**\Overview and Motivation**

Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal.

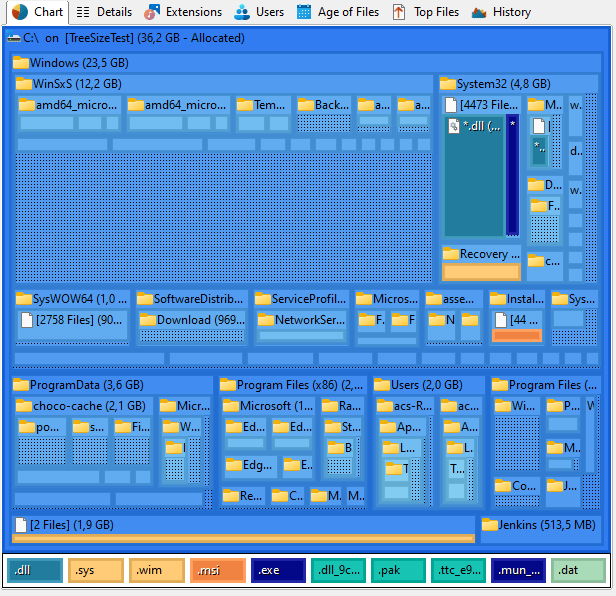
As avid gamers, we were looking to create and visualize the data corresponding to a player’s performance in a game they played. While we had a multitude of potential candidates for the game to visualize, we ended up deciding on League of Legends. This decision came from our own experiences with this title, as well as the ease of data presented by the multitude of API calls that Riot Games has access to. We have previous experience with League of Legends data and knowledge about what data is useful and what players want to see in data to help them understand their performance. This experience helped us to create useful metrics and effective visualizations.

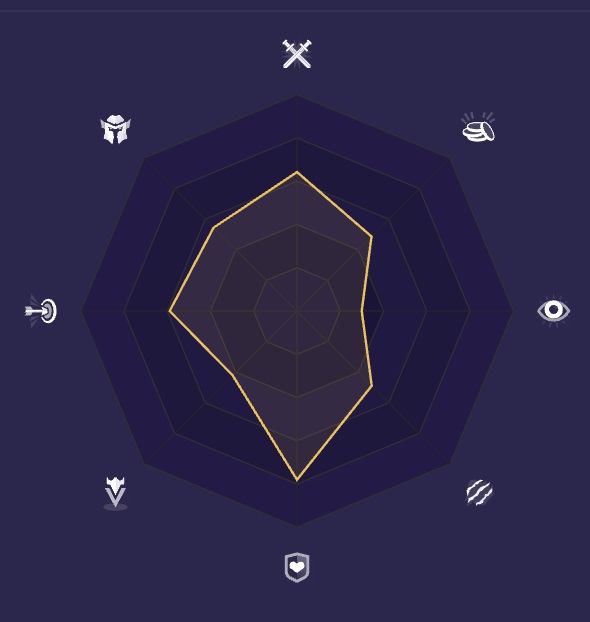
Our initial decision was to follow Apex Legends, but the difficulty in data acquisition turned us away. The data we can retrieve from the Riot API is significantly better than what we expected from third party apex sites. In the end our final project is significantly better and is strides better than what we were originally planning with Apex Legends data. The versatility and automatic up to date data with the Riot API is really cool to see. A player can play a game and we can immediately have that game in our visualizations.

**Related Work**

Anything that inspired you, such as a paper, a web site, visualizations we discussed in class, etc.

There are many sites that already exist to portray a large variety of data for a user to learn from their past League of Legends games. These websites, such as *OP.gg*, *Mobalytics*, and *League of Graphs*, were the largest inspiration and what we aim to recreate with our data. The pure amount of data given by the API calls allows us to explore multiple avenues in how different strategies and changes in games impact how a user performs. As for specific visualizations, many came from what we developed in homework assignments. The more universal charts like bar graph, line graph, and scatter plot all were developed from our assignment code. With some extra styling changes and implementation, they fit great with the rest of the project.

The calendar and radial chart were largely inspired by *Mobalytics* and how they display player data. We originally wanted to make our tree chart like the one displayed by TreeSize, but after struggling with the nested hierarchy we ended up going with a more traditional tree chart.



**Questions**

What questions are you trying to answer? How did these questions evolve over the course of the project? What new questions did you consider in the course of your analysis?

