

Craig Kimbrel

Notebook Contents

- Imports
 - Data
 - Index DataFrame to Get Pitch Types
- Pitcher Overview
 - General Pitch Data
 - Pitch Usage
 - Pitch Usage by Batter Handedness
 - Velocity by Pitch Type
 - Pitch Velocity by Inning
 - Pitcher Stuff
 - Spin Rate by Pitch Type
 - Spin Axis
 - HB & VB Axis
 - HB & VB Due to Magnus Force
 - Release Position
 - Release Extension
 - Velocity & Spin Rate
 - Avg. Bauer Units by Pitch Type
 - Pitch Usage by Count
 - Count Breakdown
 - Pitch Usage by Count
 - 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
plt = matplotlib.patches.mpatches.Rectangle
from matplotlib.ticker import PercentFormatter

import warnings
warnings.filterwarnings('ignore')
```

Data

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

# Font Dictionary
font_title = {
    "size": 14,
    "align": "center",
    "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	zone
0	FF	2021-08-31	95.6	-2.78	4.88	Kimbrel, Craig	607732	518886	out	hit_into_lap	3.

```
1
```

1	KC	2021-08-31	86.2	-2.98	5.00	Kimbrel, Craig	592567	518886	strikeout	swinging_strike	14.
---	----	------------	------	-------	------	----------------	--------	--------	-----------	-----------------	-----

```
Out[3]:
```

	release_speed	release_pos_x	release_pos_z	batter	pitcher	zone	hit_location	balls	strikes	gam
pitch_type	FF	96.57436	-2.949983	4.779795	596628.485470	518886.0	8.153846	4.572650	1.088889	0.979487
	KC	86.076569	-2.891172	4.919277	600456.184539	518886.0	11.336658	3.924051	0.675810	0.862594

```
In [4]: gen_data = kimbrel[['pitch_type', 'release_speed', 'release_spin_rate',
                        'true_spin', 'spin_eff', 'phi', 'pfx_z', 'pfx_x',
                        'is_strike', 'release_pos_x', 'release_pos_z', 'bauer_units']]
col_dict = {
    'release_speed': 'velo', 'release_spin_rate': 'spin', 'phi': 'spin_axis', 'pfx_z': 'hb', 'pfx_x': '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)
```

Index DataFrame to Get Pitch Types

```
In [5]: #kimbrel.pitch_type.value_counts(normalize = True)
r_kimbrel = kimbrel.loc[kimbrel['stand'] == 'R']
l_kimbrel = kimbrel.loc[kimbrel['stand'] == 'L']
axis[1][0].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
ff = kimbrel.loc[kimbrel['pitch_type'] == 'FF']
kc = kimbrel.loc[kimbrel['pitch_type'] == 'KC']
r_ff = r_kimbrel.loc[r_kimbrel['pitch_type'] == 'FF']
r_kc = r_kimbrel.loc[r_kimbrel['pitch_type'] == 'KC']
l_ff = l_kimbrel.loc[l_kimbrel['pitch_type'] == 'FF']
l_kc = l_kimbrel.loc[l_kimbrel['pitch_type'] == 'KC']
order = ['FF', 'KC']

ff_tilt = ff['phi'].mean()
kc_tilt = kc['phi'].mean()
```

Pitcher Overview

General Pitch Data

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

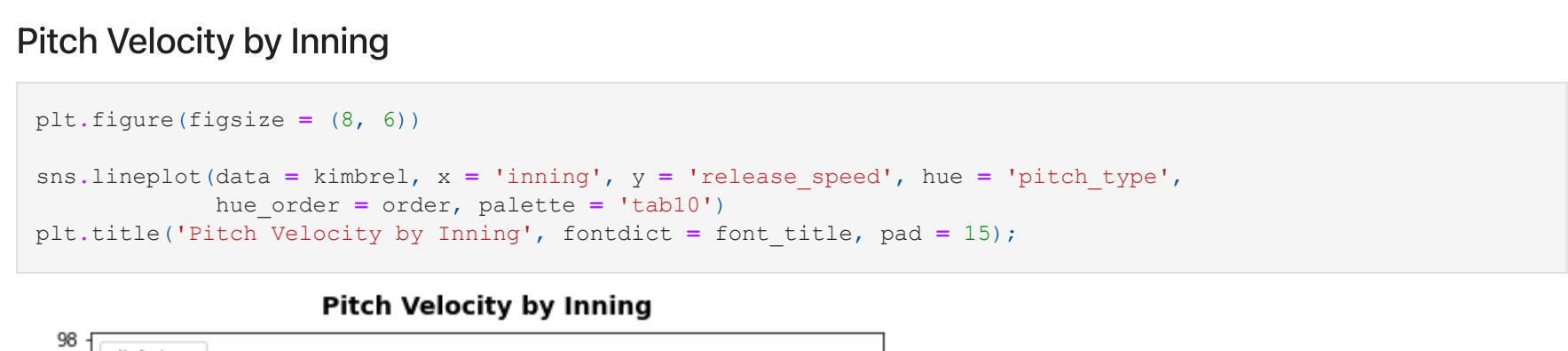
```
Out[6]:
```

	velo	spin	true_spin	spin_eff	spin_axis	hb	vb	strike	r_side	r_height	bauer
FF	96.57436	2301.47094	1627.904587	0.670275	217.770642	14.643487	10.275077	0.644444	-2.949983	4.779795	23.6
KC	86.076569	2565.739307	952.415745	0.389894	412.457447	-5.713616	-10.410973	0.613466	-2.891172	4.919277	29.7

Pitch Usage

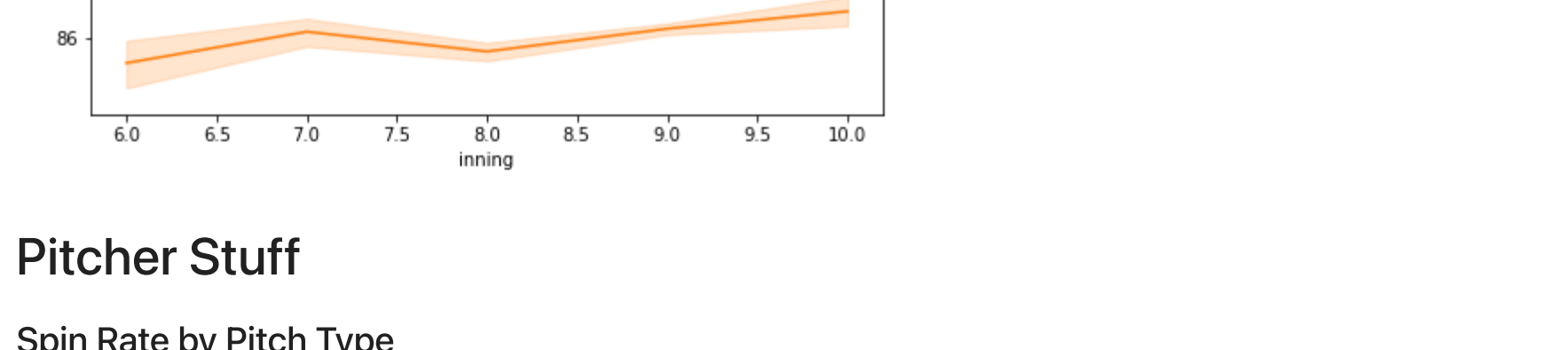
```
In [7]: plt.figure(figsize = (8, 6))
dist = round(kimbrel.pitch_type.value_counts(normalize = True), 2)
color = sns.color_palette('coolwarm_r')
plt.pie(dist, labels = order, colors = color, autopct = '%.0f%%')
title = f'Distribution of Pitch Types - Craig Kimbrel', fontdict = font_title, pad = 15);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

