Violence on the Rift: Kills as a Point Process in League of Legends across Space and Time

Sammy Rosofsky

1 Introduction

League of Legends, developed by studio Riot Games, is a competitive video game where two teams of 5 battle each other with the goal of destroying the enemy team's base. This game takes place on a 2-dimensional map, known as Summoner's Rift, shown in figure 1:

Areas of note are labeled as follows:

- 1-2: These are the team bases, with the furthest object being the nexus. A team wins when they destroy the enemy team's nexus.

- 3-5: These are lanes, respectively called top, middle, and bottom (or bot for short). These lanes contain towers for each team leading to the base. Computer controlled units periodically stream down each lane. Most action happens in the lanes as the towers must be destroyed before getting to the base.

- 6-9: These four segments make up the jungle. In the jungle are neutral computer controlled units (either team can fight them).

Figure 1: Summoner's Rift

- 10-11: This is the river. The river is the fastest way to move between lanes and also contain neutral objectives in the form of dragons and Baron Nashor. Slaying these neutral objects give powerful bonuses to the team.

Throughout the game, players gain gold through killing enemy players, killing computer controlled units, and destroying enemy structures. The killing of enemy players tends to be the most exciting moments of a game, both intrinsically and extrinsically. The former is due to the feeling of directly triumphing over your opponent, and the latter from how player kills give the largest bounties of gold. We can treat where and when kills happen as a random process, and then ask the question of whether the spatial and temporal distribution of kills satisfy Complete Spatial Randomness, where there would be no clustering of the location and times of kills.

2 Description of Data

The data for this analysis is pulled from Riot's League of Legends API, which is available for anyone that signs up for a Riot account. Through the API we can pull a match timeline object. This object contains "events", which are virtually everything a player does in game, which includes kills. Importantly, every event has x,y coordinates, where these coordinates use an arbitrary unit that fits within the game's map, as well as a timestamp, measured as hundredths of a second counting from the start of the match.

For the purpose of this analysis, I am using a sample of matches from the 20 most recently played matches of 50 highest ranked accounts on the North American server as of November 16$^{\text{th}}$, 2022. Because players are matched with other players, there exists some overlap between the matches associated with each account, and only unique matches are used. It should be noted that some aspects of the API make it nearly impossible to draw a truly random sample of matches. First, when using a personal API key (as opposed to a production API key), we are limited to only 50 requests per minute. Second, match timelines can only be referenced through a match ID, which itself can only be referenced through an account ID. There are millions of accounts just on the North American server, and each account can have upwards of hundreds of matches played in a given year, so it would be infeasible to make a process to make a truly random sample of matches with only 50 API requests per minute.

By taking a sample of matches from the highest ranked accounts, we can get an approximation of what one would expect if observing a match played at the highest level. However, some players play more often than others, and

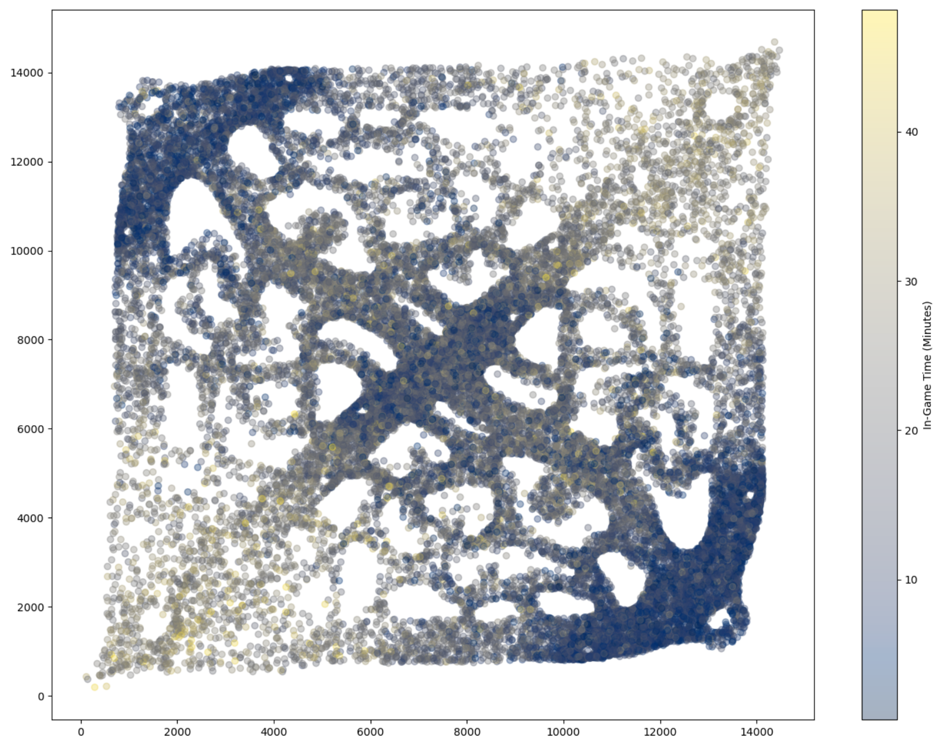

Figure 2: Location and Time of kills

each player has a unique style of player, so this method of sampling does introduce some bias.

The data set consists of 551 matches, which in total have 29,132 kills.

From figure 2, we can already observe some trends in the data. In the early game (colored more blue), kills are concentrated in the middle of each lane. This is due to computer controlled units periodically streaming through the lanes for each team. These units serve as another source of gold, and much of the early game's action revolved around trying to kill the enemy's units while preventing them from killing your own units, with this action eventually resulting in kills. Each team has towers that protect each lane, so players can only progress so far to the opposite side of the map in the early game before being stopped by a tower. Throughout the game, as players spend gold to become stronger, they can more easily destroy these towers, so as the game progresses the action spreads out further down the lane.

Another factor is that as the match progresses, neutral objectives, which are located in the river that runs between the lanes, offer rewards that become more powerful over time. These rewards are often what tips the balance of power in favor to the team that destroys that objective enough for them to end the match. Because of the greater importance on these objectives the longer the match lasts, we can see more action taking place in and around the river later in the game.

3 Method

For analyzing this point process of the location and time of kills, we will look at whether the distribution is Complete Spatial Randomness (CSR). This is when the observed points are evenly distributed throughout the sample space. We will approach this question by looking at 1^{st} order and 2^{nd} order properties.

3.1 1^{st} Order Properties

The 1^{st} order properties of a point process are the frequency of points in a given area of the sample space. If the point process is CSR, we would expect each area to have about the same number of points. To test this, we will divide the data set into a 3-dimensional grid with the x and y coordinates of each point being the x and y axes, and time being the z axis. With the data set divided, we will then count the number of points in each subregion and perform a chi-squared test.

For reference, a chi-squared test uses the following formula:

$$\chi^2 = \sum_{i}^{n} \frac{(p_i - \overline{p})^2}{\overline{p}} \tag{1}$$

Where, in this case, n is the number of regions, p_i is the number of points in the ith region, and $\overline{p}$ is the expected number of points in a region under CSR. We then find the probability of this sum being the output of a χ^2 distribution with k degrees of freedom, with:

$$k = (x-1)(y-1)(z-1) \tag{2}$$

Where x, y, z are the number of divisions of the grid across each axis.

An important aspect of conducting this chi-square test is choosing how much to divide the space. If the subregions are too large, there may not be enough granularity to see a proper difference between them, and if they're too small there may not be enough points in each subregion for a proper test. Because of this concern, multiple choices for number of divisions will be used and compared

3.2 2^{nd} Order Properties

For 2^{nd} order properties, we want to know how closely related points are to other points based on distance. For this, we will be using a nearest neighbor analysis the G-function. This involves two steps:

1. For each point, calculate the distance between the point and the point closest to it.

2. Use the G-function to find the proportion of points that have a nearest neighbor closer than various distances.

It should be noted that "distance" for this data is not exactly well-defined because it involves measurements of different units, i.e. space and time. For example, points a and b can have the same space and a distance of x based on time, and points a and c can have the same time and a distance of x based on space, and there isn't a real world interpretation for how these distances would be equal. However, for the purpose of this analysis this is not that important because the purpose is to test against an assumption of CSR, so any kind of clustering is what we're concerned with regardless of how the clustering is with regard to space specifically or time specifically.

The G-function is defined as:

$$G(r) = \frac{\#[r_{\min}(p_i) < r]}{n} \tag{3}$$

Where p_i are the points, n is the number of points, and r is a given distance. We can then plot the G-function for various distances. To see if this curve is significant, we will then simulate 100 point processes in the sample space under CSR and plot the 95^{th} and 5^{th} percentiles for $G(r)$ among these simulations.

4 Results

4.1 1^{st} Order Properties

Table 1 shows choices m for number of divisions, where each m represents an $m \times m \times m$ grid.

m	2	3	4	5	6	7	8	9
$\overline{p}$	3641.5	1078.9	455.1	158.3	59.2	23.35	9.5	4.2
χ^2	29132.0	29132.0	29125.0	19795.0	12780.9	7994.9	4851.0	3059.0
P-value	0	0	0	0	0	0	0	0

Table 1: Table with various choices for # of divisions

From Table 1, we can see that regardless of our choice for m, the probability of the distribution of the data set being observed under CSR is almost certainly 0.

4.2 2^{nd} Order Properties

Under CSR, we would expect to see intervals of small increases followed by an interval with a steep increase because the expectation is that points all have about the same distance to their nearest neighbor. In figure 3, we see a rapid increase throughout the distances up until the last few outlier distances are reached.

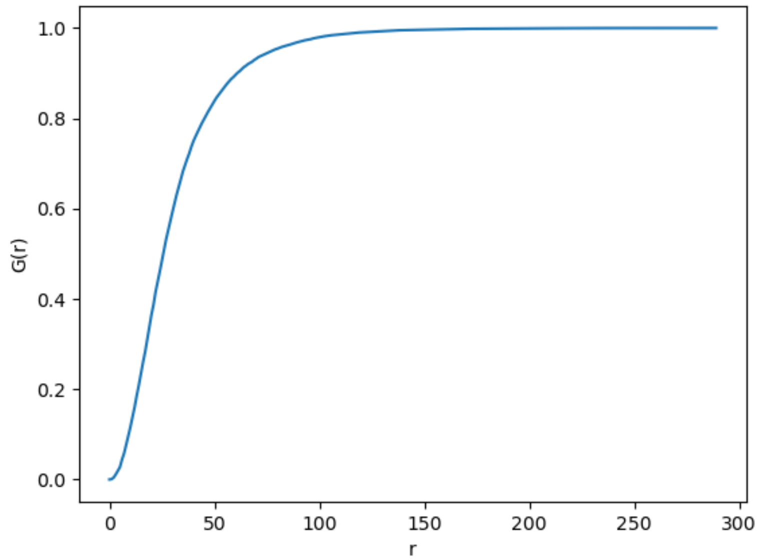

Figure 3: Proportion of points with distances to nearest-neighbor below given r

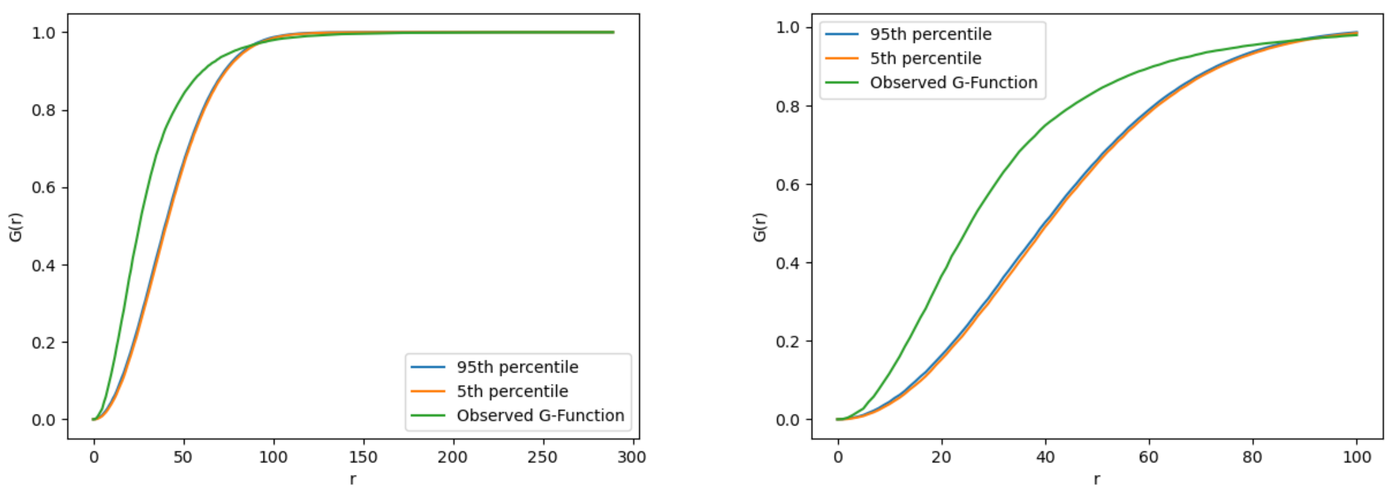

Figure 4: Comparison of observed G-curve and simulations

From figure 4, When looking across the entire range of r the 95$^{\text{th}}$ and 5$^{\text{th}}$ percentiles are practically overlapping compared to the difference between the observed $G(r)$ and the 95$^{\text{th}}$ percentile. Truncating the range down to 100 provides a better sense of the scale between these curves.

5 Discussion

When testing both 1$^{\text{st}}$ order and 2$^{\text{nd}}$ properties, we see that the probabilities of the observed data following Complete Spatial Randomness to be virtually nil. This would line up with observations made when looking at the distribution of kills across space and time. One of the basic concepts of the game is a separation between “early”,“mid”, and “late” game, where team’s are focused on different parts of the map. The early game, also known as “laning phase”, has players focused on fighting over computer-controlled units in the middle of each lane, the mid game involves players starting to move around the map more, and the late game is where teams focus on making decisive plays to end the game after getting strong enough.

From observing the distribution of kills across space and time, we can see this concept happen, and through significance tests we can confirm that a match follows some structure where action takes place at different locations across time rather than being constant throughout a match.