mixed-Rank Match Prediction in Multi-Player Online Battle Arena Games." arXiv preprint arXiv: 1711.06498.
- Keiper, M. C., R. D. Manning, S. Jenny, T. Olrich, and C. Croft. 2017. "No Reason to LoL at LoL: The Addition of Esports to Intercollegiate Athletic Departments." Journal for the Study of Sports and Athletes in Education 11 (2): 143-60.
- Lee, C-S., and I. Ramler. 2017. "Identifying and Evaluating Successful Non-meta Strategies in League of Legends." In Proceedings of the 12th International Conference on the Foundations of Digital Games.
- Lock, D., and D. Nettleton. 2015. "Using Random Forests to Estimate Win Probability before Each Play of an NFL Game." Journal of Quantitative Analysis in Sports 10 (2): 197-205.
- Maheswaran, R., Y. H. Chang, A. Henehan, and S. Danesis. 2012. "Deconstructing the Rebound with Optical Tracking Data." In MIT Sloan Sports Analytics Conference.
- Matz, D. 2017. "Winston's Lab." Retrieved from https://www. winstonslab.com.
- McFadden, D. L. 1974. "Conditional Logit Analysis of Qualitative Choice Behavior." In Frontiers in Econometrics, edited by P. Zarembka, 105-42. New York: Wilev.
- McFadden, D. L. 1978. "Quantitative Methods for Analyzing Travel Behaviour of Individuals: Some Recent Developments." In Behavioural Travel Modelling, edited by D. Hensher and Stopher, 279-318. London: Croom Helm London.
- Paul, J. 2017. "By the Numbers: Most Popular Online Games Right Now." Also available at https://nowloading.co/posts/3916216.

- Phillips, N. D. 2016. "Making Fast, Good Decisions with the FFTrees R Package." Also available at https://nathanieldphillips.com/ 2016/08/making-fast-good-decisions-with-the-fftrees-r-package.
- Quitko, F. 2017. "Crauzer/LeagueHack/EloBuddy." Also available at https://github.com/Crauzer/LeagueHack.
- Rioult, F., J-P. Métivier, B. Helleu, N. Scelles, and C. Durand. 2014. "Mining Tracks of Competitive Video Games." In AASRI Conference on Sports Engineering and Computer Science, vol. 8, 82-7.
- Riot Games. 2018. "Riot Developer Portal." Also available at https://developer.riotgames.com.
- Schubert, M., A. Drachen, and T. Mahlmann. 2016. "Esports Analytics Through Encounter detection." In 2016 MIT Sloan Sports Analytics Conference.
- Wolfram Research. 2017. "Logistic Regression (Machine Learning Method)." Also available at https://reference.wolfram.com/ language/ref/method/LogisticRegression.html.
- Xia, B., H. Wang, and R. Zhou. 2017. "What Contributes to Success in MOBA Games? An Empirical Study of Defense of the Ancients 2?" Games and Culture. 14 (5): 498-522.
- Yang, P., B. E. Harrison, and D. L. Roberts. 2014. "Identifying Patterns in Combat that are Predictive of Success in MOBA Games." In Proceedings of Foundations of Digital Games.
- Yang, Y., T. Qin, and Y-H. Lei. 2016. "Real-Time eSports Match Result Prediction." Also available at https://arxiv.org/abs/1701. 03162.