Evaluating Jacob deGrom's Pitching Dynamics

A Five-Year Analysis

Ethan Dang, Maggie Feng, Bao Hu, Jianheng Luo

Fushing Hsieh

STA 160

June 11, 2024

I. Introduction

Nowadays, baseball is one of the most popular sports in the world. As the systems and technologies supporting the sport gradually improve and matures, the strategies for winning games have become more sophisticated and data-driven. One key aspect of a successful baseball team is the performance of its pitchers. The art of pitching is a complex interaction of various factors that significantly impact a pitcher's ability to throw a strike. Understanding these factors is essential to improving performance and developing an effective pitching strategy. In addition, mostly speed and spin rate is one of the most important factors affecting the counts of strikeout. Our report focuses on Jacob deGrom, one of the premier starting pitchers in Major League Baseball, over a five-year period. This project aims to explore deGrom's performance improvements from 2015 to 2019, and analyze how specific variables-release speed and release spin rate change affect deGrom's probability of throwing a strike.

By leveraging the Statcast database on the MLB website, we aim to delve deeper into various aspects of deGrom's pitching performance. This includes examining his strikeout and pitchout ratios from 2015 to 2019 to determine if his accuracy and effectiveness have improved. A higher strikeout ratio and lower pitchout ratio are indicators of good pitching performance and can significantly impact the outcome of games.

We will also analyze the increase and decrease in the percentage of each pitch type over the five years, identify the pitch type with the best ratio, and analyze that pitch type through a breakout analysis to see if that pitch type has become more accurate. Additionally, we will explore the strikeout and pitchout ratios of each pitch type over the five years to better understand deGrom's pitching accuracy and effectiveness.

After analyzing each pitch type, we select the one pitch type with the highest accuracy and effectiveness, analyzing velocity and spin rate to see if these two variables affect the number of strikeouts and DeGrom's performance.

Nonetheless, through this study, we aim to gain a comprehensive understanding of deGrom’s pitching performance, highlighting trends and improvements that can inform future performance predictions and coaching strategies. Our analysis not only helps understand individual player performance, but also contributes to the broader field of sports analytics and its application in modern baseball.

II. Data Description

The dataset for this study is extracted from the Major League Baseball Statcast database, which employs advanced tracking technology to collect detailed data on every pitch thrown in the MLB. This rich dataset enables a deep dive into numerous aspects of a pitch and a game, providing insights that were not previously accessible.

Here are the variables we have included in our analysis:

- **Pitch_type:** The type of pitch derived from Statcast.
- **Pitch_name:** The name of the pitch derived from the Statcast Data.
- **Release_speed**: Pitch velocities
- **Release_spin_rate:** Rate of spin of ball after it is released.
- **Type:** Shorthand of pitch result. B = ball, S = strike, X = in play.
- **Balls:** Pre-pitch number of balls in count.
- **Strikes:** Pre-pitch number of strikes in count.
- **Pfx_x :** Horizontal movement in feet from the catcher's perspective.
- **Pfx_z:** Vertical movement in feet from the catcher's perspective.
- **Game_date:** Date of the Game

Each of these metrics plays a significant role in our analysis, helping us to dissect the intricacies of pitching strategies and their effectiveness. By understanding the nuances of each pitch, including its speed, spin, and movement, we can gain deeper insights into Jacob deGrom’s pitching mechanics over the examined five-year period. This comprehensive dataset not only enhances our analysis of individual pitches but also enriches our understanding of broader trends and patterns in pitching performance.

III. Methodology

To conduct our study on Jacob deGrom's pitching performance from 2015 to 2019, we utilized the ‘pybaseball’ library, a comprehensive tool for baseball data analysis. Initially, we identified deGrom’s player ID using the ‘playerid_lookup’ function, which allowed us to accurately retrieve his pitching data through the ‘statcast_pitcher’ function. This function was instrumental in specifying time ranges for targeted data extraction, focusing on regular season games to maintain consistency in our dataset.

Once the data was gathered, our next step was to cleanse and preprocess it to ensure accuracy and relevance for analysis. We removed any entries with missing values, particularly those lacking critical metrics like pfx_x and pfx_z, which represent the horizontal and vertical movement of pitches. Additionally, we converted these measurements from feet to inches to standardize the data and added a column indicating the year of each game, which facilitated temporal analysis.

Our analysis began by examining the variations in pfx_x and pfx_z across different pitch types, employing scatterplots to visualize these changes over the years. This visual representation helped highlight trends and deviations in pitching dynamics, offering insights into deGrom’s technical adjustments and effectiveness.

Further, we analyzed the ratio of bad balls (categorized as 'B' for balls and 'X' for in-play balls) to strikes. This metric provided a deeper understanding of deGrom's pitch control and strategic execution under varying game conditions. By tracking these ratios annually, we could assess improvements or declines in his performance, shedding light on his development as a pitcher.

