**Problem Statement**

It is interesting to see how geographic locations sometime play into people’s interests. These can be especially evident in the U.S. There are cultural differences such as West Coast vs East Coast, Northeast vs Midwest, and South vs North. We look at sport referencing tweets from within the U.S., and plot them on the map based on which sport they talk about. As far as which sports to consider, we look at tweets referencing baseball, basketball, football, and hockey. Although we will not get into the same level of detail, <https://blog.twitter.com/2014/nfl-fan-map-where-are-your-team-s-followers> contains an interactive map of a similar idea. This was discovered while researching, and shows some of the capabilities within this domain.

**Learning Objectives**

The intent of this project is to gain insight regarding where people were talking about certain sports. We want to explore the geographic information contained within tweets, and be able to see visually where in the US people were more likely to be talking about one sport versus another. We will be able to see areas where one sport is dominant, and other areas where chatter is split amongst many sports. In doing so, we will be able to learn methods of plotting in R, as well as tweet filtering techniques.

**Description of Algorithms Used**

TBD

**Description of Implementation Details**

At a high level, the program captures all tweets from within the US, filters them based on what professional sports league they are representative of, and then maps them based on the league. The map shows all tweets received that contain items from each league, with each league separated by color.

Looking at more detail, the data is gathered by opening a tweet stream using streamR and filtering based on location. Tweets are only collected with U.S geographic coordinates. Once the tweets are collected and stored as a data frame, they are passed through filters which check if they contain any relevant sports vocabulary. The tweets go through 4 filters, each representing a different sport. The filters contain the name of the professional league (ie MLB, NBA), as well as every team in the league. If multiple sports are detected while going through the filter, the tweet is discarded. Once the tweets have gone through the filter, they are added to a running data frame of the tweets which have been flagged by the filter as being sport related, and indicate which sport they represent. A separate data frame is then produced to contain only the geographic coordinates, and the league of the tweet. These points are then plotted on the map using ggplot. This can be run for x iterations, determined as a constant which can be modified. The duration for each tweet stream can be modified in a similar fashion.

**Description of Experimental Design**

Many iterations of build up led to the final product. It began by capturing and plotting all tweets within the U.S. Next was filtering tweets based on relevance to our topic. It began with only one league, and a few teams, which did not produce very many tweets to examine. After all the sports were added to the filters, making the map most effective was next. The main adjustment was modifying the color scheme to display color blind friendly colors, along with some other minor tweaks. The final hurdle was formatting the R code so the functionality would run in a loop, which is discussed further in Lessons Learned.

**Analysis of Results**

The results consistently show that football is the most talked about sport within the U.S. The total count of tweets it detects far outweighs any other sport, and it spreads across the U.S. It is easy to pick out cities on the map based on where the concentration of tweets are.

**Discussion of Lessons Learned**

R is a very powerful language, but not without its quirks. Local routines do not access global variables like a lower level language like C does, so in designing routines this must be kept in mind. This can also a similar case for loops.

**Resources**

http://pablobarbera.com/blog/archives/1.html