NFL Offensive Statistic Visualizer

Final Write Up

Yuval Dekel

Justin Luk

Chad Collins

Nishant Boddupalli

Goals of the Project:

This visualization aimed to provide insight into what makes a National Football League team effective and how much these factors impact the performance of the team. Since this in itself, encompassed several different goals, the project had been broken down into achieving several simpler aims. First, the visualization was expected to provide the user with an understanding of which Football metrics have the greatest impact on a team's performance within a given season or over a range of seasons. These metrics include but are not limited to touchdowns, pass yards, rush yards, field goals and fumbles. Second, the visualization aimed to give the user the opportunity to observe and analyze, in depth, the performance of any team in the National Football League. This was to be achieved by providing holistic views of that team's performance over a range of seasons and over a range of games within a single season in terms of different metrics. Finally, the visualization aimed to give the user an opportunity to discover how the teams compare to each other in terms of their performance in certain aspects or metrics of the game.

Over the course of the semester some of the aims of this project have undergone changes. *Initially, one of the goals of this visualization had been to provide the user an opportunity to compare the performance of one team against the performance of one other team over a given set of metrics.* However, this idea was changed so that the performance of a given team in terms of a particular metric was compared to that of all the other teams in terms of that particular metric. Our main purpose is to provide insight into what makes a team successful in the NFL and how teams evolve over time in certain metrics.

Examples of questions that can be answered using this visualization:

- What specific metric could make one team better than another and results in more wins?
- How has my favorite team changed over time, have they gotten worse or better? If so, which areas have they gotten weaker or stronger in?
- How much do pass yards matter versus rush yards?

- How much do discrete events affect the success of a team, such as the loss of Peyton Manning for the Colts in 2011?
- Given a bad year for a team, did they just have a bad year or was it a steady decline into failure?
- What common statistics does a #1 ranked team typically exhibit? What about the worst team?
- How do the #1 and #2 ranked teams compare across all offensive statistics, were they neck to neck or far apart across many variables?

Description of the Dataset:

In order to answer the above questions, the visualization leverages a rich dataset in the form of comma separated value files. Using the collection of National Football League game data which was collected by Sunshine Forecast Enterprises, the visualization will expose the hidden value in this widely tracked sport. Firstly, the data from this source is organized by years with a separate comma separated value file representing the National Football League data for each season. Within this file for each season/year, every game that occurred within that year is listed. For each game played, several details are provided such as: for each of the two teams involved in a match: name of the team, total score in the game, number of first down, number of third downs, rushing yards, number of attempted passes, number of completed passes, total pass yards. In order to make interacting with this data simpler and easier, a script was run to parse all the data and place it into a mongoDB database. Each mongoDB record corresponds to one football game. Each record corresponds to one row in the original comma separated value file. This formal allocation allows for the utilization of mongoDB's powerful indexing to filter specific matches based on criteria such as "matches in which a specific team played". Overall we parsed 7000 games with 14 variables per game for 32 teams. The data spanned a decade to fully capture insights into how teams change over time. The final size of our data manipulated in mongoDB ended up around 8 MB.