We are aiming to show a user a multitude of data based on a player’s performance and how it contributed to their wins or losses. We wanted to answer how a player’s champion pool contributes to their success. We also wanted to answer how a player’s performance compared to their opponents’ performance across several metrics such as gold/min, cs/min, kills, deaths, etc. Another question we wanted to answer was how much time a player plays contributes to their success. The most recently added question was comparing players against their role opponents. This question contributed to the creation of the radar chart and stacked bar chart.

Initially, we just planned on showing data over all matches to see how their play was changing over time. Eventually, though we decided that with the number of champions and roles, this wasn’t always a great metric. Thus we implemented the ability to select specific champions, as well as particular matches to filter the data.

This filtering allowed for us to finally compare specific matches to each other and see how a player is improving in certain attributes. Once we had that match comparison, we really wanted to know how we compared to the opponent in our matching role. This was best implemented in the dual bar, and radial charts. These focus on showing how we compare against our opposition to know if we are struggling, or surpassing our role opponent.

**Data**

Source, scraping method, cleanup, etc.

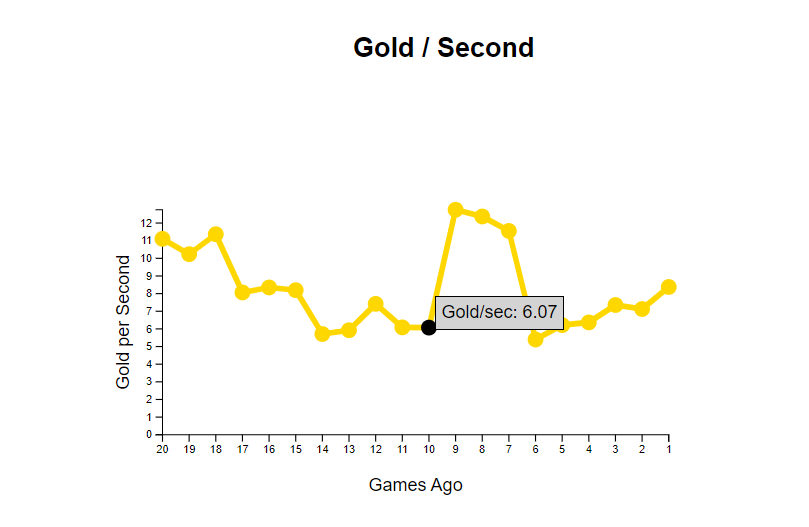
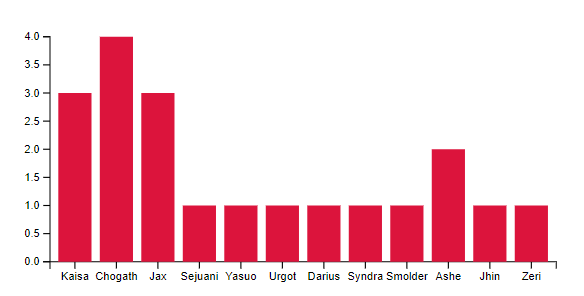
Our original data was going to come from an entirely separate game in Apex Legends. However, the API calls for both casual players and professional tournaments were lacking in their ability and magnitude of data. As such we pivoted to League of Legends as the API is significantly more supported and provides a better overview of how a game progressed. This new data required little cleanup as the APIs already provide a clean and efficient space. However, multiple calls were needed to capture the scope of the data we desired. Using an API required no scraping or cleanup other than sorting through the data we requested to extract the specific data that we needed for our charts since some data would require a larger call. An example would be if we wanted to find the gold per minute that a player got in a game. We would have to request a full match which contains data for every player. This data would have an entry that shows match time and gold acquired. We then have to divide the gold acquired by the match time and we get gold/min. Most data points did not require any math to calculate besides things like gold/min or cs/min.

