Final Project Report: Pitch Arsenal

Author: Jae Hyeon Kim, jkim554

Section 1: Data

`pybaseball` is the package used in this project, which is a Python package for baseball data analysis. There are various functions that pulls different datasets; however, the datasets of focus are the datasets obtained from the function `statcast`, `statcast_pitcher`, `pitching_stats`, and `playerid_reverse_lookup`.

The data pulled from `statcast` includes pitch-level features such as the Hand Pitcher Throws With(`p_throws`), Side of the Plate batter is standing(`stand`), horizontal and vertical release position of the ball (release_pos_x & release_pos_z), and many more totaling up to 92 columns. `statcast` has two arguments `start_dt` and `end_dt` that returns all the statcast data between those two dates. The variables of interest in this dataset are:

- `game_date` : Date of the game
- `p_throws` : Hand pitcher throws with
- `player_name` : Player's name tied to the even of the search
- `pitch_type` : The type of pitch

The data pulled from `statcast_pitcher` is a more pitcher-specific query. It has three arguments: `start_dt`, `end_dt`, as well as `player_id`. Inputting the `player_id`, which can be obtained using the `playerid_lookup` function, returns a specific pitcher's statistics of the set date range. The variables of interest in this dataset are:

- `pitch_type` : The type of pitch
- `release_speed` : Pitch velocities from 2008-16 are via Pitch F/X, and adjusted to roughly out-of-hand release point. All velocities from 2017 and beyond are Statcast, which are reported out-of-hand
- `pitch_name` : The name of the pitch derived from the Statcast Data
- `pfx_x` : Horizontal movement in feet from the catcher's perspective
- `pfx_z` : Vertical movement in feet from the catcher's perpsective

The data pulled from `pitching_stats` includes league-wide season-level pitching data. The three arguments used in this function are: `start_season`, `end_season`, and `qual`. For example, setting `qual = 10` is to only select pitchers who played at least 10 innings. The variable of interest is:

- `IDfg` : Fangraph ID

Using the fangraph id from the function above, the statcast ID can be found using the data frame returned by `playerid_reverse_lookup`. The arguments for this function are `player_ids` (list), and `key_type`, which the valid inputs are `mlbam`, `retro`, `bbref`, and `fangraphs`. The variable of interest is:

- `key_mlbam` : Statcast ID

Section 2: Question of Interest

Statistical data is crucial in baseball because they are used in numerous ways. Even commentators rely on statistical data sheets of players to make game commentaries when a game is televised nationally. Major League Baseball is the second most watched sports in the USA. As a result, fantasy baseball has become increasingly popular among a wide age spectrum. Since fantasy baseball uses real life statistics of baseball players, even the fantasy baseball users rely heavily on the statical data of the baseball players. Furthermore, the baseball strategic teams can strategize better with statistical data. With this web application tool, baseball commentators, strategic teams, and fantasy baseball users can efficiently search up the pitcher's statistical data. Therefore, narrating the game, strategizing, and making profit from fantasy baseball much more systematic. The main goal of this project is to create a web application that shows three graphs that explains the pitch-type signatures by frequency, velocity, and horizontal/vertical break.

Section 3: Testing of the Web App

Preparation

First, in order to make the data retrieval process much more efficient, the `statcast` dataset from 2008-01-01 to 2021-05-31 was stored locally as a `.csv` file. The code used to save the dataframe is shown below:

In [ ]:
```
# statcast_df = statcast(start_dt = '2008-01-01', end_dt = '2021-05-31')
# statcast_df.to_csv(r'/Users/bryankim/Desktop/Spring2021/stat430/final/sta
```

Since saving this dataset locally meant that the arguments `start_dt` and `end_dt` could not be used, the `game_date` column was formatted into `datetime`. Following code is shown below:

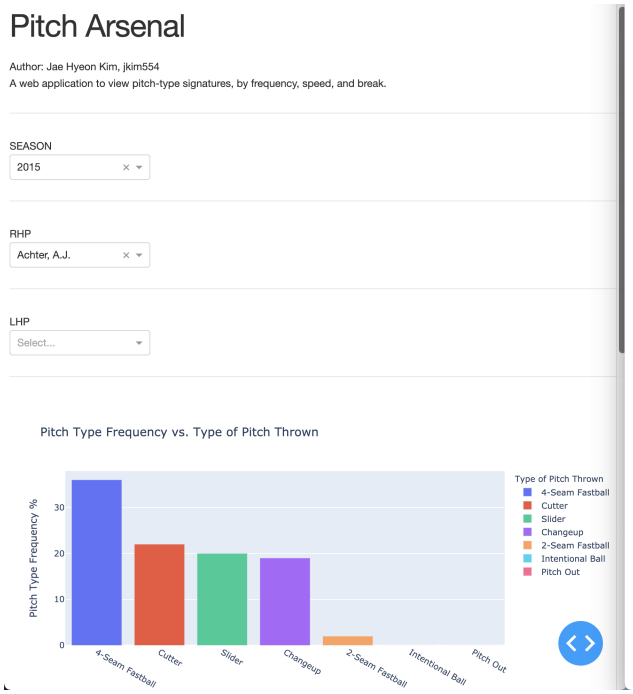
In [ ]:
```
# statcast_df['game_date'] = pd.to_datetime(statcast_df['game_date'], forma
```