To deepen our investigation, we explored the relationship between release speed and spin rate—two pivotal factors in pitch effectiveness. We performed a correlation analysis to determine how these variables interacted, revealing a moderate positive correlation that suggests higher release speeds tend to accompany higher spin rates, particularly evident in deGrom’s 4-Seam Fastball. These findings were visualized through scatterplots that mapped the release speed against the spin rate, further illustrating the consistency and variations in his pitching mechanics over time.

By integrating these analyses, our methodology not only underscores the detailed evaluation of deGrom’s pitching prowess but also enhances our understanding of how specific changes in pitching mechanics can influence overall performance. This comprehensive approach is aimed at providing actionable insights that could inform future coaching strategies and player development within baseball analytics.

IV. Result and Discussion

IV.I Overall Improvement

Year	Bad Ball Ratio	Strike Ratio
2015	0.490087	0.509913
2016	0.521810	0.478190
2017	0.493272	0.506728
2018	0.469075	0.530925
2019	0.475434	0.524566

Table 1. Ratio of Ball Balls and Strikes through the years.

In Table 1, we can see the bad ball ratio and strike ratio for deGrom over a five-year period. In 2015, the strike ratio was around 0.5, and in the next two years, the strike ratio fluctuated around this stage. Until 2018, the strike ratio gradually increased and finally stabilized at around 5.2 to 5.3. Overall, the data shows a trend of improvement in deGrom's pitching performance from 2015 to 2019. After a dip in 2016, there is a clear recovery and subsequent improvement in the following years.

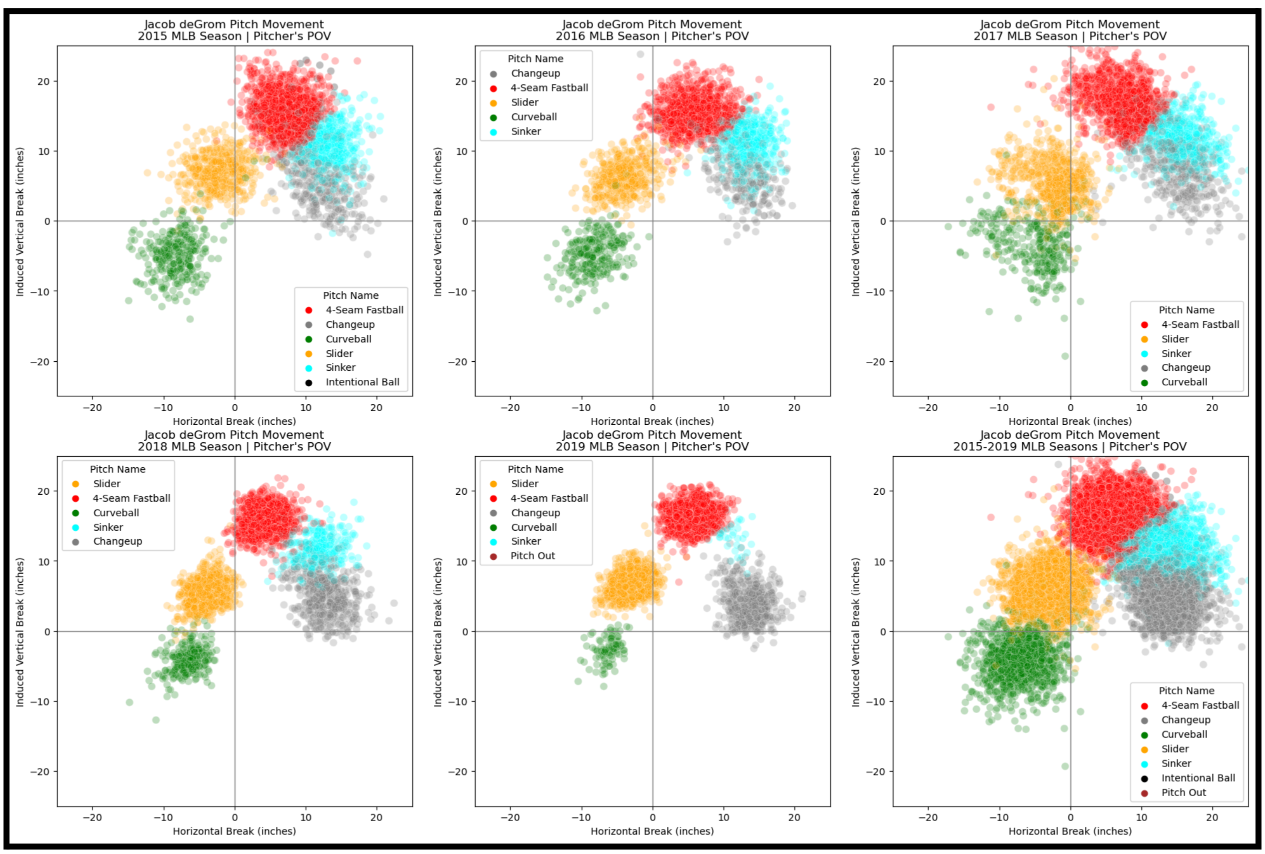

Figure 1. Scatter plots of Pitch Types by pfx_x (horizontal movement) and pfx_z (vertical movement).

