League of Legends

Data Visualization
SM-DV - autumn 2022
Group 12

Kristian Østergaard
krust19@student.sdu.dk

Oliver Winther
olvan18@student.sdu.dk

Peter Andreas Brændgaard
pebra18@student.sdu.dk

Troels Zink Kristensen
tkris17@student.sdu.dk

Deadline for submission: December 18th, 2022.
Dashboard: Shinyapps.io dashboard

Abstract

This report provides a look into League of Legends (LoL) match statistics. It provides data visualizations based on a dataset with more than 2500 different matches. The dataset is collected from the Riot API [5]. The goal is to determine which variables are most important to increase a player's win rate.

To do so 15 out of 100+ variables were chosen and examined. These were variables such as champion names, deaths, vision score, minions killed, etc. From them, graphs of different types were created. One density plot, two line plots, two scatter plots, two violin plots, two box plots, six bar charts and one animation plot were created to properly investigate and visualize the data. It was done in the programming language R.

The visualizations showed that a higher vision score leads to more victories, especially if a team's vision score is higher than their opponents. At the same time, simply looking at wards placed and killed does not show a larger difference between winning and losing teams.

The data visualizations also give an overview for picking the best champion for a certain position or against specific opponents. For picking a specific opponent, it is possible to pick the opponent and their position, allowing the visualization to guide players to finding the best champions to pick.

An animation is created to showcase the variables collected, and how they change depending on the game duration. From the animation, all variables were higher for the winning team than for the losing team, except deaths, which were higher for the losing team. Although all the variables showed a distinction between winning and losing, the variable that showed the biggest difference was the vision-related variables. All the graphs and results can be found in the shinyapps.io dashboard (see reference: [1]).

Table of Contents

1 Background and Motivation

League of Legends (LoL) is a competitive multiplayer game, where two teams of five players battle for victory. Each player on a team fulfills one of five positions (also called lanes); top, jungle, middle, bottom, and support/utility. Support and utility are used interchangeably. Each player decides which champion they want to play. Champions have different abilities and statistics to provide variety to each game. The goal of the game is to destroy the opponent's nexus (a base). To help achieve this goal, each team have minions which spawn every 30 seconds. Each team also has 11 turrets which are static defense mechanisms that defend the nexus. Players acquire gold by killing minions, champions and turrets.

It is a very popular game with an average player count of 150 million throughout the last 30 days [4]. LoL is a major source of revenue for Riot Games, the creator, but also for players. It is big in e-sports and streaming services with big income potentials for top competitors and streamers [3]. It is therefore no surprise that many players are interested in developing their skills and achieving a competitive edge to increase the odds of ranking among the very best. Furthermore, even players without top ambitions still play the game hoping to become better. The primary attraction in the game is 'Ranked', where players are assigned a specific rank. If they win more games than they lose, they will climb the ranks. This is the primary incentive behind the game.
These two factors combined: 1) the earning potential and 2) the competitive nature, are the motivation behind creating a data visualization project that can provide players with insights giving a competitive advantage over their competitors.

2 Project Objectives

As the overall goal of the game is to win, it would be most logical to explore factors/variables that have an influence on victories. The *response variable / dependent variable* is thereby "win"; a boolean value, which is the most simple type of regular categorical/nominal variables. The overall question that will be explored is:

Which variables are most important to examine in order to maximize the probability to win a game in LoL?

It has been decided to focus on three different areas that might have an effect on victories. Each focus area will have multiple independent variables that can be explored. Some of these are listed below. The variables are explained in Table 1 in section 3.

1. How does map control affect the probability of a victory?

    - Vision score.
    - Wards placed and wards killed.

2. How does the chosen champion play a role in a victory?

    - Team position, champion name.
    - Kills, assists, deaths.
    - Exploring a champion's win rate against specific champions within the same team position.

3. How does each variable change over time, and what trends can be acquired from this?

    - Game duration.
    - Relevant variables: assists, deaths, gold earned, kills, total damage dealt, total minions killed, vision score, wards killed, wards placed.

3 Data

This section describes the dataset that is utilized to answer the project objectives, and thereby the questions.

3.1 Source

The data has been collected from the Riot API, specifically, MATCH-V5 [5].