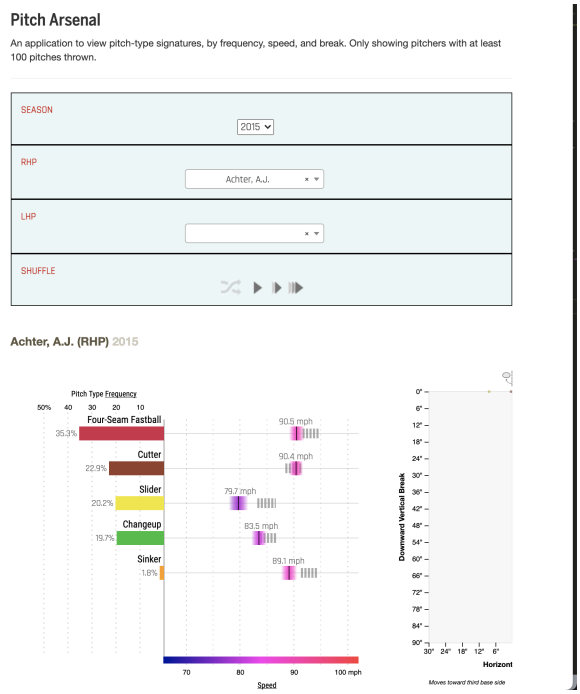
Created a list of years from 2008 to 2021 and stored it inside the variable, `yr`. Then used this list to create a drop-down menu for `Season`

In [ ]:
```
# yr = ([i for i in range (2008, 2022)])
```

Testing

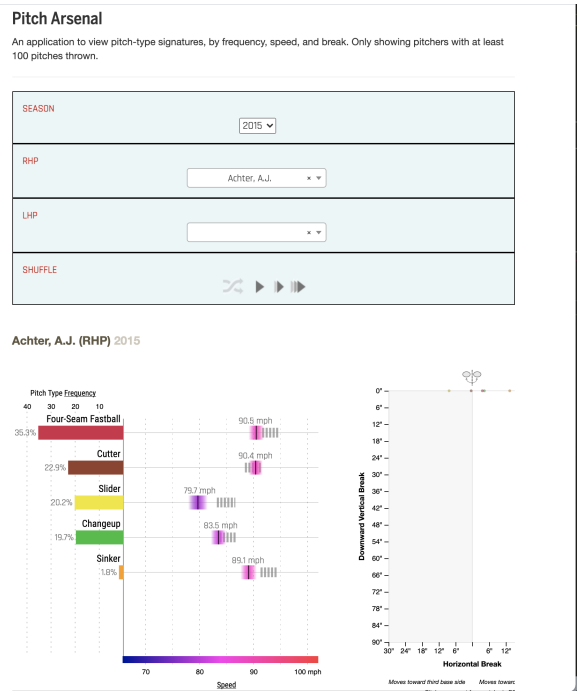
1.


Pitch Arsenal
Author: Jae Hyeon Kim, jkim554
A web application to view pitch-type signatures, by frequency, speed, and break.
SEASON
2015
RHP
Achter, A.J.
LHP
Select...
Pitch Type Frequency vs. Type of Pitch Thrown
Pitch Type Frequency %
Type of Pitch Thrown
4-Seam Fastball
Cutter
Slider
Changeup
2-Seam Fastball
Intentional Ball
Pitch Out