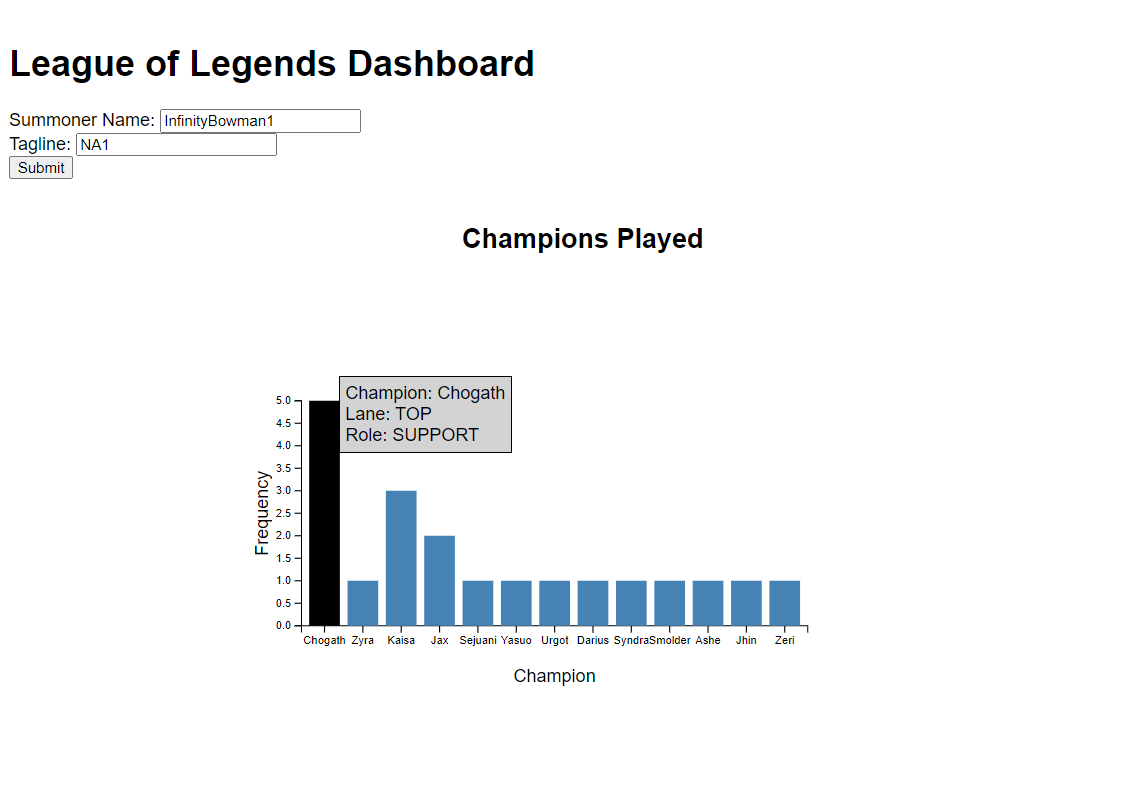
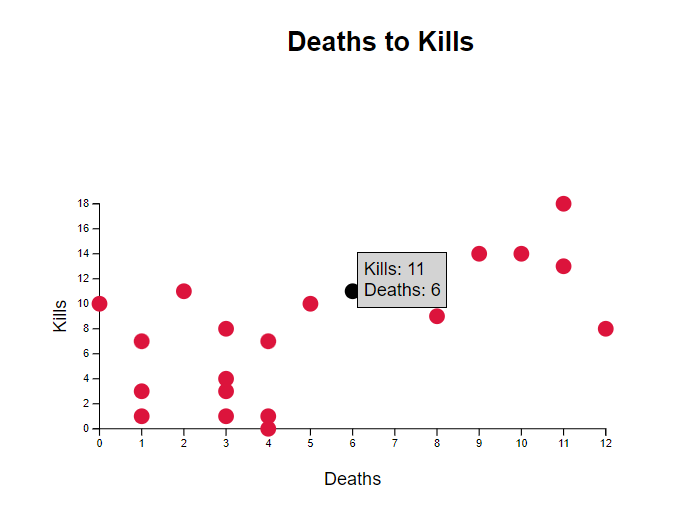
This new data required little cleanup as the APIs already provide a clean and efficient space. However, multiple calls were needed to capture the scope of the data we desired. Once we got the calls set up and working properly, it provided endless data that could be used. Not only did we have data on the match itself, but every player in the match. This allowed for better comparisons between a player and their opponent in the matching row. This is best seen in our dual bar, and radial charts as they compare the player to their foil.

**Exploratory Data Analysis**

What visualizations did you use to initially look at your data? What insights did you gain? How did these insights inform your design?

Initially, we explored many of the graphs used in assignment 3, these graphs are a wonderful baseline to assimilate the data we have and graphing in D3 that we are familiar with. We also just looked a lot at the JSON of the data to gain an idea of what aspects of the game were even workable. There is no time scalar so any visualizations that revolve around something in the middle of the game were out of the question. This led us to a multiple game approach where we compare different games together for an overall strategy. Another helpful thing for us was looking at the API documentation to see what types of data was available to us. This also steered us in the direction of grabbing match history and creating visualization of metrics in the match history.

We wanted to show the different types of damage and where they were applied with a nested tree. However, we faced two major issues in this implementation. First was the actual values, where with over a million total damage, a section of only 100 damage would be indistinguishable. The other came in our actual implementation where we were unable to efficiently create the nested layers of squares. Thus we settled for a more simplistic version showing the damage dealt to champions of the 3 main types. While the numbers are not always perfect, it was a significantly better view to understand how our damage was being dealt.  


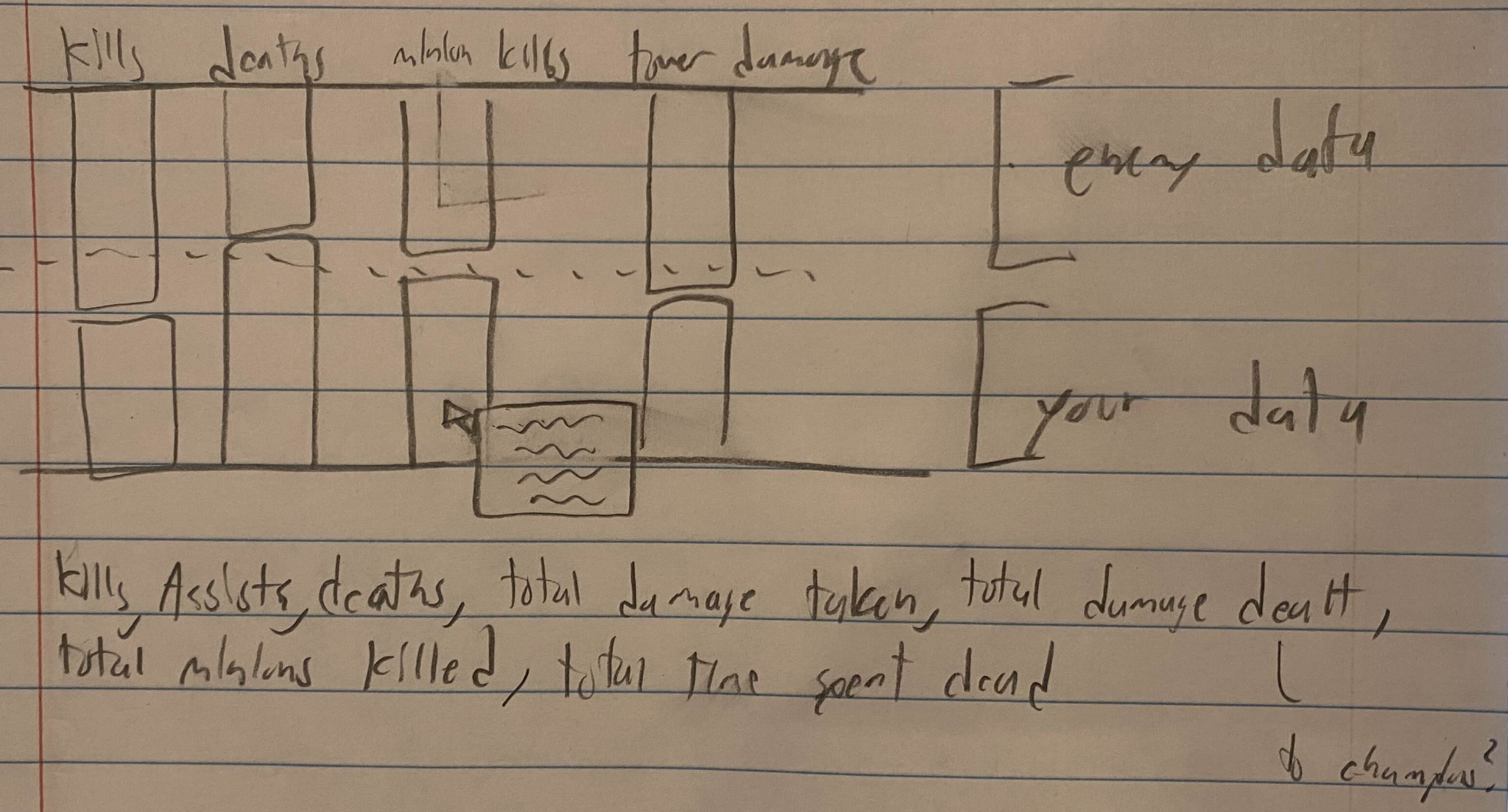
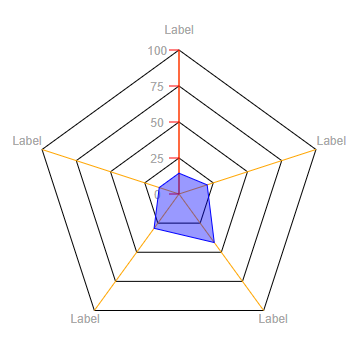