3.2 Description

The source contains all recorded LoL matches. Each match results in 10 records, one record for each player, and given that millions of games are played each day, the dataset quickly becomes massive. One record contains over 100+ variables. Some of these are unique to the player and dependent on their performance, such as damage dealt, minions killed, vision score etc. Other variables are unique to the team in the match, such as win/lose, objectives killed etc. It was not possible to analyze all the variables, which led to 15 of them have being picked. These variables are explained in Table 1 in an effort to give the reader a solid understanding of the variables' influence in the game.

Variable	Variable Description	Variable Type	Uniqueness
Game duration	Game length in seconds	Discrete	Match
Kills	Number of kills	Discrete	Player
Champion name	The player's chosen champion	Nominal	Player
Deaths	Number of deaths	Discrete	Player
Gold earned	Total gold earned	Discrete	Player
Assists	Number of kills the player has helped another player	Discrete	Player
Team id	The side the player is playing on	Nominal	Team
Team position	The position on the map the player is playing on (also called lane)	Nominal	Player
Damage dealt	Amount of damage dealt to opponent champions	Continous	Player
Vision score	A calculation on the number of wards placed, and wards killed and how many opponents they have detected	Discrete	Player
Wards killed	Number of wards killed	Discrete	Player
Wards placed	Number of wards placed	Discrete	Player
Win	Whether or not the player won	Nominal (0,1)	Team
Game id	Unique id given for the match	Discrete	Match
Minions killed	Number of minions killed	Discrete	Player

Table 1: *Variable overview.*

3.3 Processing

Riot only allows for requesting data concerning one match at a time. It is also not possible to get a list of all match ids. Therefore, one needs to go through a series of steps to get multiple matches. These are:

1. Find a match.
2. Collect data on match.
3. Save player ids of players in match to a list.

4. Request match history of player ids.
5. Save match ids in a list.
6. Repeat step 1-5.

Following these steps will yield an ever growing list of both player ids and match ids. It was decided to stop collecting more data after 2500 matches.
The Riot API returns match data in JSON format. To make the data easier to work with, the desired variables were extracted from this JSON structure to a CSV format.

4 Visualization and Dashboard

4.1 Design

For the visualizations, the variable that will be compared in most of the graphs is the *win* variable (the response variable), which is of boolean type. Therefore, many of the visualizations will include this value as a second or third variable. The visualizations will generally show how the variables differ depending on if the game was won or lost and try to visualize, which variables affect the response variable the most. Since the most prioritized variable is a boolean, the graphs will use a colour palet to show victories and defeats. These colours will be light red for the games lost and cyan blue for the games won. For the remaining graphs, the colour palette will be black on white background, since they will be generated with two numerical values.

4.2 Must-Have Features

The must-have features are features that are important for the project. Furthermore, there are some must-have requirements stated by the project guidelines, which are listed below:

- Minimum three types of graphs.
- At least one animation graph.
- Minimum eight graphs in total.

These requirements should be considered when designing and implementing the dashboard to fully accommodate the project scope and objectives. The chosen graphs will be explained below by reference to the sub-questions defined in section 2.

How does map control affect the probability of a victory?

Map control within LoL can be assessed in several ways using multiple variables. For this project, the focus has been on three variables that are interconnected and important for

map control. These are vision score, wards placed, and wards killed.

A density plot is used to visualize the density of the vision scores for each game and then comparing this to the response variable: *win*. Box plots and violin plots are created to show which team positions that have the highest number of wards placed and wards killed. Violin plots are especially well-fitted for the distribution of the data, showing both the means and variances. It is also interesting to determine how the vision score difference between the two teams impacts the win rate; this has been done with a line plot. All these plots will be altered when selecting or deselecting one or more of the five team positions available.

How does the chosen champion play a role in a victory?

The champion a player chooses in LoL is an important factor in the game. The player should of course be able to play that specific champion, but also choose the correct one against the chosen champion within the same team position on the other team. One team has to choose the champion before the other team; however, if lucky to be number two, then it would be important to know which champion to pick based on that. The variables to be analyzed here are: team position, champion name, and win. The analysis will not be based on direct contact between the two champions, but merely on whether the entire team won the game.

Within the dashboard, it is possible to pick a specific team position together with a champion. A bar chart has been used to visualize the chosen champion's win rate against other champions within the specified team position. Furthermore, one bar chart is created per team position to show the champions with the highest win rates in those positions.

How does each variable change over time, and what trends can be acquired from this?