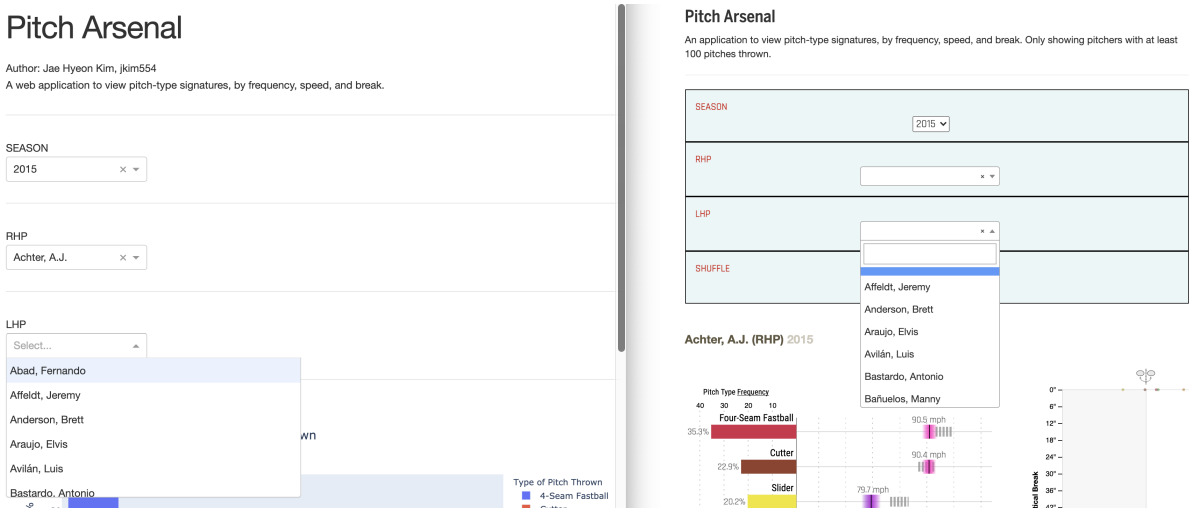

Pitch Arsenal
An application to view pitch-type signatures, by frequency, speed, and break. Only showing pitchers with at least 100 pitches thrown.
SEASON
2015
RHP
Achter, A.J.
LHP
SHUFFLE
Achter, A.J. (RHP) 2015
Pitch Type Frequency
Four-Seam Fastball
Cutter
Slider
Changeup
Sinker
Speed
Downward Vertical Break
Horizontal Break

2.


Type of Pitch Thrown
Release Speed vs. Type of Pitch Thrown
Release Speed (mph)
Type of Pitch Thrown
Downward Vertical Break vs. Horizontal Break of Pitch
Downward Vertical Break (inches)
Horizontal Break (inches)
color
4-Seam Fastball
Cutter
Slider
Changeup
2-Seam Fastball
Intentional Ball
Pitch Out


Pitch Arsenal
An application to view pitch-type signatures, by frequency, speed, and break. Only showing pitchers with at least 100 pitches thrown.
SEASON
2015
RHP
Achter, A.J.
LHP
SHUFFLE
Achter, A.J. (RHP) 2015
Four-Seam Fastball
Cutter
Slider
Changeup
Sinker
Speed
Downward Vertical Break
Horizontal Break

3.



Above three images show examples of how I tested my web application.

1. The first image shows that the web app successfully takes the name input with the names that have abbreviations such as "A.J.", which it was not able to before. This function was implemented by taking in the input name from either RHP or LHP drop-down menu, then finding a matching name (`player_name`) within the dataset pulled from `statcast` function, then finally finding the associated pitcher id (`pitcher`). Then by using the pitcher id, the pitcher data was pulled by using the `statcast_pitcher` function.
2. The second image shows that the box-plot graph of my web app has very similar values to the box-plot graph created by Savant.
3. The third image shows that the drop-down menu options of my web application is very similar to the drop-down menu options created by Savant. On my web application, the options are the same except for the first option which is not included in the web application created by Savant.

Section 4: Conclusion & Discussion

What I did well:

- What I did well on this web app is that the web application runs successfully without errors. Additionally, I was able to make the RHP and LHP menus update after user selects a certain season. I was also able to successfully execute the part where only one pitcher can be selected from the drop-down menus (either RHP or LHP). Also, the three graphs are visually easy to comprehend. The order of the pitch names that appear on the graphs are all in the same order as well as the legends. Furthermore, the pitch names are ordered in a descending order by pitch type frequency percentage. As a result, eventhough there may be three different graphs, the users can easily understand the statistical data.

What could be improved:

- Due the fact that this is the first experience with dash, there could be some improvements that could be made. My web application lacks the visual aesthetic features from the one Savant provides. Additionally, the formatting of the graphs and the type of graphs can be altered so that the web application can provide data in a condensed and simple manner. If someone is to continue working on my project, they should try to add the "Shuffle" option under the LHP drop-down menu. Also, they should try to combine the bar graph with the box-plot graph to

make the graphs look more similar to the ones shown by Savant. They could also try and tackle adding "League Avg", "Opacity = Frequency" checkboxes, which was not required for this project.

In [ ]: