Can You Stop Elly De La Cruz from Stealing Second Base?

In this notebook, we aim to find a relationship between stolen base metrics and stolen base output for Elly De La Cruz, and eventually find ways to stop Elly De La Cruz from stealing second base. This notebook simply contains our code used to draw our conclusions. We will expand much more on our methodology and results in the PDF that comes with this code.

1. Imports

Here, we import the data that we need to start our analysis. We used data tables from Baseball Savant, so we need to load in datasets as well as some python packages.

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:

```
'''         Elly de la cruz data            '''
elly = pd.read_csv("./ellydelacruz.csv")
elly = elly.drop(columns=['Runner', 'Fielder', 'pitcher_stealing_runs', 'Ball', 'Strike', 'Link'])
elly = elly.sort_values(by = "at_pitch_release")
elly = elly.sort_values(by = "SB/CS")
elly["Jump"] = np.nan
elly['Jump'] = elly['Jump'].astype(float)
```

2. Data Manipulation

To proceed with our project, we needed to create some new statistics to make sure that we had all the information we needed. This involves manipulation of our data, which are explained below.

Creating new Stats

To obtain the “Jump” data, we had to subtract the “at_pitch_release” data from our “lead_distance_gained” data for each row. We also added a new column into our data frame called “Jump”, displayed in the Imports section.

```
'''         Fill Jump           '''
for i, row in elly.iterrows():
    elly.loc[i, 'Jump'] = elly.loc[i, 'at_pitch_release'] - elly.loc[i, 'lead_distance_gained']
```

Creating new dataframes

We then split up our complete data into three different dataframes. These three data frames included: A dataframe where Elly De La Cruz was successful in stealing a base, one where he was caught stealing, and one where there was a pitchout.

```
'''         Print table with pitchout occurences when Elly steals        '''
elly_po = elly.copy()
elly_po.set_index('Notes')
for i, row in elly_po.iterrows():
    temp = row['Notes']
    base = row['SB/CS']
    if temp != 'Pitchout' or (temp == 'Pitchout' and base == 'SB'):
        elly_po.drop(i, axis = 0, inplace = True)
```

```
'''         Print table with only Elly's Stolen Base occurances        '''
elly_sb = elly.copy()
elly_sb.set_index('SB/CS')
for i, row in elly_sb.iterrows():
    temp = row['SB/CS']
    if temp != 'SB' :
        elly_sb.drop(i, axis = 0, inplace = True)
```

```
'''          Print table with only Elly's caught stealing occurances          '''
elly_cs = elly.copy()
elly_cs.set_index('SB/CS')
for i, row in elly_cs.iterrows():
    temp = row['SB/CS']
    if temp == 'SB':
        elly_cs.drop(i, axis = 0, inplace = True)
```

3. Results and Plots

The goal of this project was to find a way to prevent Elly De La Cruz from stealing second base. Below is the code I used to generate some plots that helped me get to my conclusion. I used the newly created data frames from section 2 to create these plots for my conclusion.

This block of code creates a plot that gets the relationship between the velocity of the pitch, the lead distance from 1B at pitch release, and the stolen base outcome.

```
'''         Velocity of ball VS Distance from 1B at Pitch Release         '''
ax = elly_sb.plot(kind = "scatter", x='Velocity', y='at_pitch_release', label='Stolen Base', color='green', title = "Velocity of Pitch vs Distance from 1B at Pitch Release")
elly_cs.plot(kind = "scatter", x='Velocity', y='at_pitch_release', label='Caught Stealing', color='red', ax=ax)
elly_po.plot(kind = "scatter", x='Velocity', y='at_pitch_release', label='Pitchout', color='blue', ax=ax)
ax.set_xlabel("Velocity of Pitch (mph)")
ax.set_ylabel("Distance from 1B at Pitch Release (ft)")
plt.show()
```

This block of code creates a plot that gets the relationship between the Handedness of the pitcher, the lead distance from 1B, and the stolen base outcome.

```
'''         Handedness vs Lead Distance          '''
ax = elly_sb.plot(kind = "scatter", x='Handedness', y='lead_distance_gained', label='Stolen Base', color='green', title = "Handedness vs Lead Distance")
elly_cs.plot(kind = "scatter", x='Handedness', y='lead_distance_gained', label='Caught Stealing', color='red', ax=ax)
elly_po.plot(kind = "scatter", x='Handedness', y='lead_distance_gained', label='Pitchout', color='blue', ax=ax)
ax.set_xlabel("Handedness")
ax.set_ylabel("Lead Distance from 1B (ft)")
plt.show()
```

This block of code creates a plot that gets the relationship between the Handedness of the pitcher, the distance from 1B at pitch release, and the stolen base outcome.

```
'''         Handedness vs Distance from 1B at pitch release          '''
ax = elly_sb.plot(kind = "scatter", x='Handedness', y='at_pitch_release', label='Stolen Base', color='green', title = "Handedness vs Distance from 1B at Pitch Release")
elly_cs.plot(kind = "scatter", x='Handedness', y='at_pitch_release', label='Caught Stealing', color='red', ax=ax)
elly_po.plot(kind = "scatter", x='Handedness', y='at_pitch_release', label='Pitchout', color='blue', ax=ax)
ax.set_xlabel("Handedness")
ax.set_ylabel("Distance from 1B at Pitch Release (ft)")
plt.show()
```

This block of code creates a plot that gets the relationship between the jump that Elly De La Cruz gets and the stolen base outcome.

```
'''         Jump vs Stolen Base Outcome         '''
ax = elly_sb.plot(kind = "scatter", x='Jump', y='SB/CS', label='Stolen Base', color='green', title = "Jump vs Stolen Base Outcome")
elly_cs.plot(kind = "scatter", x='Jump', y='SB/CS', label='Caught Stealing', color='red', ax=ax)
elly_po.plot(kind = "scatter", x='Jump', y='SB/CS', label='Pitchout', color='blue', ax=ax)
ax.set_ylabel("Outcome")
plt.gca().invert_yaxis()
ax.set_xlabel("Jump (ft)")
plt.show()
```

5. Acknowledgements

I would like to thank Baseball Savant for providing the data used for this project. I would also like to thank Richard Diamond, who gave me guidance throughout this project, and the internet resources that I used that include StackOverflow, ChatGPT, GeeksForGeeks and W3 schools.