In order to fully understand how variables affect the win rate, an animation has been created to visualize how the variables change over time (game duration). The variables that can be chosen are: assists, deaths, gold earned, kills, total damage dealt, total minions killed, vision score, wards placed, and wards placed. A line plot is created for both victory and defeat. As the game length varies a lot for each game, time intervals have been created. The user can choose intervals of 60 to 600 seconds with a step value of 30 seconds. The larger the length of the interval, the more smooth the lines will be.

In total, these types of plots will be generated, including how many of each type:

- Density plots: 1
- Line plots: 2
- Scatter plots: 2
- Violin plots: 2
- Box plots: 2
- Bar charts: 6
- Animation plots: 1

In total, there will be generated six different graph types, one animation graph, and 16 graphs in total; this fulfills the three aforementioned requirements described in the project guidelines.

Downloads

There are two additional must-have requirements from the project guidelines, which are a link to the dashboard ([1]), and an option to download the report from the dashboard. It is possible to download multiple documents from the dashboard. Within the "Introduction" page, the project guidelines, the dataset, and the report can be downloaded.

4.3 Optional Features

All graphs are simple and easy to understand. The response variable, win, is a boolean value, which results in a rather simple colour palette; blue for a victory (win = true), and red for a defeat (win = false). These two colours are well differentiated, which also means that people with colour blindness are able to differentiate the two colours as well. The most typical difficult colour combinations for people with colour blindness are red-green and blue-yellow [2], meaning that red-blue is a good choice of colour combination. However, another colour combination could have been used that would be just as good.

The animation graph is by default a Graphics Interchange Format (GIF) that has a certain length and a specific number of frames per second. It is not possible to stop or skip through the animation. A nice, but optional, feature is a video instead of a GIF, where it is possible to control the animation.

5 Story and Results

This section will explain the results of the implemented visualizations to fully answer the questions made in section 2. There will be references towards graphs within the dashboard; these will be referenced by the keyword: *graph/plot x.y.*

5.1 Vision

Before looking at the data, one would expect that generally having better scores in vision-related variables, would increase one's win rate. Vision-related variables refer to the following: vision score, wards killed and wards placed.

5.1.1 Overall

By looking at the results overall, the data seems to support this. The density of vision scores (see Figure 1) by winning players is slightly skewed to the right compared to losing players. This supports the initial hypothesis that players with a higher vision score win more games.

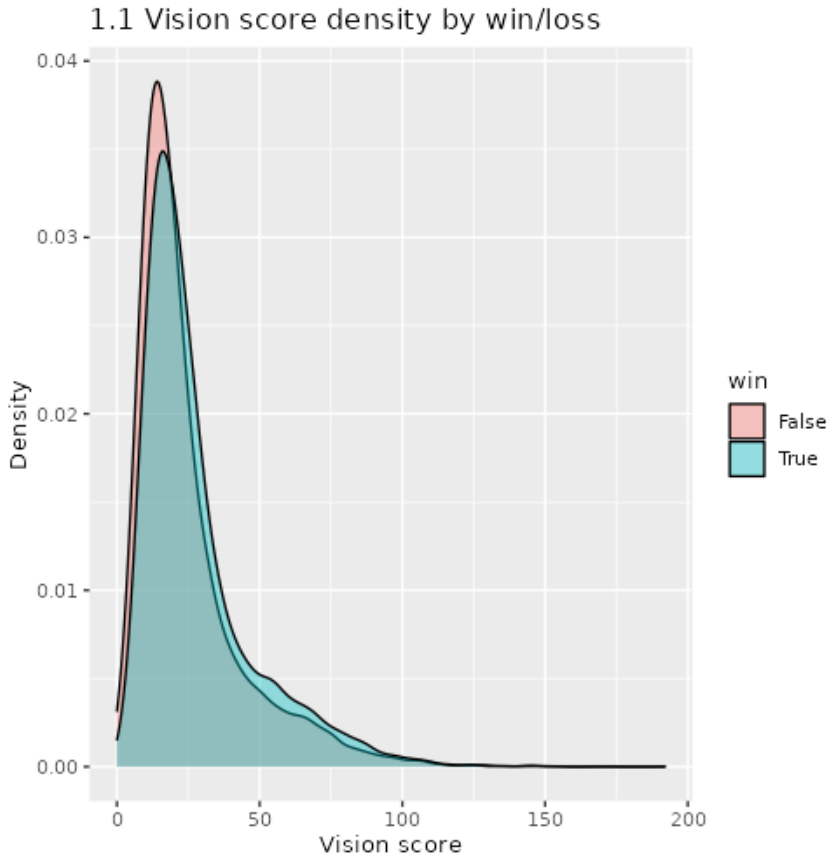

Figure 1: *Vision score density by win/loss*

When looking at the difference in vision score between two opposite teams (graph 1.2), it can observed that with an increasing difference in vision score, the rate of winning is almost proportionally dropping (for the team with the lower vision score). So as stated before, there is a slight increase in win rate for higher vision score in general, but more significantly when looking at the difference in vision score when comparing two teams in a single game. This leads to the conclusion that the relation of vision score for each game