Example screenshots

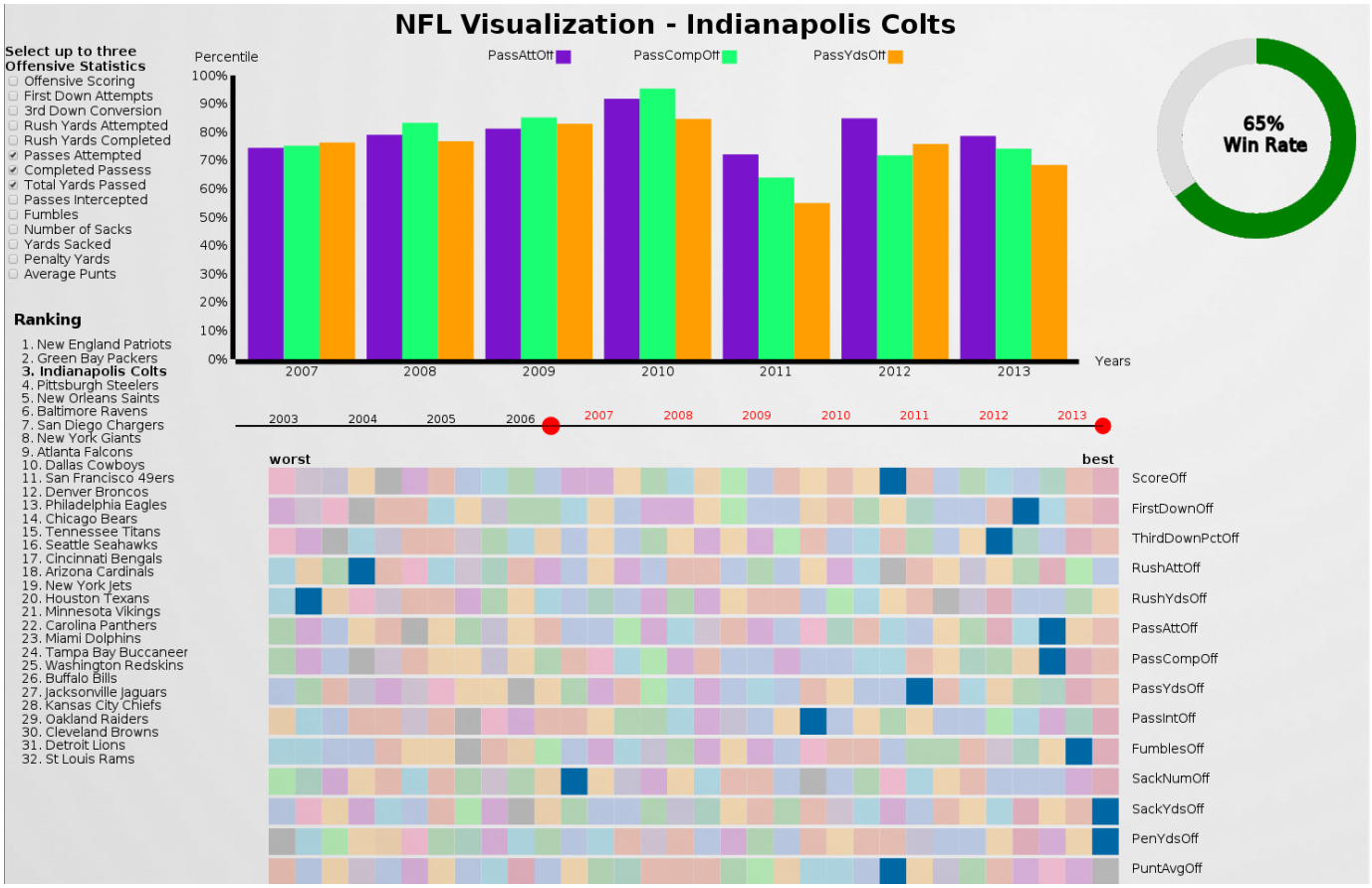


Figure 1: The Colts are shown from 2007-2013, in particular 2011 suffers a dip in metrics due to Peyton Manning's leave that year. It is evident with a quick glance that the Colts suffered in performance in 2011 to what many can attribute to the loss of a star quarterback. This offers some hard evidence that one strong player can greatly affect a team's effectiveness if he is a recurring decision maker. With our visualization, users can quickly develop insights from viewing team trends over time and correlating it to pivotal discrete events in the sports world.