Distribution of Pitch Types - Craig Kimbrel



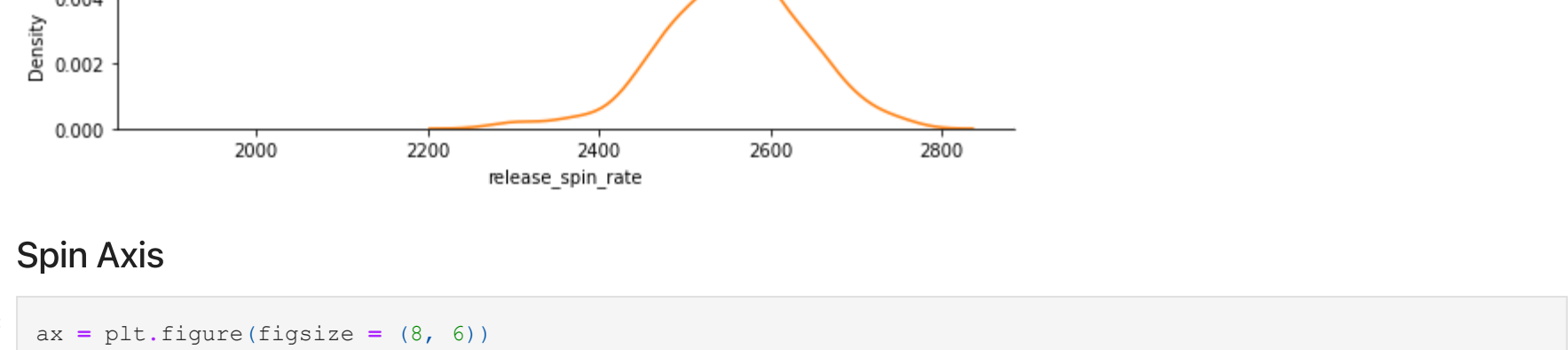
Pitch Usage by Batter Handedness

```
In [8]: blue = '#002D72'
red = '#D50027'
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)
axis[1][0].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
dist_l = l_kimbrel.pitch_type.sort_values(ascending = False)
axis[1].hist(dist_l, weights = np.ones(len(dist_l)) / len(dist_l), color = blue)
axis[1].yaxis.set_major_formatter(PercentFormatter(1))
axis[1].set_title('Distribution of Pitch Types - RHH', fontdict = font_title, pad = 15)
axis[1].set_title('Distribution of Pitch Types - LHH', fontdict = font_title, pad = 15);
```



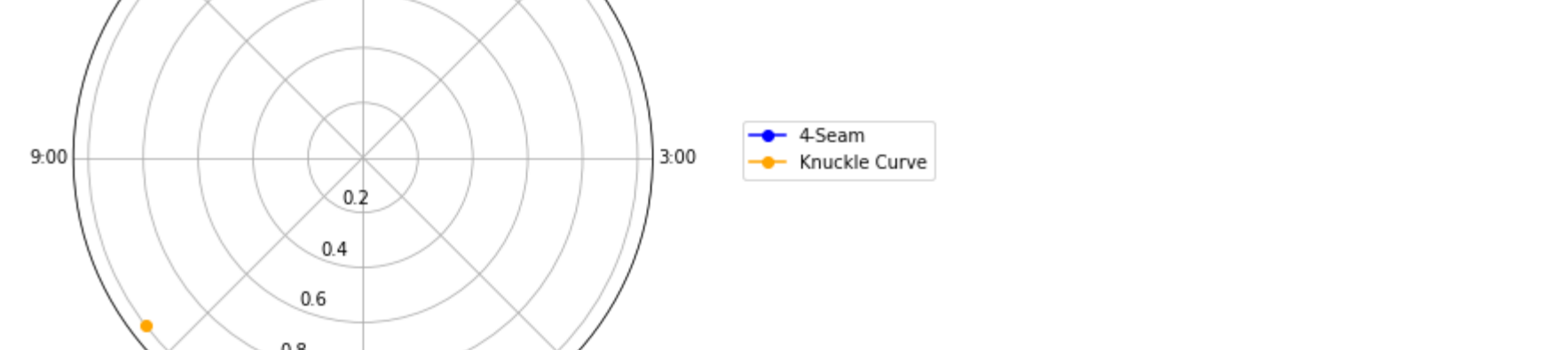
Velocity by Pitch Type

```
In [9]: plt.figure(figsize = (8, 6))
sns.kdeplot(data = kimbrel, x = 'release_speed', shade = 'fill', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.title('Distribution of Velocity by Pitch Type - Craig Kimbrel', fontdict = font_title, pad = 12);
```



Pitch Velocity by Inning

```
In [10]: plt.figure(figsize = (8, 6))
sns.lineplot(data = kimbrel, x = 'inning', y = 'release_speed', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.title('Pitch Velocity by Inning', fontdict = font_title, pad = 15);
```