has an overall higher significance than comparing general vision scores and win rates. In regards to the data, it clarifies that while the general vision score to density supports the initial expectations, the significant support of the initial hypothesis shows when looking at the difference in vision score compared to win rate.

5.1.2 Wards

The difference between the number of wards placed and killed is very small between losing and winning teams. Both are ever so slightly higher for winning teams as expected, but it is difficult to see anything conclusive.

5.1.3 Deaths

When choosing to look at the number of deaths in comparison to the vision score difference, it was assumed that the higher the vision score a player had the lower the deaths that player will have in comparison to the other players. However, graph 1.5 it shows that the higher the vision score the higher the deaths of the players. This contradicted the expectation the group had and therefore graphs 1.6 and 1.7 were created to showcase that the reason the deaths to vision score was this twisted was that the games where vision score became the highest were also the longest-running games, and in this game, the number of deaths was higher because of the game duration.

5.1.4 Violin

The two violin plots, 1.8 and 1.9, provide a more detailed view of how many wards the different roles place and kill. They tell mostly the same story as just looking at the means. However, they also reveal that jungle and mid laners kill more wards when they are winning. It is difficult to say whether players who focus on killing more wards are more likely to win in these roles. It is also likely that players who are already winning due to other factors are able to kill more wards due to their existing lead.

5.2 Champions

Two different kinds of histograms are used to investigate how the chosen champion plays a role in a victory. The first is an interactive graph based on the users' input used during a game. The second is static graphs used before a game and for analysis. Both graphs have been made interactive enabling users to zoom, pan and hover specific bars to see the exact win rate of the champion. This is to create a better user experience in case the x-axis comes out of view depending on the users' screen size.

5.2.1 Champion pick during a game

To figure out how the chosen champion plays a role in a victory, it was important to investigate lane pick and champion pack together. For that reason the graph 'Best Champion' (graph 2.1) has been made, which takes two input variables: lane and champion. It is a tool that should be used during champion selection to help players choose a champion when their opponent chooses a champion before them. E.g., if they see that their opponent has chosen Aatrox in the top lane, they will find that Urgot has the highest win rate against that champion. So they would be inclined to choose Urgot. It takes the two input variables and analyzes all the data and finds matches where the champion variable in the lane variable was played and whether the game was won. It then aggregates them and finds the champions that have the lowest win rate against them. With the result seen in Figure 2.

