

# Notebook Contents

- Imports
- Data
- Pitcher Overview
  - General Pitch Data
  - Pitch Usage
  - Pitch Usage by Batter Handedness
  - Velocity by Pitch Type
  - Pitch Velocity by Pitch Number
- Pitcher Stuff
  - Spin Rate by Pitch Type
  - Spin Axis
  - Spin Efficiency
  - HB & VB Axis
  - HB & HB Due to Magnus Force
  - Release Position
  - Release Extension
  - Velocity & Spin Rate
  - Avg. Bauer Units by Pitch Type
- Count Breakdown
  - Pitch Usage by Count
- Heatmaps
  - Pitch Location by Pitch Type
  - Hard Hit Summary by Pitch Type
  - 4-Seam Heatmaps
  - Knuckle Curve Heatmaps

## Imports

```
In [1]: import math
import matplotlib.patches as mpatches
from matplotlib.patches import Rectangle
import matplotlib.ticker
import PercentFormatter
from warnings import filterwarnings
filterwarnings('ignore')
```

## Data

```
In [2]: kimbrel = pd.read_csv('../data/craig-kimbrel.csv')
kimbrel.drop(columns=['Unnamed: 0'], inplace = True)
kimbrel.dropna(subset=['pitch_type'], inplace = True)

# Font Dictionary
font_title = {
    'size': 14,
    'weight': 'bold',
    'verticalalignment': 'center_baseline',
    'horizontalalignment': 'center'
}

pd.set_option('max_columns', None)
print(kimbrel.shape)
kimbrel.head(2)
```

```
Out [2]:
```

	pitch_type	game_date	release_speed	release_pos_x	release_pos_z	player_name	batter	pitcher	events	description	zon
0	FF	2021-08-31	95.6	-2.78	4.88	Kimbrel, Craig	607732	518886	out	hit_into_play	3.
1	KC	2021-08-31	86.2	-2.98	5.00	Kimbrel, Craig	592567	518886	strikeout	swinging_strike	14.

```
In [3]: kimbrel.groupby('pitch_type').mean()
```

```
Out [3]:
```

	release_speed	release_pos_x	release_pos_z	batter	pitcher	zone	hit_location	balls	strikes	gam
FF	96.617436	-2.949983	4.779795	596628.485470	518986.0	8.163846	4.572650	10.888889	0.974987	
KC	86.076559	-2.891172	4.919277	604564.184539	518986.0	11.336658	3.924051	0.675810	0.962594	

```
In [4]: gen_data = kimbrel[['pitch_type', 'release_speed', 'release_spin_rate',
                          'true_spin', 'spin_eff', 'phi', 'pfx_x', 'pfx_z',
                          'is_strike', 'release_pos_x', 'release_pos_z', 'bauer_units']]
col_dict = {
    'release_speed': 'velo', 'release_spin_rate': 'spin', 'phi': 'spin_axis', 'pfx_x': 'hb', 'pfx_z': 'vb',
    'is_strike': 'strike', 'release_pos_x': 'r_height', 'release_pos_z': 'r_side'
}
gen_data.rename(columns = col_dict, inplace = True)
hit_labels = [1, 2, 3, 4, 5]
kimbrel['hard_hit_summary'] = pd.qcut(kimbrel['launch_speed'], [0, .5262, .617, .7283, .8278, 1],
                                     labels = hit_labels)
```

## Pitcher Overview

### General Pitch Data

```
In [6]: gen_data.groupby(['pitch_type'], sort = False).mean()
```

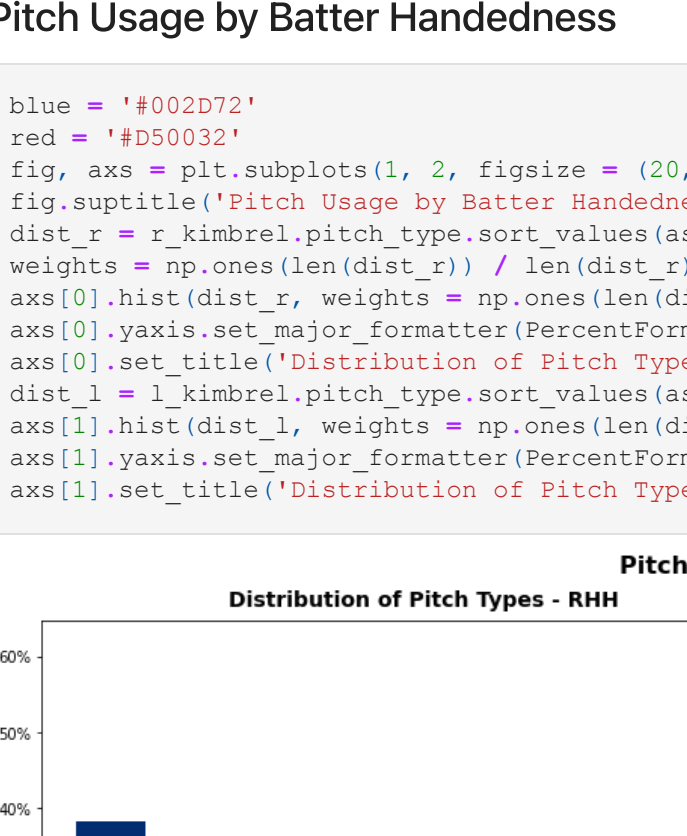
```
Out [6]:
```

	velo	spin	true_spin	spin_off	spin_axis	hb	vb	strike	r_side	r_height	bauer
FF	96.617436	2301.417094	1527.904587	0.670275	217.770642	14.643487	10.275077	0.644444	-2.949983	4.779795	23.8
KC	86.076559	2257.793017	952.416745	0.389894	412.457447	-5.713616	-10.410973	0.613466	-2.891172	4.919277	29.1

### Pitch Usage

```
In [7]: plt.figure(figsize = (8, 6))
dist = round(kimbrel.pitch_type.value_counts(normalize = True), 2)
color = sns.color_palette('coolwarm', 2)
plt.pie(dist, labels = order, colors = color, autopct = '%.0f%%')
plt.title('Distribution of Pitch Types - Craig Kimbrel', fontdict = font_title, pad = 15);
```

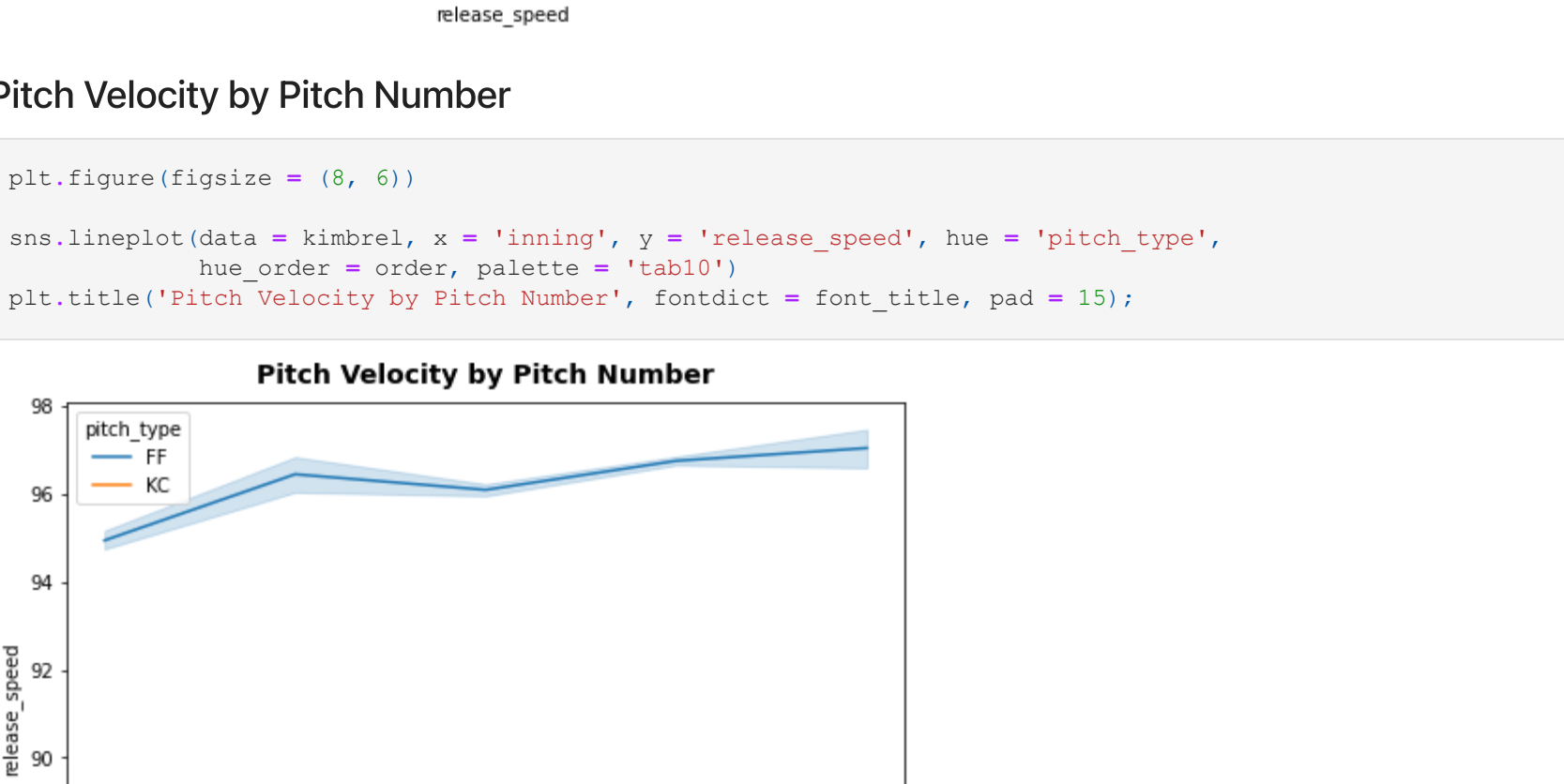
**Distribution of Pitch Types - Craig Kimbrel**



### Pitch Usage by Batter Handedness

```
In [8]: blue = '#002D72'
red = '#D50032'
fig, axs = plt.subplots(1, 2, figsize = (20, 6))
fig.suptitle('Pitch Usage by Batter Handedness', fontsize = 16, fontweight = 'bold')
dist_r = r_kimbrel.pitch_type.sort_values(ascending = False)
weights = np.ones(len(dist_r)) / len(dist_r)
axs[0].hist(dist_r, weights = weights, ones(len(dist_r)) / len(dist_r), color = blue)
axs[0].set_title('Distribution of Pitch Types - RHH', fontdict = font_title, pad = 15)
dist_l = l_kimbrel.pitch_type.sort_values(ascending = False)
axs[1].hist(dist_l, weights = np.ones(len(dist_l)) / len(dist_l), color = red)
axs[1].set_title('Distribution of Pitch Types - LHH', fontdict = font_title, pad = 15);
```

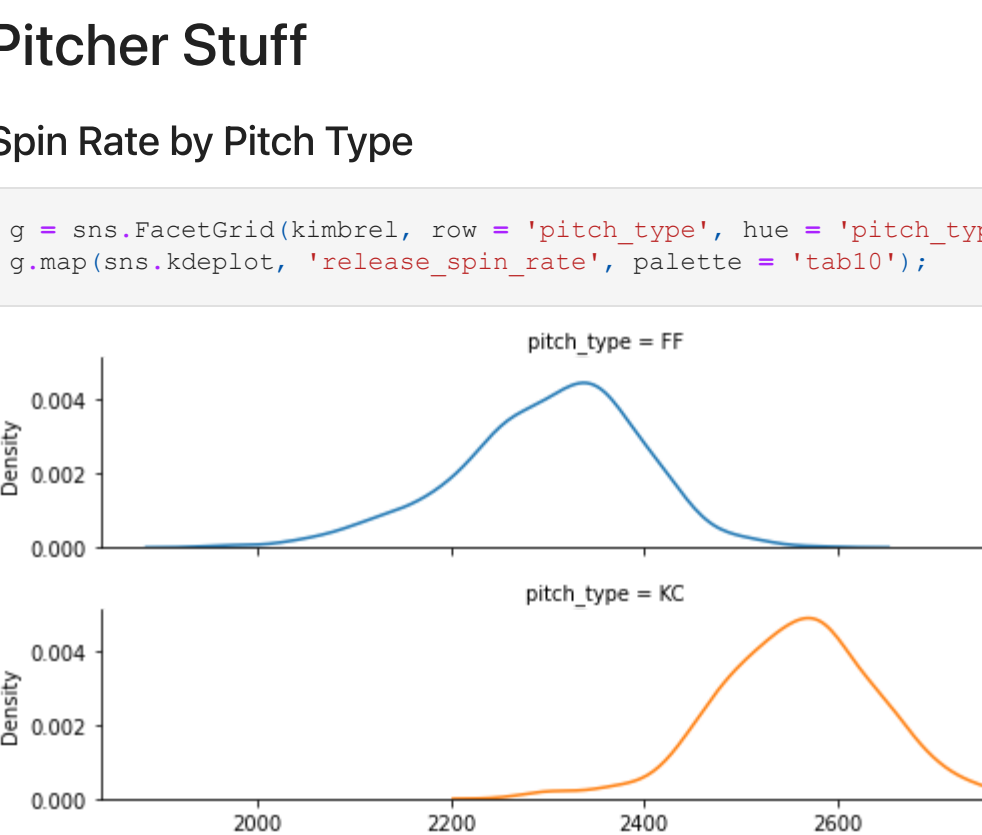
**Pitch Usage by Batter Handedness**



### Velocity by Pitch Type

```
In [9]: plt.figure(figsize = (8, 6))
sns.kdeplot(data = kimbrel, x = 'release_speed', shade = 'fill', hue = 'pitch_type',
            axline = True, legend = True, axlabel = 'release_speed', fontdict = font_title, pad = 12);
```

**Distribution of Velocity by Pitch Type - Craig Kimbrel**



### Pitch Velocity by Pitch Number

```
In [10]: plt.figure(figsize = (8, 6))
lineplot(data = kimbrel, x = 'inning', y = 'release_speed', hue = 'pitch_type',
         axline = True, legend = True, axlabel = 'release_speed', fontdict = font_title, pad = 15);
```

**Pitch Velocity by Pitch Number**

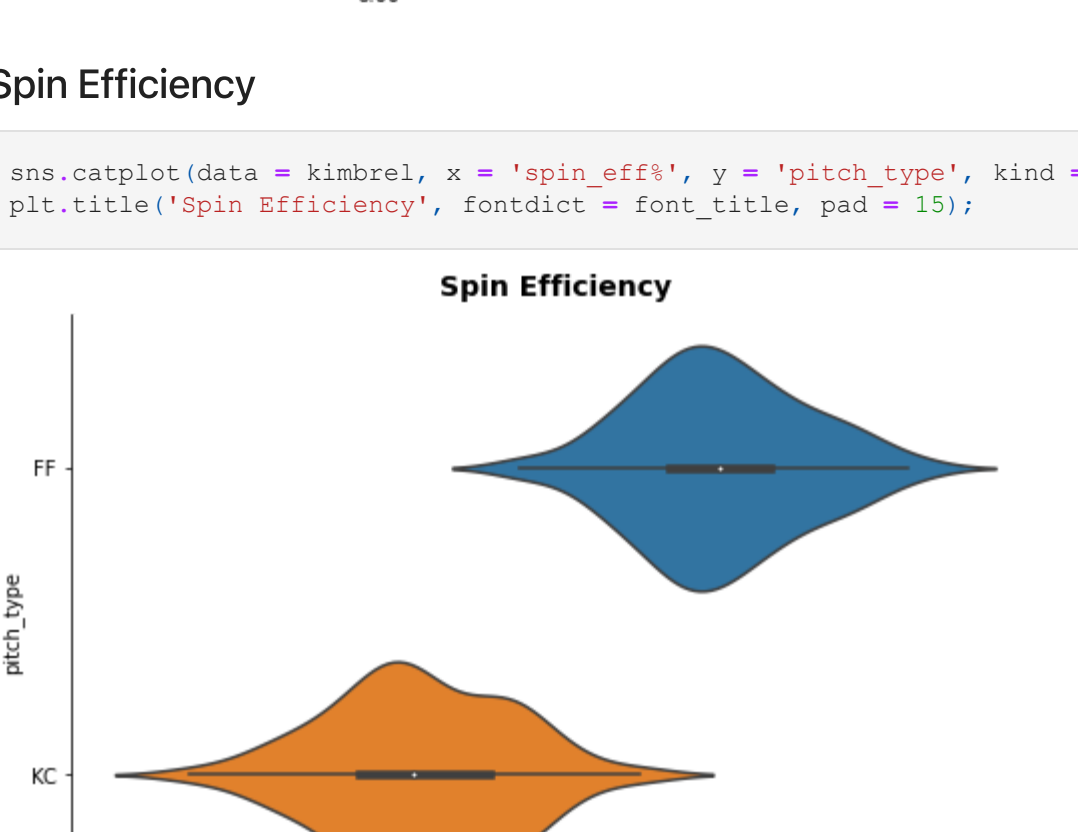


## Pitcher Stuff

### Spin Rate by Pitch Type

```
In [11]: g = sns.FacetGrid(kimbrel, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4)
g.map(sns.kdeplot, 'release_spin_rate', fill = True, color = 'black', linewidth = 2)
```

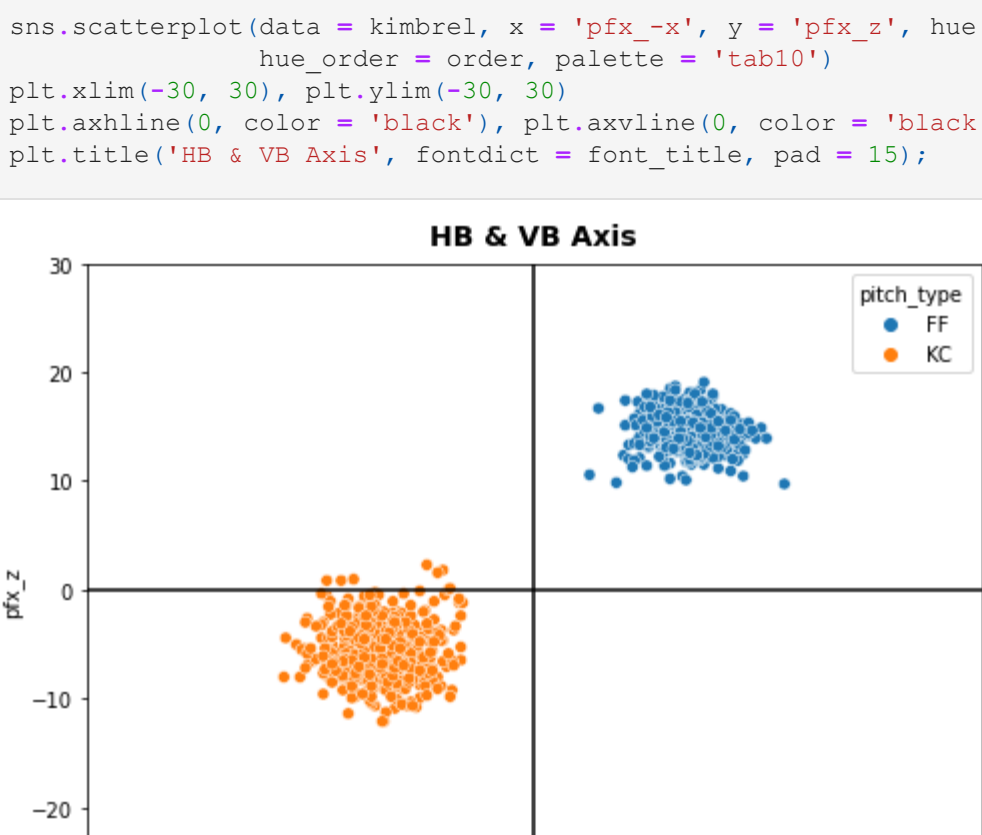
**Spin Rate by Pitch Type**



### Spin Axis

```
In [12]: ax = plt.figure(figsize = (8, 6))
ax = plt.subplot(polar = True, theta_direction = 'out')
ax.set(math.radians(font_title['size']), 1, color = 'blue', marker = 'o', label = '4-Seam')
ax.plot(math.radians(font_title['size']), 1, color = 'orange', marker = 'o', label = 'Knuckle Curve')
ax.set_title('Spin Axis', fontdict = font_title, pad = 15);
```

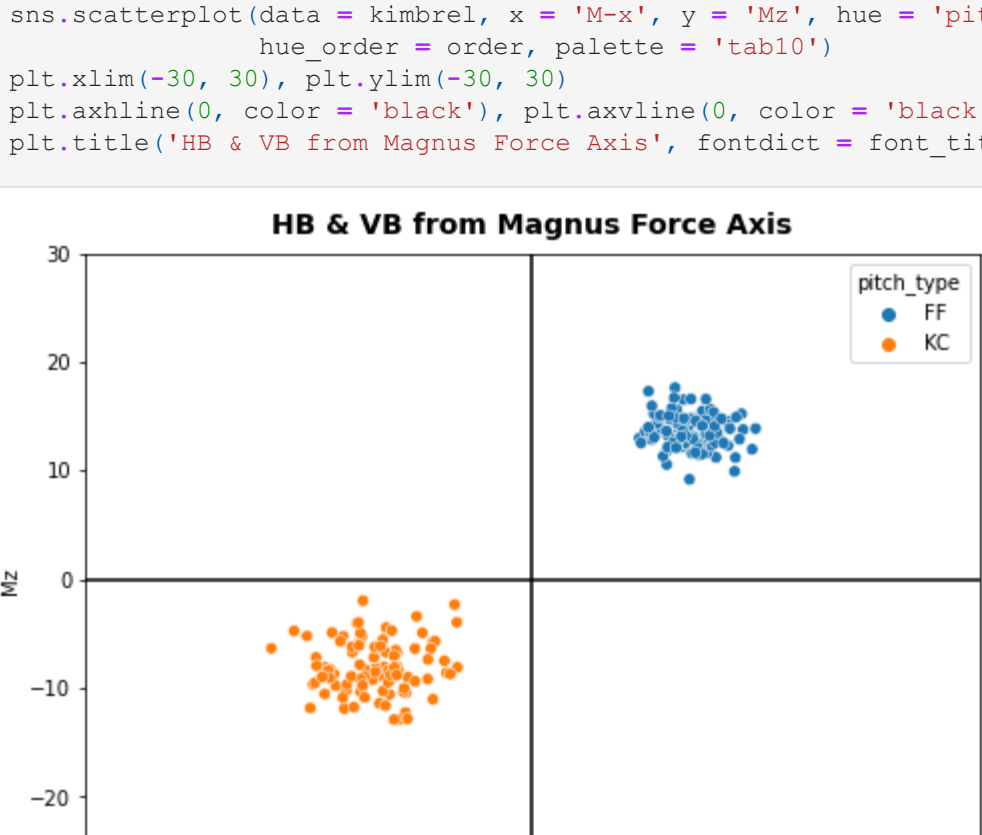
**Spin Axis**



### Spin Efficiency

```
In [13]: sns.kdeplot(data = kimbrel, x = 'spin_eff', y = 'pitch_type', kind = 'violin', aspect = 1.5, palette = 'tab10')
plt.title('Spin Efficiency', fontdict = font_title, pad = 15);
```

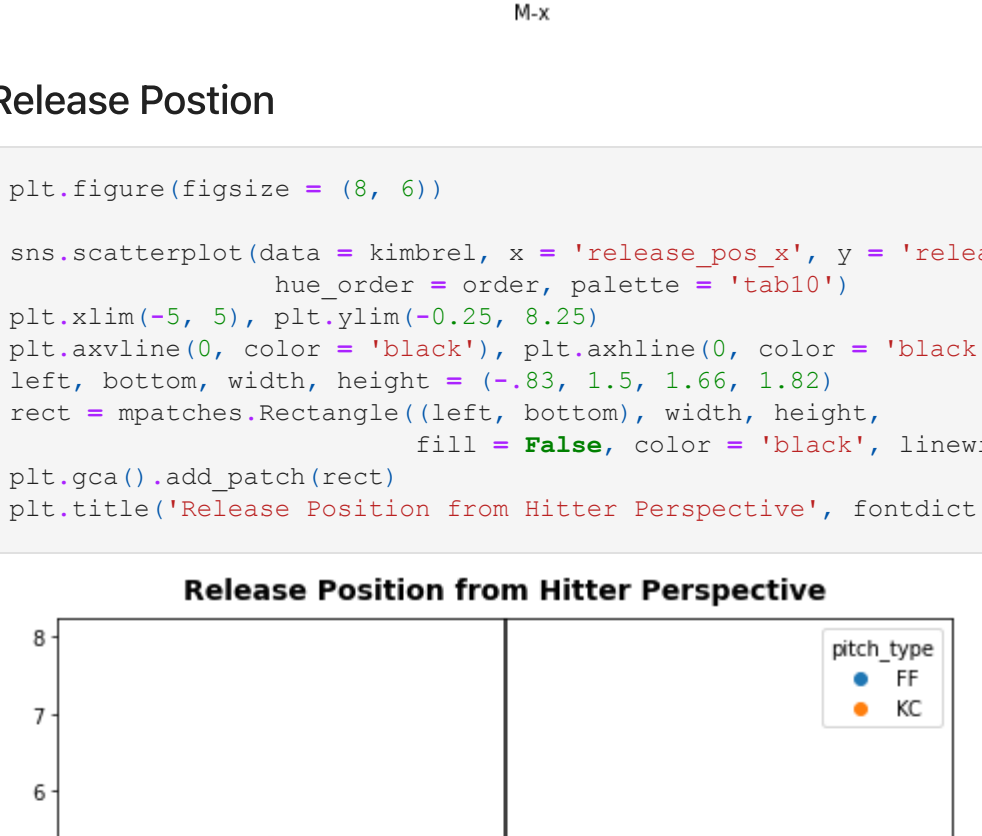
**Spin Efficiency**



### Horizontal & Vertical Break Axis

```
In [14]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'pfx_x', y = 'pfx_z', hue = 'pitch_type',
               axline = True, legend = True, axlabel = 'pfx_x', fontdict = font_title, pad = 15);
```

**HB & VB Axis**



### Horizontal & Vertical Break Due to Magnus Force Axis

```
In [15]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'Mx', y = 'Mz', hue = 'pitch_type',
               axline = True, legend = True, axlabel = 'Mx', fontdict = font_title, pad = 15);
```

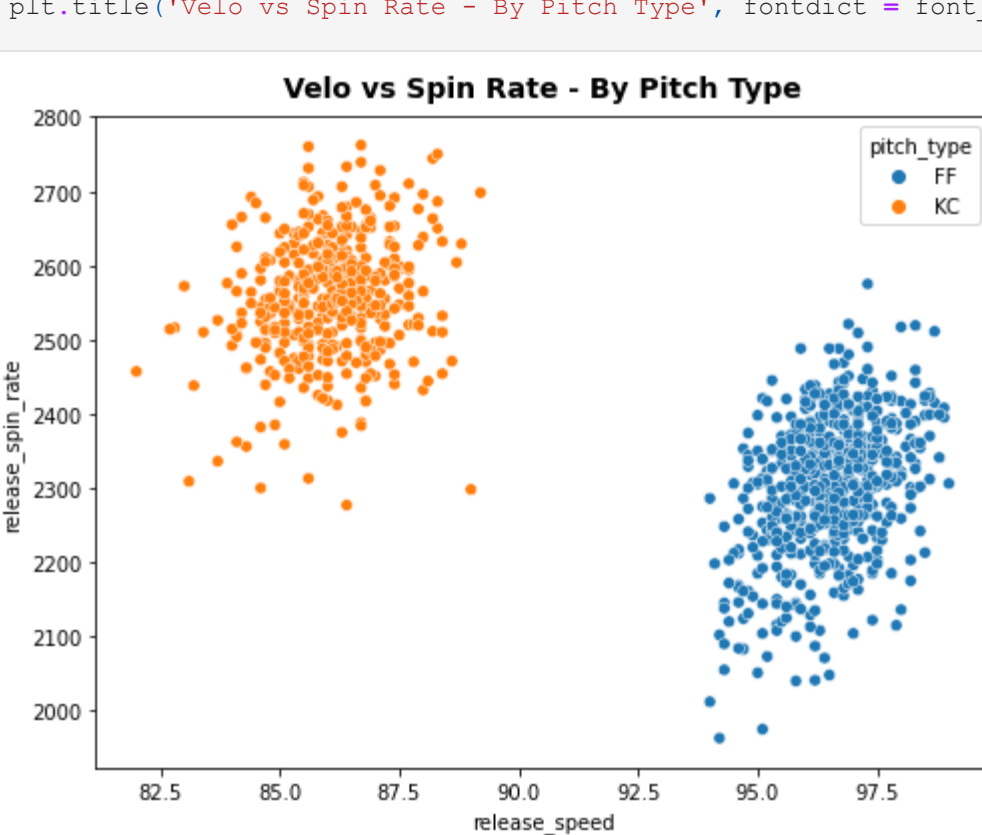
**HB & VB from Magnus Force Axis**



### Release Postion

```
In [16]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'release_pos_x', y = 'release_pos_z', hue = 'pitch_type',
               axline = True, legend = True, axlabel = 'release_pos_x', fontdict = font_title, pad = 15);
```

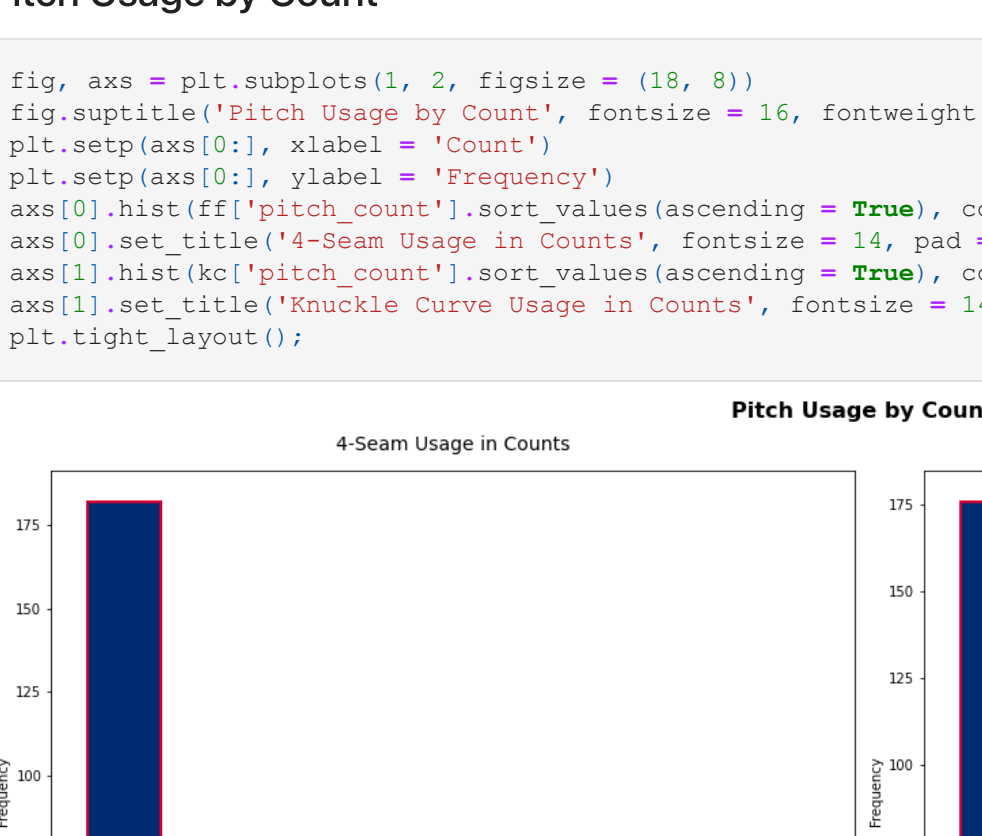
**Release Position from Hitter Perspective**



### Release Extension (Side View)

```
In [17]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'release_extension', y = 'release_pos_z', hue = 'pitch_type',
               axline = True, legend = True, axlabel = 'release_extension', fontdict = font_title, pad = 15);
```

**Release Extension (Side View)**



### Velocity & Spin Rate

```
In [18]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'release_speed', y = 'release_spin_rate', hue = 'pitch_type',
               axline = True, legend = True, axlabel = 'release_speed', fontdict = font_title, pad = 15);
```

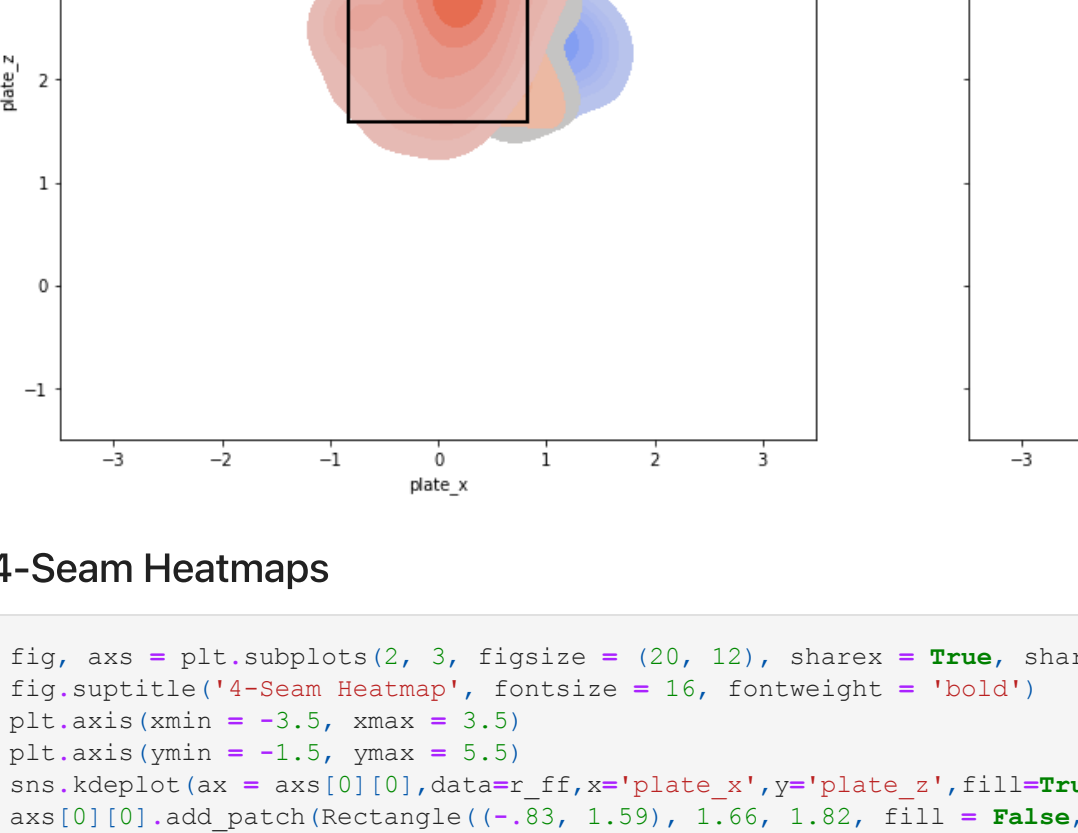
**Velo vs Spin Rate - By Pitch Type**



### Bauer Units

```
In [19]: g = sns.FacetGrid(kimbrel, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4, )
g.map(sns.kdeplot, 'bauer_units', palette = 'tab10')
```

**Bauer Units**



## Count Breakdown

### Pitch Usage by Count

```
In [20]: fig, axs = plt.subplots(1, 2, figsize = (18, 8))
fig.suptitle('Pitch Usage by Count', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0], xlabel = 'Count')
plt.setp(axs[1], xlabel = 'Count')
axs[0].hist(ff['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
axs[1].hist(kc['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
axs[0].set_title('4-Seam Usage in Counts', fontsize = 14, pad = 15)
axs[1].set_title('Knuckle Curve Usage in Counts', fontsize = 14, pad = 15)
plt.tight_layout();
```

**Pitch Usage by Count**



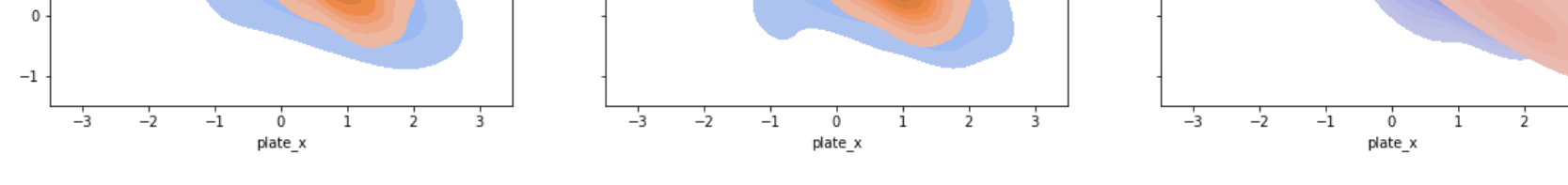
## Heatmaps

### \*All From Hitters' Perspective

### Pitch Location by Pitch Type

```
In [21]: fig, axs = plt.subplots(1, 2, figsize = (18, 8), sharex = True, sharey = True)
fig.suptitle('Pitch Location by Pitch Type', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0], xlabel = 'plate_x', ylabel = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
axs[1].set_title('4-Seam Location vs All Hitters', fontsize = 14, pad = 15)
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = True, hue = 'is_strike', palette = 'coolwarm'))
axs[1].set_title('Knuckle Curve Location vs All Hitters', fontsize = 14, pad = 15);
```

**Pitch Location by Pitch Type**



### Hard Hit Summary by Pitch Type

```
In [22]: fig, axs = plt.subplots(1, 2, figsize = (18, 8), sharex = True, sharey = True)
fig.suptitle('Hard Hit Summary by Pitch Type', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0], xlabel = 'plate_x', ylabel = 'plate_z', fill = True, hue = 'hard_hit_summary', palette = 'coolwarm')
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
axs[1].set_title('4-Seam Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15)
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = True, hue = 'hard_hit_summary', palette = 'coolwarm'))
axs[1].set_title('Knuckle Curve Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15);
```

**Hard Hit Summary by Pitch Type**



### 4-Seam Heatmaps

```
In [23]: fig, axs = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('4-Seam Heatmap', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0], xlabel = 'plate_x', ylabel = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
axs[1].set_title('Location vs RHH', fontsize = 14, pad = 15)
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = True, hue = 'swing_miss', palette = 'coolwarm'))
axs[1].set_title('Swing and Miss vs RHH', fontsize = 14, pad = 15)
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = True, hue = 'BABIP_value', palette = 'coolwarm'))
axs[1].set_title('BABIP Value vs RHH', fontsize = 14, pad = 15);
```

**4-Seam Heatmap**



### Knuckle Curve Heatmaps

```
In [24]: fig, axs = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('Knuckle Curve Heatmap', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0], xlabel = 'plate_x', ylabel = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
axs[1].set_title('Location vs RHH', fontsize = 14, pad = 15)
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = True, hue = 'swing_miss', palette = 'coolwarm'))
axs[1].set_title('Swing and Miss vs RHH', fontsize = 14, pad = 15)
axs[0].add_patch(Rectangle((-0.8, 1.59), 1.66, 1.62, fill = True, hue = 'woba_value', palette = 'coolwarm'))
axs[1].set_title('woba Value vs RHH', fontsize = 14, pad = 15);
```

**Knuckle Curve Heatmap**