Pitcher Stuff

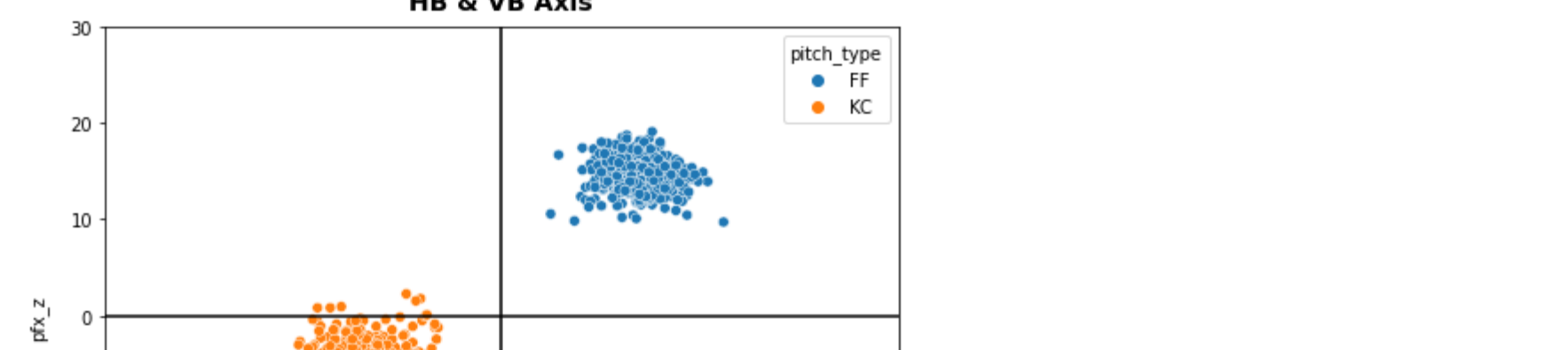
Spin Rate by Pitch Type

```
In [11]: g = sns.FacetGrid(kimbrel, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4)
g.map(sns.kdeplot, 'release_spin_rate', palette = 'tab10');
```



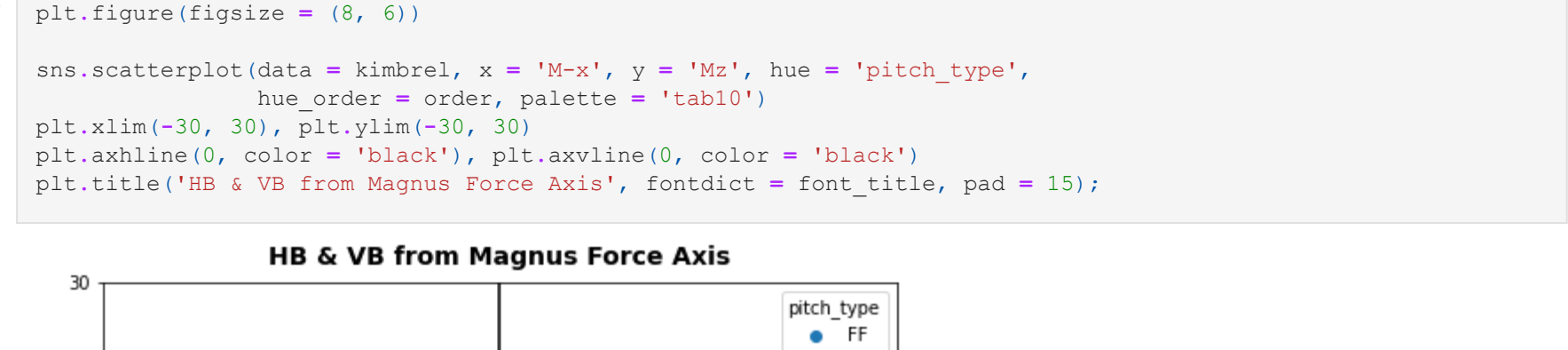
Spin Axis

```
In [12]: ax = plt.figure(figsize = (8, 6))
ax = plt.subplots(polar = True, data_direction = 'l')
ax.plot(math.radians(ff_tilt), 1, color = 'blue', marker = 'o', label = '4-Seam')
ax.plot(math.radians(kc_tilt), 1, color = 'orange', marker = 'o', label = 'Knuckle Curve')
axis[0].set_title('4-Seam', fontdict = font_title, pad = 15)
ax.set_xticklabels('450', '900', '1350', '1800', '2250', '2700', '3150', '3600')
ax.set_xticklabels('450', '900', '1350', '1800', '2250', '2700', '3150', '3600')
ax.set_theta_zero_location('S')
ax.set_title('Spin Axis', fontdict = font_title, pad = 15);
```