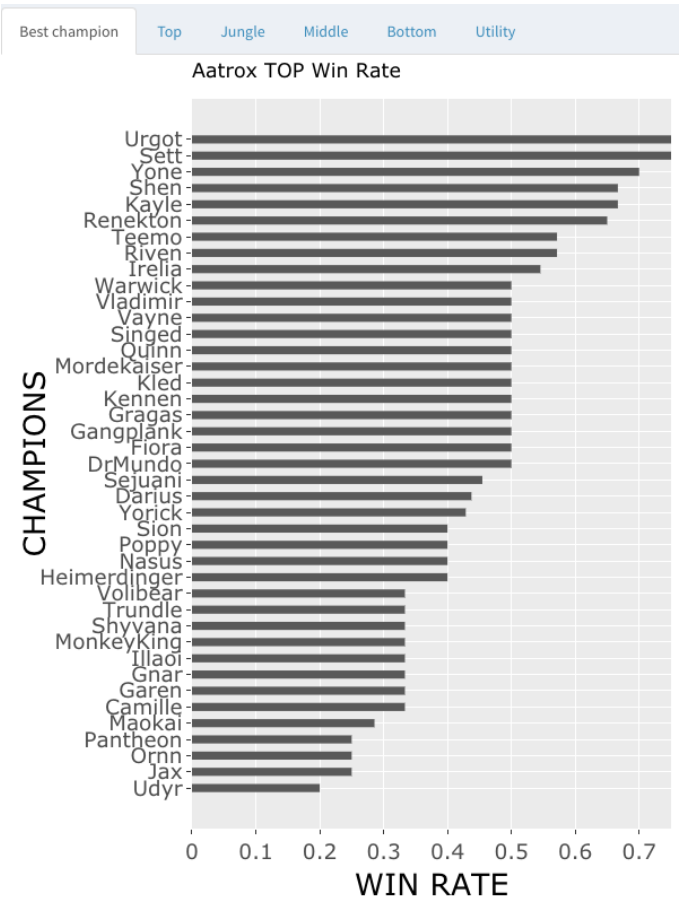

Figure 2: *Champion pick during a game.*

5.2.2 Champion pick before a game

League of Legends is a game with many patches where new champions are released and other champions are made stronger or weaker than they were before. This means that

there always will be some champions that are naturally better than others as the game can never be perfectly balanced. This is why it is crucial to investigate the currently strong champions before entering a game to help you with your champion pick. While it is great to use the above-mentioned tool when your opponent chooses before you, half of the time you will be the one that has to choose first. For this purpose, you can rely on the five graphs for each lane to see the currently top-performing champions. Each lane is represented in its own tab, "Top", "Jungle", "Middle", "Bottom", and "Utility", respectively.

For the best champion pick during a game, it was expected that the difference in win rates to be large with some champions being quite better than others against a certain champion in a lane. This proved to be the case, with some champions having more than 15% better win rates than other champions against a specific champion. For the best champion pick before a game, it was expected that the variance in win rates to be pretty small with a max of around 5% difference. However, it turned out to be a lot larger. The difference between the top-performing and bottom-performing champions is as much as 30%. This is most likely due to the limited number of records that the graphs are made from. The more data to pull from, the less variance and the expectations would most likely correspond better with reality.

5.3 Animation

The animation within the dashboard has been implemented using the `gganimate` package. Two `geom_segment` elements have been used to draw lines for victory and defeat over time (the boolean response variable: win). These two lines show how one of the nine available variables changes over time; one line for victories, and one for defeats. The animation plot tells a story about how variables change through time. It can be argued that a static plot could have achieved the same story. This is partly true, and partly wrong. As a static plot could have shown the same result as an animated graph, the animation tells the story from the shortest to the longest game, forcing the viewer to see what happens during the animation, which could lead to details hidden in a static plot.

5.3.1 Data frame

Each row in the dataset corresponds to statistics for a specific player. Each player has their own data, which is not relevant to the animation. As the response variable is equal for all five players within the same team, a sum of all values has been made for every team in every game. The resulting data frame is visualized in Figure 3.

	gameId	teamId	gameDuration	assists	deaths	goldEarned	kills	totalDamageDealt	visionScore	wardsKilled	wardsPlaced	win	totalMinionsKilled
	<dbl>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<chr>	<int>
1	6071320001	Blue	2278	61	40	75818	36	963644	213	32	80	False	772
2	6071320001	Red	2278	53	36	80185	40	1042391	211	25	80	True	849
3	6071339733	Blue	1362	56	39	48202	35	372553	75	9	34	False	415
4	6071339733	Red	1362	41	35	54079	39	478376	109	7	49	True	475
5	6071339748	Blue	1829	49	38	63577	37	691458	140	13	60	False	629
6	6071339748	Red	1829	65	37	66193	38	722152	144	27	55	True	650
7	6071350802	Blue	1437	30	15	41804	16	458085	105	8	48	False	522
8	6071350802	Red	1437	23	16	49690	15	581098	119	10	46	True	636
9	6071351675	Blue	1905	83	41	69162	47	673621	108	9	45	True	545
10	6071351675	Red	1905	48	47	65241	40	751518	105	11	49	False	594

… with 4,958 more rows

Figure 3: *The data frame for the animation data.*

5.3.2 Plot

When the variable and length of the intervals have been chosen, a button: "Generate animation" has to be activated. A method within the server observes this event and will create the plot with the given user input. This is needed in order to stop creating a plot every time the user chooses a new variable and/or changes the interval length. The plot will only be generated after a button click. As a consequence, an error message is displayed: *"Error: An error has occurred. Check your logs or contact the app author for clarification."*, as it cannot find the expected output before it has been generated.

Below is an example of the animation in two frames. The variable: vision score, is used as an example to see how it changes over time. Figure 4 and Figure 5 show the start and the end of the animation, respectively.