**Design Evolution**

What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course. Did you deviate from your proposal?

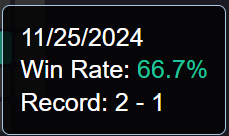
One visualization we knew that we wanted was to display the characters that had been played. This would give the user an idea of where their strength lye, and by giving some sort of win percentage, we would further gain an idea of how their performance was impacted. Once we got further into the development we realized how critical being to sort by champions would be. Before you could only look at a single match at a time for a character. Once we could sort by which champion played, our ability to see how we progressed or played was significantly improved.

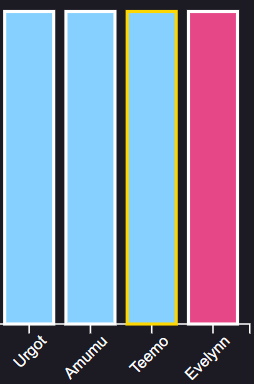
Our radial chart started with just showing the user’s data in the different key statistics. Once we created our dual bar chart though, we knew that the comparison to the opponent could easily be applied and would enhance the information we could gain from the chart. Adding the actual opponent was trivial and provided a very clear comparison in the sections the user excelled and lacked at. We also decided on a calendar heatmap since that would provide an easy way to visualize their performance history. The calendar heatmap is able to concisely show a lot of different data such as wins/losses, games played, and date those games were played on.



**Implementation**

Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.

The first element of interactivity we wanted was in being able to have a clear value associated with the data in the visualizations. This meant having some sort of interaction where the value can always be clearly identified. We felt it was most intuitive that when your mouse hovered over any element, it should display the information in a clear text form. While it may sound counter-intuitive to having a visualization, when making comparisons and needing strict values, having that option was crucial for our project.

Another element we knew we needed was the ability to filter and select outside of the match selection dropdown at the top. It was hard to actively select a match with the data we needed to display, so we thought that selecting data points would be best. This allows for a clear selection of a specific match the user is already interested in. So instead of guessing which match they had played in for that performance, a simple click would display the info. The same is true for filtering by a champion. Not wanting to add a further dropdown for all champions, our bar chart of champions played became the perfect selection tool. It allowed us to easily select a character we have data on, and adjust the page to reflect the user’s performance on their selected champ.



**Evaluation**

What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?

The biggest lesson came in picking the right visualization of the data. While we often stuck with the visualizations we selected, we had to do lots of brainstorming on the possible directions we could go and what we wanted the final design to look like. We also had to take into consideration what data would be the most valuable. With all the data we had available, this task became much harder as we could have made countless different graphs and had to stick to a few to keep the project moving in a timely manner.

We believe that the visualizations we have provided allows the user to get a more than clear understanding in what attributes they need to improve on to win more games. They can compare champions to see which ones fit their strengths, compare to their opponents with the same roles, and view trends in their results to see how streaks might occur.

The overall effectiveness of our data visualizations is quite good and fairly quick to understand. However, the one chart we question is the dual bar chart and its ease of comprehension. While a simpler bar chart might convey the true numbers better, we wanted to focus on percentages and how much or little more the difference is. While the legend is decent for this chart, a future change could further expand on explaining the use of the percentage for viewing and which metrics are good when they are higher and which are good when they are lower. Lower deaths is better and high gold is better.