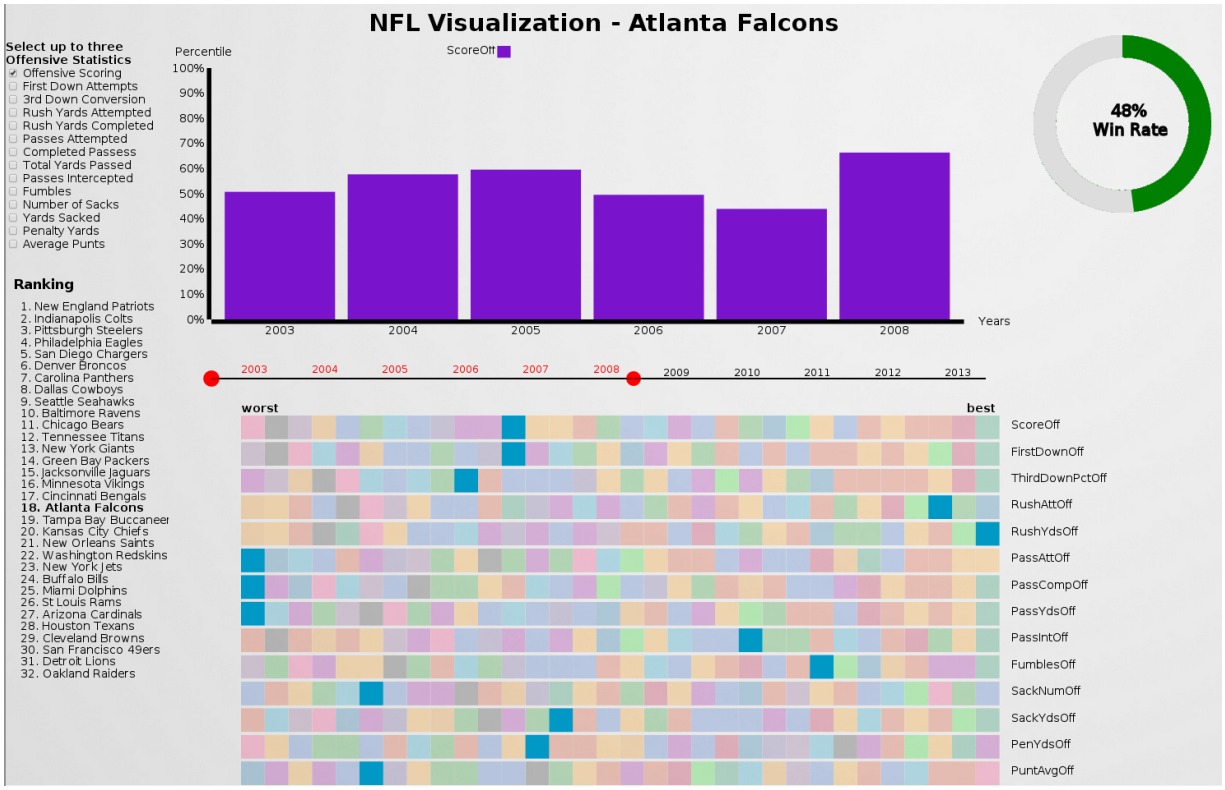


Figure 2: This is the basic layout users initially see, in the first visualization users can see the trend of Falcon's scoring per game over time for the selected years (2003 to 2008) in this case. The radial win rate quickly shows an approximation of their aggregated record over 6 years. The bottom visualization ranks the Falcons by metric across all 32 teams in the league and on the left you can see where the Falcons rank by win rate over the 6 years. Immediately you can see the Falcons show off a strong rush game, however in the NFL that pales in comparison to a talented passer and lightning fast wide receivers. The Falcons come up short in all forms of pass metrics, which can be correlated to their low scoring rates.