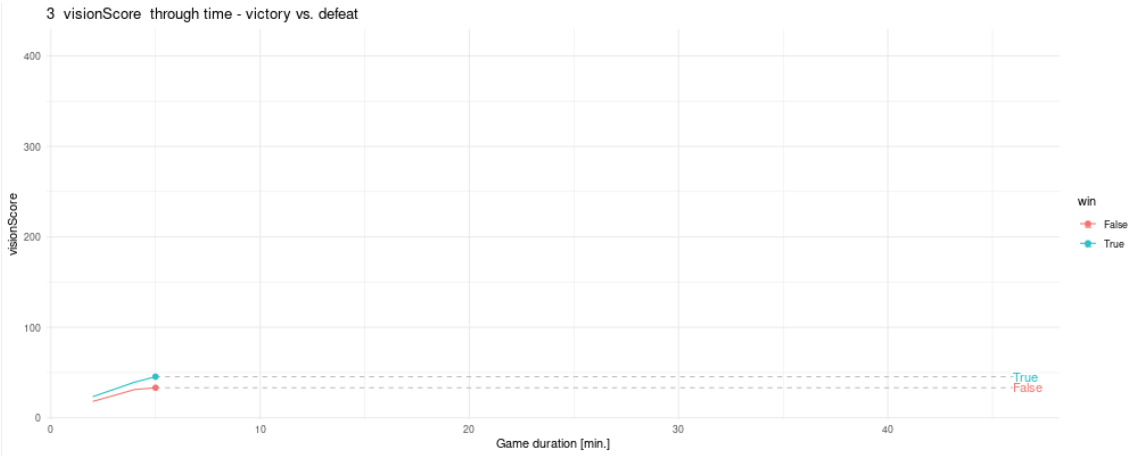

Figure 4: *Animation for the variable: vision score - the start of the animation.*

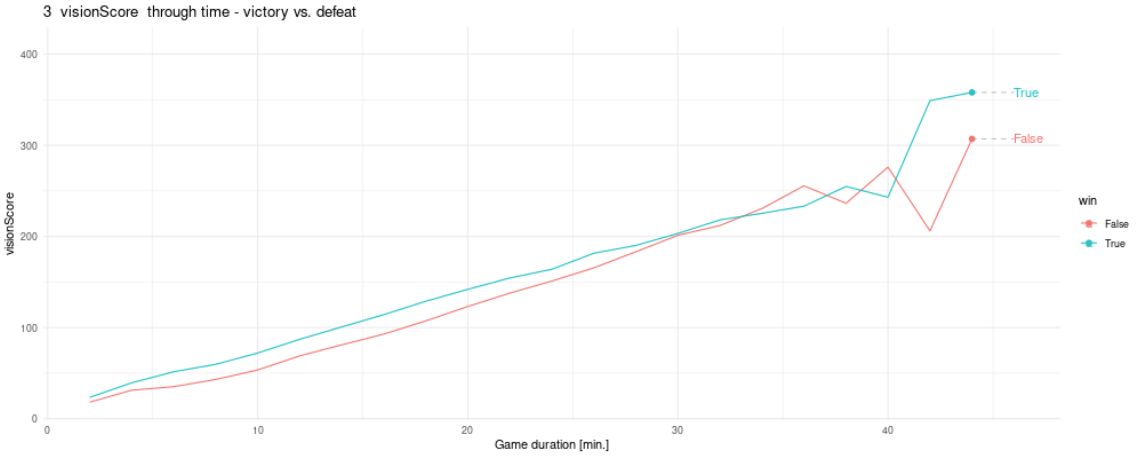

Figure 5: *Animation for the variable: vision score - the end of the animation.*

5.3.3 Variables

Each of the nine variables will be analyzed below based on how they change over time. A comparison of what was expected and what was the reality is given as well.

Assists, deaths and kills - It is expected that all three variables increase along with the game duration. Furthermore, the number of assists and kills should be larger for a victory than for a defeat, whereas it is the opposite for deaths. It is clear to see from the graph that all three variables increase with the game duration, and more assists and kills are acquired for victories than for defeats, and more deaths for defeats than for victories. The gap between the two lines increases with time but decreases in long-lasting games.

Gold earned - It is expected that the amount of gold earned increases with the game duration. This amount should be higher for a victory than for a defeat. A linear relationship is found between the variable and the game duration for most of the animation. The amount of gold earned is larger for victories, and the gap has the same trend as before.