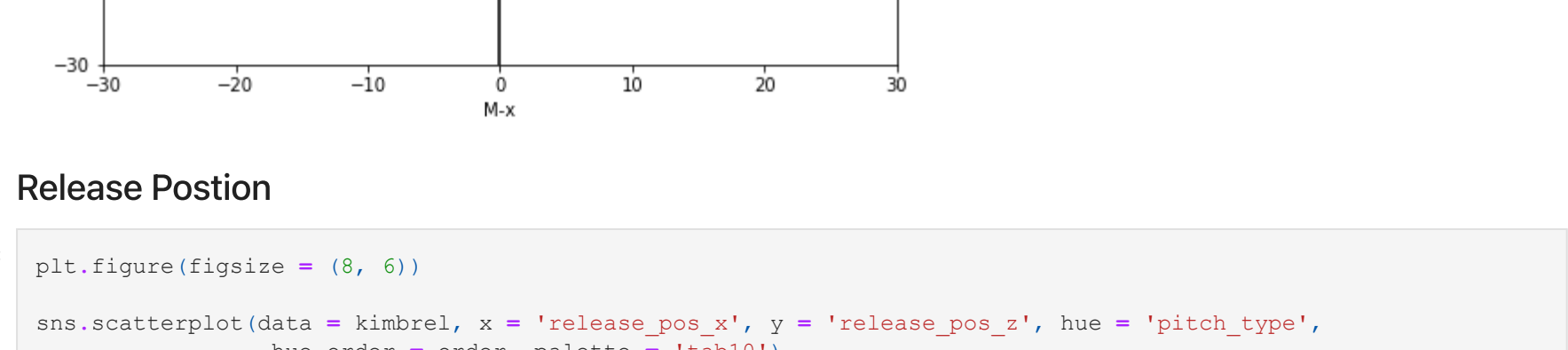
Spin Efficiency

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



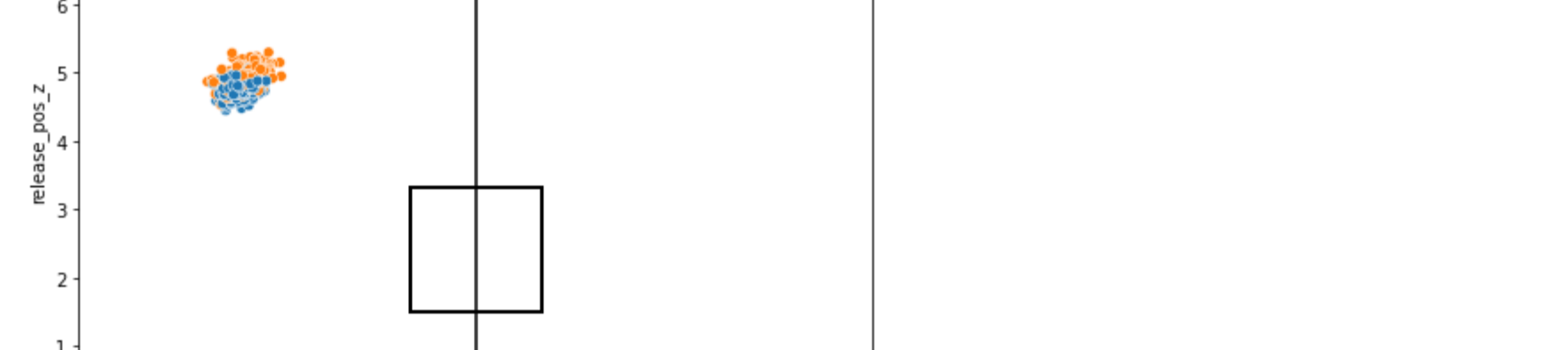
Horizontal & Vertical Break Axis

```
In [14]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'pfx_x', y = 'pfx_z', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.xlim(-30, 30), plt.ylim(-30, 30)
plt.axline(0, color = 'black'), plt.axline(0, color = 'black')
plt.title('HB & VB Axis', fontdict = font_title, pad = 15);
```



