

NHL Game Data (2000-2001 through 2019-2020)

Maya Barron

[R script](#)

Background:

This app explores NHL Game Data from the 2000-2001 season to the 2019-2020 season from 31 active NHL teams and 2 defunct NHL teams (Atlanta Thrashers and Phoenix Coyotes). The only active NHL team that is not included in this visualization is the Seattle Kraken, as they were introduced to the league in the 2021-2022 season.

Unexpected Findings:

There is a very significant outlier based on shots on goal. The Tampa Bay Lightning had 88 shots on goal in a single game, and they only managed to score three goals in that contest. Based on the data used for this project, the mean shots on goal between all contests is 29.8, so by that standard, 88 shots on goal is ridiculously high. The next most shots on goal in a single game is 64, which was accomplished by the Anaheim Ducks who scored four goals that game. Additionally, it's interesting that two teams have scored more than ten goals in a single game during the time period when this data was collected (the Washington Capitals scored 12 goals and the Ottawa Senators scored 11 goals).

I was also surprised to see that there wasn't as much correlation between shots on goal and goals as I would have thought. One would probably expect more goals to be scored in games where the puck is shot at the net more, but there doesn't seem to be much of a correlation between those two variables. In fact, I found one game played by the Colorado Avalanche where the team had 6 shots on goal and 6 goals, which means that every attempted goal was scored.

Interface Creation:

I manipulated the data quite a lot before creating the interface. Since this data runs up to Covid times, I wanted to get rid of any games that weren't played so they didn't cause confusion on the plots. I also had to combine two different datasets so the team names would be visible and usable, rather than just relying off of the numeric team_id that wouldn't be understood by the general audience. I also had to rework the New York team names as they appeared in the format of "NY teamname teamname" in the original dataset.

I started my interface creation by first sketching out what I wanted the final visualization to look like with the graphs and dropdown menus. I then began the creation by first focusing on making the dropdown menu work in cooperation with the graphs. I then tried to add on the brush that I was planning to use, but it wasn't going well and I kept running into "Error: object Object". I didn't know exactly how to solve this issue, so I broke down my code and tested everything I could inside of R before fixing it and trying it again in shiny. I also tried just making the brush work without the dropdowns which helped me solve most of my previous problems. Eventually I got both the dropdowns and the brush to work together and create a working visualization.

I chose "team" and "game outcome" as my dropdowns because when I search for hockey related things on the internet, I usually know specifically which teams and what types of games (winning, losing, or both) I want to look at before I look for information, so I figured the general public would want the option to select these variables before looking at graphs. I wanted two different plots to break up the information and make the whole shiny app more visually appealing. In the NHL, the home team typically wears their colored jerseys while the away team wears white, so I made my bar graph reflect that.

Reactive Structure:

I have one `selectInput()` for the teams, and one `selectInput()` for the game outcome. When the shinyApp is initially created, the team `selectInput()` is populated with "New York Rangers" and "New York Islanders" while the game outcome `selectInput()` is populated with "win". I have one `plotOutput()` for the scatter plot, which also includes a brush, and one `plotOutput()` for the bar plot. Both of the plots respond to inputs from the dropdown menus, and the scatter plot is colored according to the teams selected. When a brush is applied, the selected points on the scatter plot become a little bigger, and the bar plot only takes into account points that were selected.