Our observations show that as the spread of each of Jacob deGrom's main pitch types (four-seam fastball, slider, changeup) has continued to shrink over the years, it indicates that his control and technique have improved.

The chart also shows that among all pitch types, the sinker and curveball are clearly less frequent. Relatively speaking, the four-seam fastball has been used more and more frequently and has maintained a relatively high strike rate over the years. This consistent performance makes it a reliable pitch that contributes to its widespread use in his pitching strategy.

In 2017, we noticed that deGrom's pitching spread increased compared to 2015 and 2016, but tightened again in 2018 and 2019. This shows that his pitching technique has improved over the years and maintained consistency. The widening spread in 2017 suggests that his pitching style went through a period of adjustment or experimentation, but he honed his pitching skills in the following years.

Additionally, in 2018, deGrom's strike ratio for his curveball exhibited a notable increase. This improvement is evident in the 2018 scatter plot, which displays a narrower scatter range for curveballs compared to previous years. The reduced scatter range indicates enhanced technical precision in deGrom's pitching mechanics. The correlation between the narrower scatter range and the increased strike ratio suggests that this refined precision contributed significantly to the improved performance of his curveball.

As a result, we can conclude that deGrom's overall pitching skills have shown a continuous improvement trajectory from 2015 to 2019. During this period, the accuracy of his various pitch types has seen significant enhancement. deGrom also appears to have progressively identified the pitch types he excels at and those he does not, allowing him to refine his pitching strategy and focus on his strengths.

To gain a deeper understanding of deGrom's improvement trends, the next step of our research will involve a detailed analysis of each of his pitch types. By comparing the performance and precision of different pitch types, we aim to identify specific areas of improvement and further elucidate the factors contributing to deGrom's overall development as a pitcher.

IV.II Pitch Type and Strike Ratios

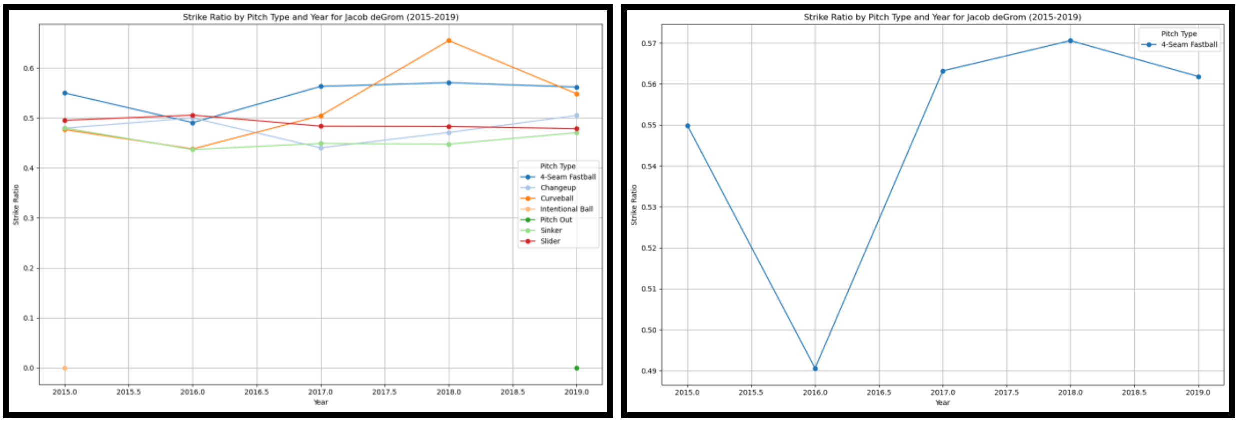

Figure 2. Line plot of all strike ratio by pitch type. Line plot of 4-Seam Fastball strike ratio.

Pitch Name	2015	2016	2017	2018	2019	Cumulative
4-Seam Fastball	Ball: 467	Ball: 352	Ball: 403	Ball: 400	Ball: 491	Ball: 2113
	Strike: 854	Strike: 495	Strike: 704	Strike: 784	Strike: 863	Strike: 3700
	Total: 1553	Total: 1009	Total: 1250	Total: 1374	Total: 1536	Total: 6722
	Ratio: 0.5499	Ratio: 0.4906	Ratio: 0.5632	Ratio: 0.5706	Ratio: 0.5618	Ratio: 0.5504
Changeup	Ball: 133	Ball: 67	Ball: 173	Ball: 182	Ball: 166	Ball: 684
	Strike: 199	Strike: 112	Strike: 393	Strike: 243	Strike: 255	Strike: 982
	Total: 415	Total: 224	Total: 393	Total: 516	Total: 505	Total: 2053
	Ratio: 0.4801	Ratio: 0.5000	Ratio: 0.4402	Ratio: 0.4709	Ratio: 0.5049	Ratio: 0.4783
Curveball	Ball: 131	Ball: 95	Ball: 105	Ball: 59	Ball: 51	Ball: 413
	Strike: 174	Strike: 113	Strike: 160	Strike: 167	Strike: 84	Strike: 665
	Total: 365	Total: 258	Total: 317	Total: 255	Total: 93	Total: 1288
	Ratio: 0.4767	Ratio: 0.4379	Ratio: 0.5047	Ratio: 0.6549	Ratio: 0.5484	Ratio: 0.5163
Sinker	Ball: 182	Ball: 143	Ball: 156	Ball: 94	Ball: 14	Ball: 589
	Strike: 273	Strike: 183	Strike: 224	Strike: 132	Strike: 16	Strike: 828
	Total: 569	Total: 419	Total: 499	Total: 295	Total: 34	Total: 1816
	Ratio: 0.4798	Ratio: 0.4368	Ratio: 0.4489	Ratio: 0.4475	Ratio: 0.4706	Ratio: 0.4559
Slider	Ball: 172	Ball: 147	Ball: 233	Ball: 262	Ball: 344	Ball: 1158
	Strike: 261	Strike: 224	Strike: 339	Strike: 372	Strike: 489	Strike: 1685
	Total: 527	Total: 443	Total: 701	Total: 770	Total: 1022	Total: 3463
	Ratio: 0.4953	Ratio: 0.5056	Ratio: 0.4836	Ratio: 0.4831	Ratio: 0.4785	Ratio: 0.4866

Table 2. Pitch Data for Jacob deGrom (2015-2019)

Analyzing deGrom's pitch data from 2015 to 2019, we can observe several trends that give insight into his pitching strategy and effectiveness. Each pitch type's performance over the years is quantified by its strike ratio, which is critical in assessing deGrom's evolution as a pitcher.

Starting with the 4-Seam Fastball, which is recognized as deGrom's best pitch based on its high strike ratio, we see a consistent number of strikes compared to balls thrown across the years. From 2015 to 2019, the strike ratio improved, peaking in 2018 with a ratio of 0.570597. It remained strong in 2019, showcasing its reliability and deGrom's ability to maintain performance.

The Changeup also shows effective usage, with the strike ratio slightly fluctuating over the years. It maintained a fairly consistent performance, indicating deGrom's control over this pitch, particularly in 2017 and 2019 with strike ratios above 0.5.

The Curveball presents an increasing trend in its effectiveness, especially noted in 2018 with a strike ratio of 0.654902, the highest among all his pitches that year. This suggests that deGrom had been particularly successful with the curveball during this period, but saw a decrease in 2019.

The Sinker shows variability in its usage and effectiveness, with a significant improvement in strike ratio in 2018. Despite this improvement, there was a drop in the total number of sinkers thrown by 2019. As a result, this could suggest a strategic shift in deGrom's approach or an adjustment based on the pitch's previous performance.

For the Slider, there was a notable increase in strike ratio in 2018, followed by a slight drop in 2019. However, the slider remained a strong pitch for deGrom, as shown by the consistently high number of strikes thrown.

Nonetheless, the data clearly supports the 4-Seam Fastball as deGrom's most reliable pitch over the years, marked by a strong and stable strike ratio. His mastery of this pitch, coupled with strategic improvements in the curveball and slider, highlight his development as a pitcher capable of adjusting his approach and refining his skills effectively. Analyzing these trends helps us understand the critical aspects of pitch selection and execution, crucial for developing future pitching strategies and training methods.

IV.III Release Speed and Release Spin Rate

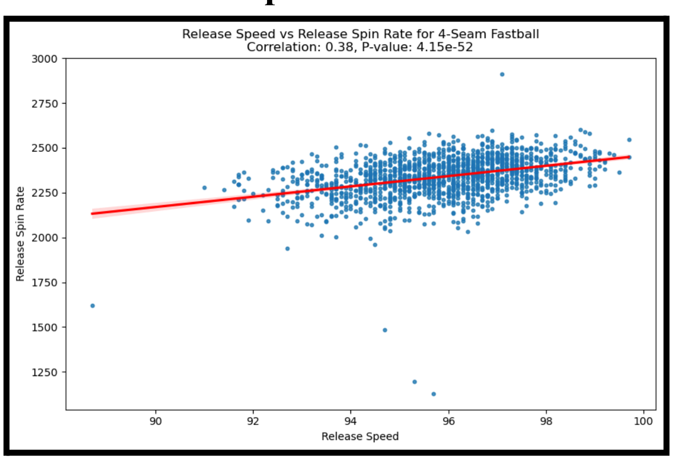

Figure 3. Correlation of release speed and release spin rate.

Observing Figure 3, generally, as deGrom's release speed increases, his spin rate also tends to increase, thus indicating there is a positive correlation between the two features. However, the correlation is moderate with a value of 0.38, suggesting other features influence deGrom's spin rate. Additionally, Figure 3 illustrates the range of deGrom's 4-Seam Fastball pitches. His release speed typically lies between 92 and 99 miles per hour (mph), with a concentration between 94 and 98 miles per hour. His release spin rate falls in the range of 2000 and 2500 revolutions per minute (rpm). This range highlights deGrom's ability to throw the fastball at varying velocities and spin rates, which can be useful in future pitching strategies to mislead batters.

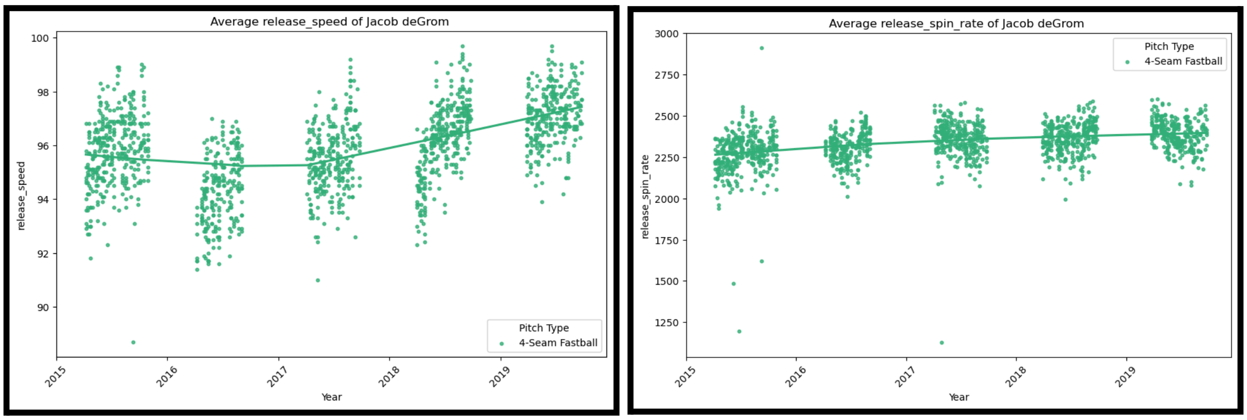

Figure 4. Scatterplots of release speed and release spin rate.

Analyzing the left scatter plot of average release speed, we see the fluctuations in deGrom's release speed from 2015 to 2019. We observe a small decrease from 2015 to 2016, creating a dip in 2017, then a slight upward trend from 2017 to 2019. Besides the dip from 2016 to 2017, deGrom's average release speed is around 92 to 98 mph. After the dip in 2017, deGrom has been increasing his fastball release speed, peaking at 2019 with around 99 mph, showing a consistent improvement in his fastball pitch. Additionally, the clusters within each year have a moderate spread and few outliers, indicating that deGrom consistently throws his 4-seam fastball between a certain range within each year.

Studying the right scatter plot of the average release spin rate, we see a slight increase in deGrom's spin rate throughout the years. Most of his release spin rate lies in the range of 2000 and 2500 rpm, with a few outliers around 1250 and 1750 rpm. Most of his outliers occurred in 2015, meaning those pitches could have been experimental or an unusual pitch error. As the years increase, we see there are fewer and fewer outliers, indicating a regular release spin rate. Furthermore, the clusters are dense and have less spread, showing his 4-seam fastball pitches are stable.

As deGrom's release speed and release spin rate increase, his fastball pitching will improve and will become more effective. Notice the plots from Figure 4, deGrom has a reliable foundation in his 4-seam fastball pitch. While he still has minor deviations in both features, his overall fastball pitch is improving and following an upward trend. By understanding these patterns and outliers, deGrom can strategize and optimize his fastball pitches to enhance effectiveness and consistency.

V. Conclusion

Nonetheless, this comprehensive study of Jacob deGrom's pitching performance from 2015 to 2019 has clarified significant insights into the factors contributing to his effectiveness on the mound. Over the five-year period, deGrom demonstrated marked improvements in control and precision, as evidenced by the increasing strike ratios and refined pitch dynamics observed in our analysis. Notably, the 4-Seam Fastball emerged as a consistently reliable pitch, maintaining high strike ratios and showing gradual enhancements in release speed and spin rate. The Curveball and Slider also reflected significant advancements, contributing to a diversified and effective pitching arsenal. Our findings affirm that accurate monitoring and analysis of pitch-specific metrics can yield critical insights into a pitcher's development and strategic adjustments. Future strategies could benefit from focusing on these metrics to further enhance pitching performance. The data-driven approach adopted in this study not only underscores the value of advanced sports analytics, but also sets a benchmark for ongoing improvements in coaching techniques and player development within baseball.