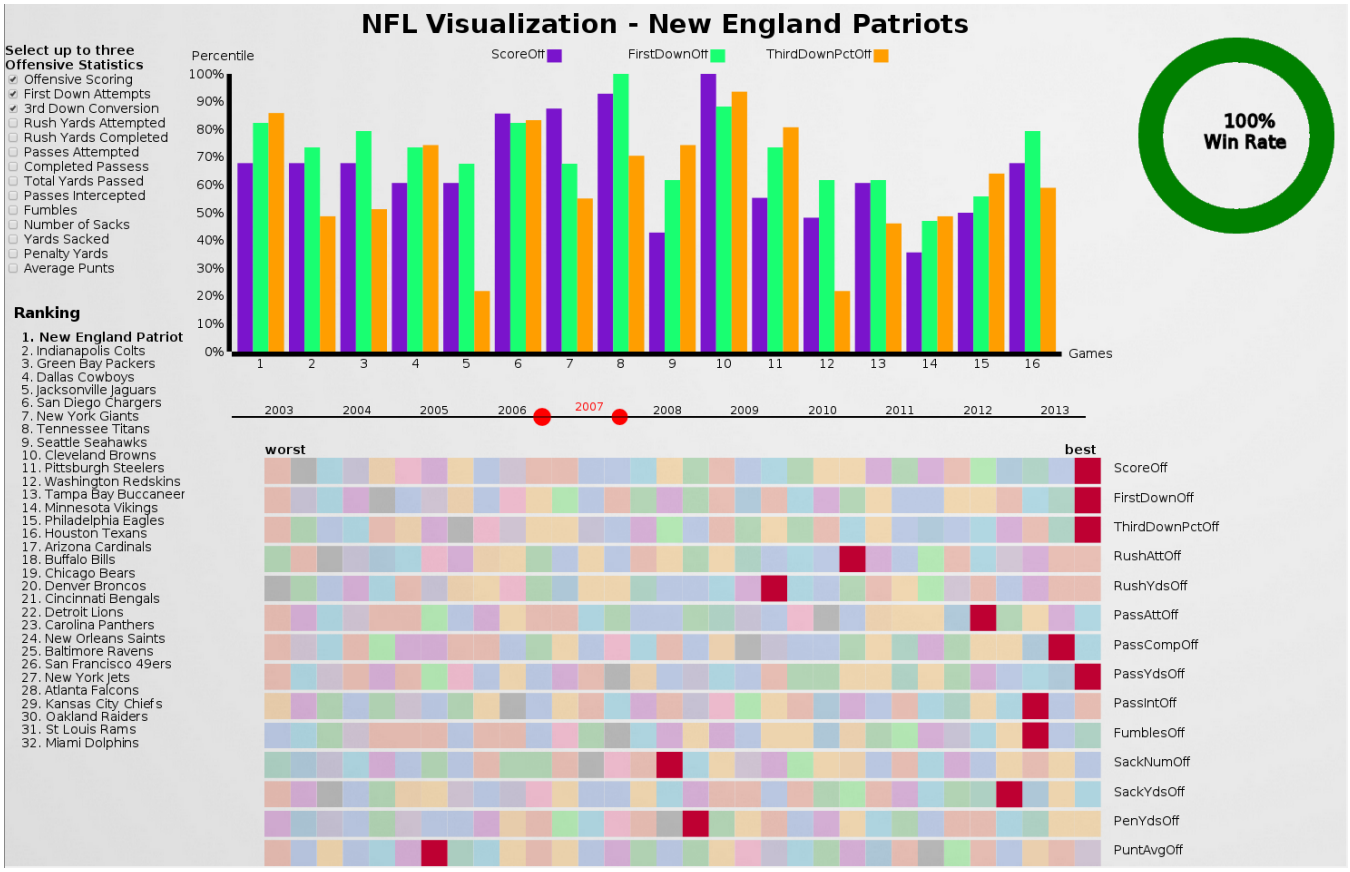


Figure 3: This visualization indicates have the New England Patriots fared compared to the other teams across different metrics during the 2007 season when they had a 100% win rate. The bar chart provides a game-wise breakdown of the New England Patriots performance in terms of the selected metrics. It is clear their ability to gain first downs from third down conversions attributed to a high offensive score. In addition, their high completion of passes and leading pass yards contributed to a highly effective attack. This skill in conjunction with an ability to avoid sacks, fumbles, and interceptions proved effective in their attainment of an undefeated season.