Total damage dealt and total minions killed - It is expected that the total damage dealt and minions killed increase with the game duration. A victory should result in a larger amount of damage dealt and minions killed. Linear relationships are found for both variables until reaching long-lasting games. The victory line is above the defeat line for total damage dealt except for when a game is about 37 min. long. The two lines for total minions killed are almost the same from 20-35 min., where the victory line is above before and after that.

Wards killed and wards placed - It is expected that the winning team will have a larger number of wards killed and placed, as this will lead to great advantages throughout the game. The relationship is quite linear in the start but becomes more ”messy” in long-

lasting games. The gap between the two lines is found to be very close to each other, where the victory line is just above the defeat line until about 30-35+ min. when they cross each other multiple times.

Vision score - It is expected that the vision score is higher for a victory than for a defeat. The same observations for wards killed and wards placed are observed in the vision score, which makes sense as these two variables contribute to the vision score. However, the gap between the lines are larger with this variable compared to the other two ones.

5.3.4 Length of intervals

The analysis was made with the nine variables using a length of intervals equal to 120 seconds. The length can be chosen to be between 60 and 600 seconds with a step value of 30 seconds. The interval indexes are made from this equation:

$$intervalIndex = floor\left(\frac{gameDuration - min(data\_animation\$gameDuration)}{input\$animation\_length_of_intervals) + 1}\right)$$

- gameDuration - the game duration in seconds.
- data_animation - the data used within the animation.
- animation_length_of_intervals - the given length of the intervals acquired from user input.

The reason for creating these intervals is to remove sinusoidal lines, moving up and down, creating noise within the graph. The larger the intervals are, the less sinusoidal they are; however, the more inaccurate they will be as well. Figure 5 was created using intervals of 120 seconds. Figure 6 was created using an interval of 300 seconds, which results in fewer variations.

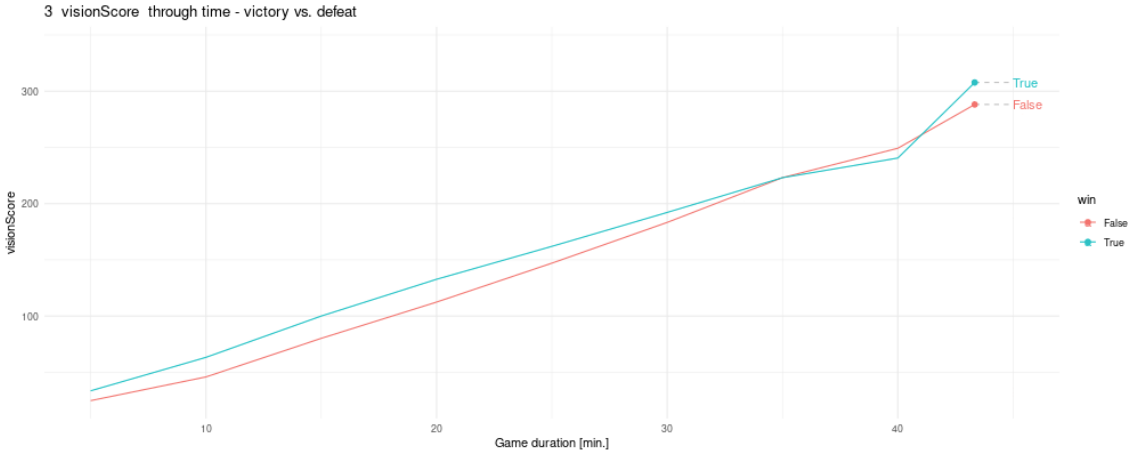

Figure 6: *Animation for the vision score at intervals of 300 seconds.*

5.3.5 Summary and conclusion

The question to be answered is (question 3 from section 2):

"How does each variable change over time, and what trends can be acquired from this?"

The expectations for each variable were correct for most of the cases. All variables had higher values for the winning team, except deaths, which is of course higher for the losing team. In some cases, the opposite was true for certain game lengths, but only for long-lasting games (35+ min.), and a few times in medium-long games (25-35 min.). Furthermore, the gap between the two lines started close to each other and then started to increase up until late game, where it decreased at first, and then the lines crossed each other multiple times. From the visualizations and analyses, it is clear that it is a great advantage to have higher stats in all variables (except deaths) in early game. This will increase the possibility of winning the game, as the gap between victories and defeats is greatest there. Nevertheless, when it comes to late game, this is not always the case, depending on the individual game. This is probably a result of champions reaching the same level and having enough gold to purchase all needed items to even out the possibility to win for both teams. Therefore, in order to maximize the possibility to win a game, a team should focus on hitting hard early game.