Appendix:

```
import pandas as pd
import numpy as np
import pybaseball as pyb
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr

pyb.cache.enable()

features_to_keep_main = [
    'pitch_type',
    'game_date',
    'release_speed',
    'release_pos_x',
    'release_pos_z',
    'player_name',
    'pitch_name',
    'events',
    'zone',
    'balls',
    'strikes',
    'game_year',
    'pfx_x',
    'pfx_z',
    'plate_x',
    'plate_z',
    'outs_when_up',
    'inning',
    'inning_topbot',
    'release_spin_rate',
    'release_extension',
    'delta_home_win_exp',
    'delta_run_exp'
]

start_time_main = '2015-04-01'
end_time_main = '2020-07-15'


def get_pitchers_info(firstname, lastname, features_to_keep=features_to_keep_main,
start_time=start_time_main, end_time=end_time_main):
    player_info = pyb.playerid_lookup(lastname, firstname)

    if player_info.empty:
        raise ValueError(f"No player found for name: {firstname} {lastname}")

    player_id = player_info['key_mlbam'].iloc[0]
    print(f'Pitcher ID: {player_id}')
    data = pyb.statcast_pitcher(start_time, end_time, player_id=player_id)
    filtered_data = data[features_to_keep]
    filtered_data = filtered_data.dropna()

    earlist = pd.to_datetime(sorted(filtered_data.game_date.unique())[0])
```

```
    latest = pd.to_datetime(sorted(filtered_data.game_date.unique())[-1])
    print(f'Loaded data for pitcher {firstname} {lastname} from {earlist} to
{latest}')
    print(f'with {filtered_data.shape[0]} data points and {filtered_data.shape[1]}
features')
    print()
    return filtered_data


def plot(df, columnName, playerName, pitch_type_filter=None):
    df['game_date'] = pd.to_datetime(df['game_date'])
    df['date_num'] = df['game_date'].map(pd.Timestamp.toordinal)
    df['year'] = df['game_date'].dt.year

    plt.figure(figsize=(10, 6))

    # Define a color palette
    unique_pitch_types = df['pitch_name'].unique()
    palette = sns.color_palette("husl", len(unique_pitch_types))

    # Create a dictionary to map each pitch type to a color
    color_map = {pitch_type: color for pitch_type, color in zip(unique_pitch_types,
palette)}

    if pitch_type_filter:
        # Filter dataframe for the specified pitch type
        df = df[df['pitch_name'] == pitch_type_filter]
        unique_pitch_types = [pitch_type_filter]

    # Plot each pitch type separately
    for pitch_type in unique_pitch_types:
        subset = df[df['pitch_name'] == pitch_type]
        sns.regplot(x='date_num', y=columnName, data=subset, scatter_kws={'s': 10,
'color': color_map[pitch_type]}, line_kws={'color': color_map[pitch_type]},
lowess=True, label=pitch_type)

    ax = plt.gca()
    # Set x-ticks to years
    years = df['year'].unique()
    ax.set_xticks([pd.Timestamp(f'{year}-01-01').toordinal() for year in years])
    ax.set_xticklabels(years, rotation=45)

    plt.xlabel('Year')
    plt.ylabel(columnName)
    plt.title(f'Average {columnName} of {playerName}')
    plt.legend(title='Pitch Type')

    plt.show()


def check_correlation(df, speed_column, spin_column, pitch_type_filter=None):

    if pitch_type_filter:
        df = df[df['pitch_name'] == pitch_type_filter]
```

```
    correlation, p_value = pearsonr(df[speed_column], df[spin_column])

    print(f'Pearson correlation coefficient: {correlation}')
    print(f'P-value: {p_value}')

    plt.figure(figsize=(10, 6))
    sns.regplot(x=speed_column, y=spin_column, data=df, scatter_kws={'s': 10},
line_kws={'color': 'red'})

    plt.xlabel('Release Speed')
    plt.ylabel('Release Spin Rate')
    plt.title(f'Release Speed vs Release Spin Rate for {pitch_type_filter if
pitch_type_filter else "All Pitches"}\nCorrelation: {correlation:.2f}, P-value:
{p_value:.2e}')

    plt.show()


firstname = 'Jacob'
lastname = 'deGrom'
player_name = 'Jacob deGrom'
pat_df = get_pitchers_info(firstname, lastname)


plot(pat_df, 'release_spin_rate', player_name, '4-Seam Fastball')
plot(pat_df, 'release_speed', player_name, '4-Seam Fastball')
check_correlation(pat_df, 'release_speed', 'release_spin_rate',
pitch_type_filter='4-Seam Fastball')


def plot_pitch_movement(pat_df, years):
    all_data = []

    # Create subplots for individual year scatter plots
    fig_scatter, axes_scatter = plt.subplots(2, 3, figsize=(18, 12))
    axes_scatter = axes_scatter.flatten()

    for i, year in enumerate(years):
        # Filter data for each year
        yearly_data = pat_df[pat_df['game_year'] == year].copy()

        # Clean and transform the data
        yearly_data.loc[:, 'pfx_x_in_pv'] = -12 * yearly_data['pfx_x']
        yearly_data.loc[:, 'pfx_z_in'] = 12 * yearly_data['pfx_z']
        yearly_data.loc[:, 'year'] = year
        all_data.append(yearly_data)

        # Create dynamic pitch colors
        unique_pitches = yearly_data['pitch_name'].unique()
        pitch_colors = {
            "4-Seam Fastball": "red",
            "2-Seam Fastball": "blue",
            "Sinker": "cyan",
            "Cutter": "violet",
            "Fastball": "black",
```

```
            "Curveball": "green",
            "Knuckle Curve": "purple",
            "Slider": "orange",
            "Changeup": "#7f7f7f",
            "Split-Finger": "beige",
            "Knuckleball": "gold",
            "Intentional Ball": "black",
            "Pitch Out": "brown"
        }

        # Plot pitch movement for each year
        sns.scatterplot(ax=axes_scatter[i], data=yearly_data, x='pfx_x_in_pv',
y='pfx_z_in', hue='pitch_name', palette=pitch_colors, alpha=0.25, s=60)
        axes_scatter[i].axvline(0, color='gray', linewidth=1)
        axes_scatter[i].axhline(0, color='gray', linewidth=1)
        axes_scatter[i].set_xlim(-25, 25)
        axes_scatter[i].set_ylim(-25, 25)
        axes_scatter[i].set_aspect('equal', adjustable='box')
        axes_scatter[i].set_title(f'Jacob deGrom Pitch Movement\n{year} MLB Season |
Pitcher\'s POV')
        axes_scatter[i].set_xlabel('Horizontal Break (inches)')
        axes_scatter[i].set_ylabel('Induced Vertical Break (inches)')
        axes_scatter[i].legend(title='Pitch Name')

    # Combine all years data
    cumulative_data = pd.concat(all_data)

    # Plot cumulative pitch movement in the last subplot
    sns.scatterplot(ax=axes_scatter[-1], data=cumulative_data, x='pfx_x_in_pv',
y='pfx_z_in', hue='pitch_name', palette=pitch_colors, alpha=0.25, s=60)
    axes_scatter[-1].axvline(0, color='gray', linewidth=1)
    axes_scatter[-1].axhline(0, color='gray', linewidth=1)
    axes_scatter[-1].set_xlim(-25, 25)
    axes_scatter[-1].set_ylim(-25, 25)
    axes_scatter[-1].set_aspect('equal', adjustable='box')
    axes_scatter[-1].set_title('Jacob deGrom Pitch Movement\n2015-2019 MLB Seasons |
Pitcher\'s POV')
    axes_scatter[-1].set_xlabel('Horizontal Break (inches)')
    axes_scatter[-1].set_ylabel('Induced Vertical Break (inches)')
    axes_scatter[-1].legend(title='Pitch Name')

    plt.tight_layout()
    plt.show()

def fetch_degrom_data():
    # Fetch the player's ID
    player_lookup = pyb.playerid_lookup('deGrom', 'Jacob')
    degrom_id = player_lookup['key_mlbam'].values[0]

    # Define years to loop through
    years = [2015, 2016, 2017, 2018, 2019]
    all_data = []

    # Initialize an empty DataFrame to store cumulative results
    cumulative_pitch_counts = pd.DataFrame()
```

```
    for year in years:
        # Fetch the data for each year
        degrom_data = pyb.statcast_pitcher(f'{year}-03-01', f'{year}-12-01',
degrom_id)

        # Clean and transform the data
        degrom_cleaned_data = degrom_data.dropna(subset=['pitch_name', 'type'])

        # Calculate ball and strike counts per pitch type
        pitch_counts = degrom_cleaned_data.groupby(['pitch_name',
'type']).size().unstack(fill_value=0)
        pitch_counts['Total Count'] = pitch_counts.sum(axis=1)
        pitch_counts['Strike Ratio'] = pitch_counts['S'] / pitch_counts['Total Count']
        pitch_counts.reset_index(inplace=True)

        # Add year information
        pitch_counts['Year'] = year
        all_data.append(pitch_counts)

        # Append to cumulative results using pd.concat
        cumulative_pitch_counts = pd.concat([cumulative_pitch_counts, pitch_counts],
ignore_index=True)

    # Combine all years data
    cumulative_data = pd.concat(all_data)

    return cumulative_pitch_counts, cumulative_data



years = [2015, 2016, 2017, 2018, 2019]

plot_pitch_movement(pat_df, years)


def fetch_degrom_data():
    # Fetch the player's ID
    player_lookup = pyb.playerid_lookup('deGrom', 'Jacob')
    degrom_id = player_lookup['key_mlbam'].values[0]

    # Define years to loop through
    years = [2015, 2016, 2017, 2018, 2019]
    all_data = []

    # Initialize an empty DataFrame to store cumulative results
    cumulative_pitch_counts = pd.DataFrame()

    for year in years:
        # Fetch the data for each year
        degrom_data = pyb.statcast_pitcher(f'{year}-03-01', f'{year}-12-01',
degrom_id)

        # Clean and transform the data
        degrom_cleaned_data = degrom_data.dropna(subset=['pitch_name', 'type'])
```

```
        # Calculate ball and strike counts per pitch type
        pitch_counts = degrom_cleaned_data.groupby(['pitch_name',