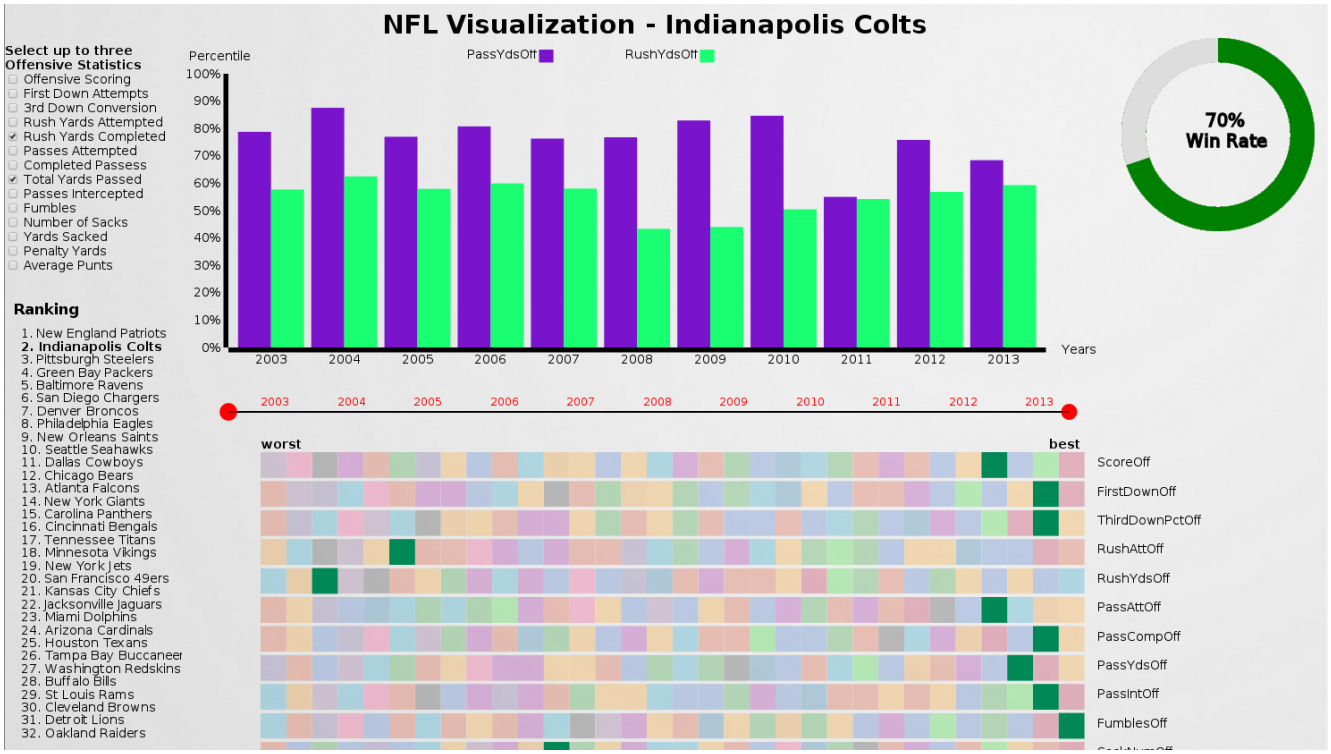


Figure 4: This image shows the performance of the Indianapolis Colts between the years 2003 and 2013. From the bar chart it is very clear that the team had a pass yards average that is significantly closer to the highest pass yards by any team for that season. On the other hand, the team also had a very mediocre performance in terms of average rush yards, compared to the best rush yards performances. Also, from the bottom visualization, it is clear that the Colts rank 3rd from the bottom in terms of rushing yards over this period and 4th from the top in passing yards. However, during this time, the Colts had a 70% win percentage. This shows that pass yards are significantly more important that rushing yards in affecting the performance of a team.

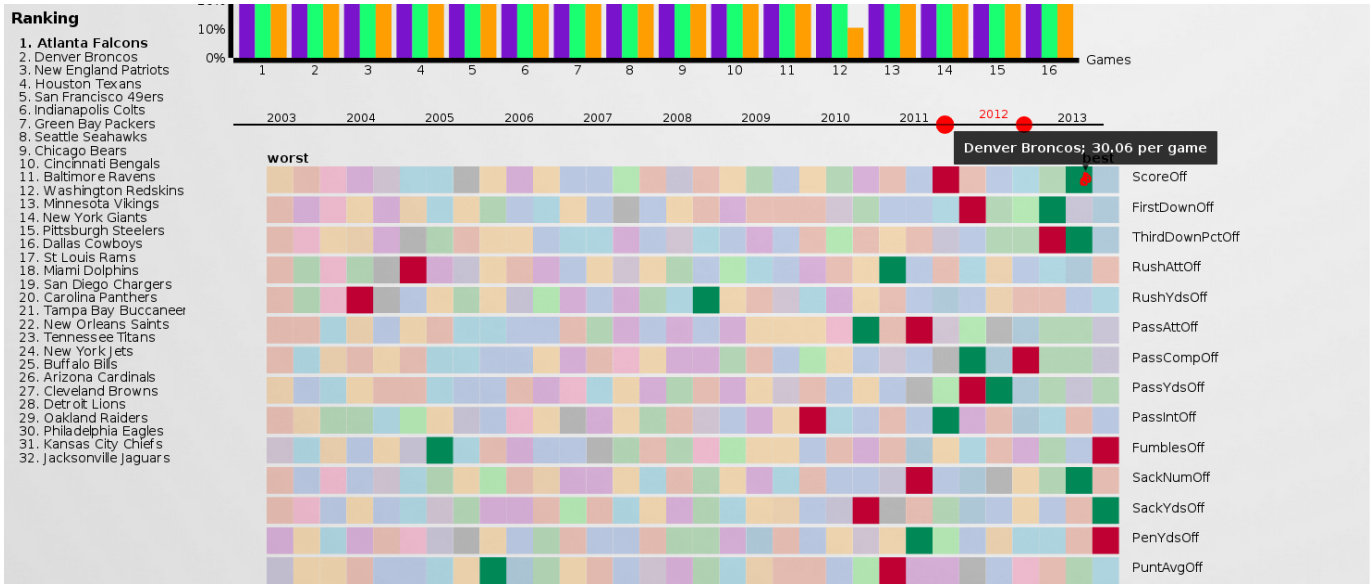


Figure 5: This image shows how the Denver Broncos, indicated by red, fared against the Atlanta Falcons, indicated by dark green, over various metrics during the 2012 season when the Denver Broncos had the highest win rate and the Atlanta Falcons were second. This provides insight into the fact that teams that had a head to head competition for the top rank tended to compete very closely in most of the metrics. The major outliers are those that represent rush yards - metrics that as evidenced before are not very indicative of a teams performance in the NFL.

IDEs used and advantages

For this project, the most common IDE/texteditor used was Notepad++. The user interface for Notepad++ is very convenient for writing code because it provides considerable assistance to the user. Variables declared are often reused multiple times within code and in languages like Javascript, where this occurs quite often, Notepad++ has a feature of providing autocomplete options for the words being typed. Over the course of writing a large file, this could save a considerable amount of time and effort. Another important feature of writing good code is proper indentation for readability and maintenance. The automatic indentation features in Notepad++ and the features for keeping track of open/closed parenthesis and open/closed curly braces is particularly useful especially when writing code in Javascript which often includes many levels of nested functions. The indentation features also make debugging the code much easier. The lack of an auto code correct feature forces the user to ensure that no mistakes were made thus giving the user a better opportunity to develop proficiency in the coding language. To test out the written Javascript code, a web browser was necessary. Chrome was the most widely used web browser in testing the code for this project. The features provided by chrome such as inspecting elements and the console allow the user to get an in depth view of the html document generated or even test minor changes in the browser itself. This helps immensely because in the case of some part of the code not working as intended, the user could look into what lines of code cause errors. The console feature was the most useful in debugging the code. This is because any syntax errors always show up in the console. In addition, some text or data can always be logged to the console. This could help in determining if a certain portion of the code is running and which breaks the program.

While working in a group on a coding project, it becomes vitally important to modularize the code and have different individuals working on different areas of the script. For this reason, it becomes very important to have a competent application to allow code from different users to be merged into a single script. The version control offered by Git was very useful in this regard.

This allowed different individuals to work on different parts of the code at various times and then bring it all together without one script affecting the changes in the other.

Information Visualization Toolkit Review:

The toolkit that was implement in this visualization was Data Driven Documents also known as D3. There were many reasons for why D3 was chosen over other options. This included the fact that D3 works on the web. It is very easy to view visualization made using D3 and all it needs is access to the data and a web-browser. Other options such as Flare which is a predecessor of D3 need at least one other plugin in order to run seamlessly on a web browser.

D3 was also found to be a very flexible toolkit. This was because it works so smoothly with other existing web technologies.

One of the biggest advantages that D3 provides is that it does not limit the scope of the developer's creativity by providing pre-existing charts or features that need to be implemented and that are often very difficult and complicated to customize according to specific needs of the visualization. This is primarily because D3 is neither a graphics library nor a data processing library. One of the key aspect of it is that it provides effective tools that can be used to link data and graphics. Because it neither focuses too heavily on complicated graphic-features, it offers more flexibility and because it does not focus too heavily on drawing data from sources, its functionality is an ideal balance between these two very important aspects of visualizations making it a very effective toolkit.

Team Member Contributions:

Due to the nature of the project, it was found necessary to break down the work involved in creating the deliverables. For this reason, the 4 individual members in the group took responsibility for different aspects of the project. Justin Luk was the team leader. Chad Collins was the Keeper of the "to do" list. Nishant Boddupalli was in charge of convening the meetings and integrating the final report and Yuval Dekel was responsible for integrating the work for the

mid-term progress evaluations. As far as developing the project itself was concerned, each of the team members played an equal part. Yuval Dekel took charge of developing the backend of the project including populating the mongoDB database with data from the comma separated value files. Yuval Dekel also took responsibility for developing the second view of the visualization. Justin Luk was responsible for developing code to scale y axis according to the data selected and developing code for the radial chart at the top right corner of the visualization. Chad Collins was in charge of developing the code to display the radio buttons to filter the displayed data. Nishant Boddupalli was in charge of developing the code for a range slider that could be used for dynamically filtering the data that gets displayed in the visualization.

On the whole, the members of the team got a very good opportunity to learn how to divide up responsibilities within a team so that responsibility of the majority of the work for the project is not placed upon one or two individuals while at the same time no individual is left with little or nothing to do. Also the importance of effective communication and assisting each other during the process of developing the project stood out to all members of the team. The most important learning, specifically in Computer Science related projects, was the idea that coordination and working in parallel streams is vitally important.

Code Module Percentages

The backend was setup by our group 100% in creating and setting up mongoDB alongside Node.js to store all of our data and parse it. Ultimately it was a dynamic and powerful method to quickly aggregate and manipulate decently large sets of data.

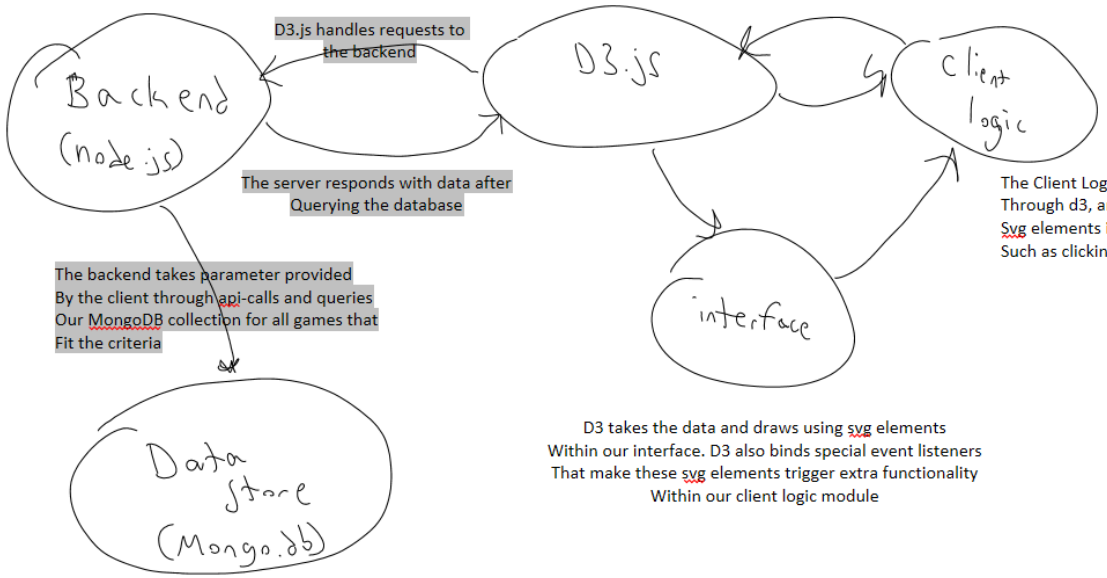
Our interface spanned 4 main module pieces that combined into our final visualization. The first visualization consisted of the upper bar chart that allows dynamic querying and details on demand. The basis of the chart was leveraged from the creator of D3, however we augmented and added multiple custom features. Ultimately the bar chart was roughly 20% imported and 80% custom written for our needs and purposes.

The second visualization was created 100% from scratch by our group in order to fit our exact needs to best visualize our data set. Over many iterations, we created what we believe to be

an effective and easy to use information visualization that leverages details on demand, filtering, querying, and highlighting/linking with the first visualization.

The additional modules include a radial visualization to quickly display a team's win rate in a digestible manner. This was 75% taken from an example the creator of D3 made, but was custom fitted for our needs. Our group added labels and used the module to feed in relevant data to draw an accurate display for the final 25%. We also created a 100% custom timeline that acted as a filtering anchor for both of our visualizations. This augments the ranking of teams over time and also offers a unique single season view to drill down further.

The remainder of the architecture diagram is to provide a quick view of how all the pieces of technology we used synced up to provide our final visualizations.



The Backend is developed in Node.js and contains mostly Logics that looks through a Collection of games and Aggregates metrics such as Average per-game values for Each metric

The Client Logic mostly takes the data from our server Through d3, and reformats it into arrays that can be bound to Svg elements in our interface. It also sets up listeners for events Such as clicking on elements and modifying criteria.

D3 takes the data and draws using svg elements Within our interface. D3 also binds special event listeners That make these svg elements trigger extra functionality Within our client logic module

The Mongo Database contains a Single record for each game that was Played between 2003 and 2013, no additional Data such as average-statistics for a team Is stored.