Further development could have been made to the animation. A specific animation length and frames per second have been chosen to make the graph easy to see what is going on, but also that it does not take too long to see it from the beginning. At the end of the animation, a pause has been included, which makes it possible to see the end result. However, it could have been improved by changing the graph from a GIF to a video. In this way, the user would be able to pause and skip through the animation to compare the two lines even better. This was not achieved, as it was difficult to do, and resulted in errors not being solved.

6 Discussion and Conclusion

6.1 Summary

LoL is an online competitive game with a massive player base. With its competitive nature and high stakes in both amateur and professional tournaments, a competitive edge is crucial. This edge can be gained by utilizing data visualization. Providing players with a quick overview of the most important factors contributing to a victory. The data was collected using Riot's API, and with it an investigation was launched into answering the following question:

Which variables are most important to examine in order to maximize the probability to win a game in LoL?

15 out of hundreds of variables were chosen and examined. These were things such as "champion name", "deaths", "wards killed", etc. From them, graphs of different types were created. One density plot, two line plots, two scatter plots, two violin plots, two box plots, six bar charts and one animation plot were created to properly investigate and visualize the data. It was done in the programming language R.
From these graphs, a few different results were found. Vision score had quite a significant effect. The bigger the difference in overall vision scores, the larger chance for a victory up towards 80% win rate. Another important variable that was found was the champion pick against a specific opponent. If you are able to pick a champion that directly counters the opponent you have a significant win rate boost. An interesting observation came when looking into the variable "deaths". Visualizing it gave some unpredictable results. This was because long-lasting games in general had more deaths, and so, the data visualization would be more focused on long-lasting games vs. short games. Therefore, it had nothing to do with deaths. Here it would have been necessary to divide the games up into intervals to make sure games of equal lengths were compared.

Furthermore, a different type of graph was made. An animation plot which plots nine of the 15 variables individually. When plotting the variable "minions killed", it was expected to be very important for the win rate, but it proved only to be the case in early game. As soon as the game reached late game, it evened out and did not matter as much. Other variables such as deaths, kills etc. were also of great significance, as one would expect.
All the graphs were put into a shinyapps.io dashboard, which can be found on the front page and in section 7.

6.2 Challenges

The limited number of records due to CPU constraints had an impact, which resulted in that the graphs do not represent reality completely. However, the graphs themselves work as intended and it is merely a question of CPU power available to render them with a higher volume of data making the visualizations more realistic.

6.3 Improving the course

Allowing students to choose their own datasets is a great way to structure the course. Providing students with a higher degree of freedom, thereby increasing the motivation. The flow of the course could be improved if it were to speed up a little. Some basic things were explained in too much detail, which did not provide much additional value.

7 References

[1] Group 12. *Data Visualization - League of Legends*. 2022. URL: https://troelszink.shinyapps.io/shiny_app/?_ga=2.102699905.1090781357.1670855845-563752812.1670855845.

[2] National Eye Institute. *Color Blindness*. 2019. URL: https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness.

[3] novint. *How much do League of Legends Streamers Make*. 2022. URL: https://www.novint.com/how-much-do-league-of-legends-streamers-make/.

[4] Active Player. *League of Legends Live Player Count and Statistics*. 2022. URL: https://www.activeplayer.io.

[5] Riot Developer Portal. *MATCH-V5*. 2022. URL: https://tinyurl.com/mrxj8xv5.

8 Glossary

assists Helping to kill a champion, but not achieving the final blow. 3

champion A character/avatar that you choose to play as in a game. 3

early game A specific point in the game, where not much time has elapsed yet, and only a few items have been purchased. 14, 15

GIF Graphics Interchange Format. 6, 14

late game A specific point in the game, where a certain amount of time has elapsed and a lot of items have been purchased. 14, 15

LoL League of Legends. 1, 2, 4, 5, 14, 15, I

minion A little unit that can be killed to collect gold and level up your champion. 3

team position There are five different positions/roles on each team - top, middle, jungle, bottom, utility/support. 3

vision score A metric to measure how much vision a player has provided for his team and denied for the opposing team. https://leagueoflegends.fandom.com/wiki/Vision_score. 3

ward A thing that will enable vision in places where champions are not present, and thereby cannot see anything. 3

9 Contributions

Student	Percent of Contribution (PoC)
Oliver	100
Peter	100
Troels	100
Kristian	100