'type']).size().unstack(fill_value=0)
        pitch_counts['Total Count'] = pitch_counts.sum(axis=1)
        pitch_counts['Strike Ratio'] = pitch_counts['S'] / pitch_counts['Total Count']
        pitch_counts.reset_index(inplace=True)

        # Add year information
        pitch_counts['Year'] = year
        all_data.append(pitch_counts)

        # Append to cumulative results using pd.concat
        cumulative_pitch_counts = pd.concat([cumulative_pitch_counts, pitch_counts],
ignore_index=True)

    # Combine all years data
    cumulative_data = pd.concat(all_data)

    return cumulative_pitch_counts, cumulative_data


def plot_strike_ratio(cumulative_pitch_counts, pitch_type_filter=None):
    # Prepare data for plotting Strike Ratio by year
    plot_data = cumulative_pitch_counts[['pitch_name', 'Strike Ratio', 'Year']]

    # Apply pitch type filter if provided
    if pitch_type_filter:
        plot_data = plot_data[plot_data['pitch_name'] == pitch_type_filter]
        plot_data_pivot = plot_data.pivot(index='Year', columns='pitch_name',
values='Strike Ratio')
    else:
        # Pivot the data to get pitch types as columns
        plot_data_pivot = plot_data.pivot(index='Year', columns='pitch_name',
values='Strike Ratio')

    # Define a consistent color palette
    pitch_types = plot_data_pivot.columns
    colors = sns.color_palette('tab20', len(pitch_types))
    color_map = dict(zip(pitch_types, colors))

    # Create plot
    plt.figure(figsize=(12, 8))
    for column in plot_data_pivot.columns:
        plt.plot(plot_data_pivot.index, plot_data_pivot[column], marker='o',
label=column, color=color_map[column])

    plt.title('Strike Ratio by Pitch Type and Year for Jacob deGrom (2015-2019)')
    plt.xlabel('Year')
    plt.ylabel('Strike Ratio')
    plt.legend(title='Pitch Type')
    plt.grid(True)

    plt.tight_layout()
    plt.show()
```

```
# Fetch data
cumulative_pitch_counts, cumulative_data = fetch_degrom_data()

# Plot Strike Ratio without filter
plot_strike_ratio(cumulative_pitch_counts)

# Plot Strike Ratio with filter
plot_strike_ratio(cumulative_pitch_counts, pitch_type_filter='4-Seam Fastball')



player_lookup = pyb.playerid_lookup('deGrom', 'Jacob')
degrom_id = player_lookup['key_mlbam'].values[0]

years = [2015, 2016, 2017, 2018, 2019]
overall_ratios = []

for year in years:
    degrom_data = pyb.statcast_pitcher(f'{year}-03-01', f'{year}-12-01', degrom_id)

    degrom_cleaned_data = degrom_data.dropna(subset=['type'])

    total_counts = degrom_cleaned_data['type'].value_counts()
    bad_ball_count = total_counts.get('B', 0) + total_counts.get('X', 0)
    strike_count = total_counts.get('S', 0)
    total_count = bad_ball_count + strike_count

    bad_ball_ratio = bad_ball_count / total_count if total_count != 0 else 0
    strike_ratio = strike_count / total_count if total_count != 0 else 0

    overall_ratios.append({
        'Year': year,
        'Bad Ball Ratio': bad_ball_ratio,
        'Strike Ratio': strike_ratio
    })

overall_ratios_df = pd.DataFrame(overall_ratios)

print("Overall Bad Ball and Strike Ratios by Year")
print(overall_ratios_df)


all_data = []


cumulative_pitch_counts = pd.DataFrame()

for year in years:
    degrom_data = pyb.statcast_pitcher(f'{year}-03-01', f'{year}-12-01', degrom_id)

    degrom_cleaned_data = degrom_data.dropna(subset=['pitch_name', 'type'])

    pitch_counts = degrom_cleaned_data.groupby(['pitch_name',
'type']).size().unstack(fill_value=0)
```

```
    pitch_counts['Total Count'] = pitch_counts.sum(axis=1)
    pitch_counts['Bad Ball Ratio'] = (pitch_counts['B'] + pitch_counts['X']) /
pitch_counts['Total Count']
    pitch_counts['Strike Ratio'] = pitch_counts['S'] / pitch_counts['Total Count']
    pitch_counts.reset_index(inplace=True)

    pitch_counts.columns.name = None
    pitch_counts.rename(columns={'B': 'Ball Count', 'S': 'Strike Count'},
inplace=True)

    pitch_counts['Year'] = year
    all_data.append(pitch_counts)

    print(f"Ball and Strike Ratio for {year}")
    print(pitch_counts)
    print("\n")

    cumulative_pitch_counts = pd.concat([cumulative_pitch_counts, pitch_counts],
ignore_index=True)

cumulative_counts = cumulative_pitch_counts.groupby(['pitch_name']).sum()
cumulative_counts['Bad Ball Ratio'] = (cumulative_counts['Ball Count'] +
cumulative_counts['X']) / cumulative_counts['Total Count']
cumulative_counts['Strike Ratio'] = cumulative_counts['Strike Count'] /
cumulative_counts['Total Count']
cumulative_counts.reset_index(inplace=True)

print("Cumulative Ball and Strike Ratio from 2015 to 2019")
print(cumulative_counts)
```