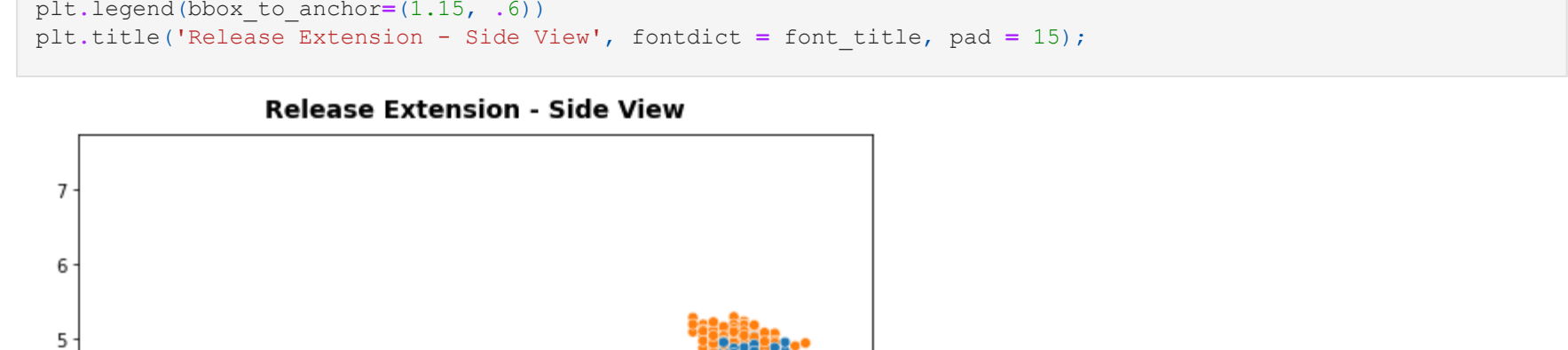
Horizontal & Vertical Break Due to Magnus Force Axis

```
In [15]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'Mx', y = 'Mz', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.xlim(-30, 30), plt.ylim(-30, 30)
plt.axline(0, color = 'black'), plt.axline(0, color = 'black')
plt.title('HB & VB from Magnus Force Axis', fontdict = font_title, pad = 15);
```



Release Postion

```
In [16]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'release_pos_x', y = 'release_pos_z', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.xlim(-5, 5), plt.ylim(-0.25, 0.25)
axis[0].set_title('Release Position from Hitter Perspective', fontdict = font_title, pad = 15);
```



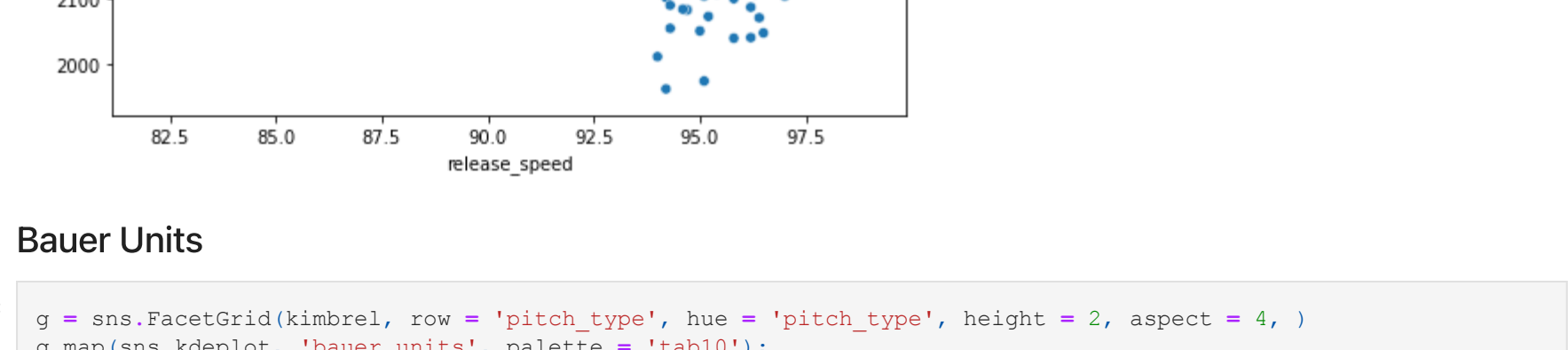
Release Extension - Side View

```
In [17]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'release_extension', y = 'release_pos_x', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.xlim(0, 7.75), plt.ylim(-1.5, 1.5)
plt.axline(0, .95), (7.75, 0), color = 'brown'), plt.axline(1, 0, .05, color = 'black')
plt.legend(bbox_to_anchor=(1.5, .6))
plt.title('Release Extension - Side View', fontdict = font_title, pad = 15);
```



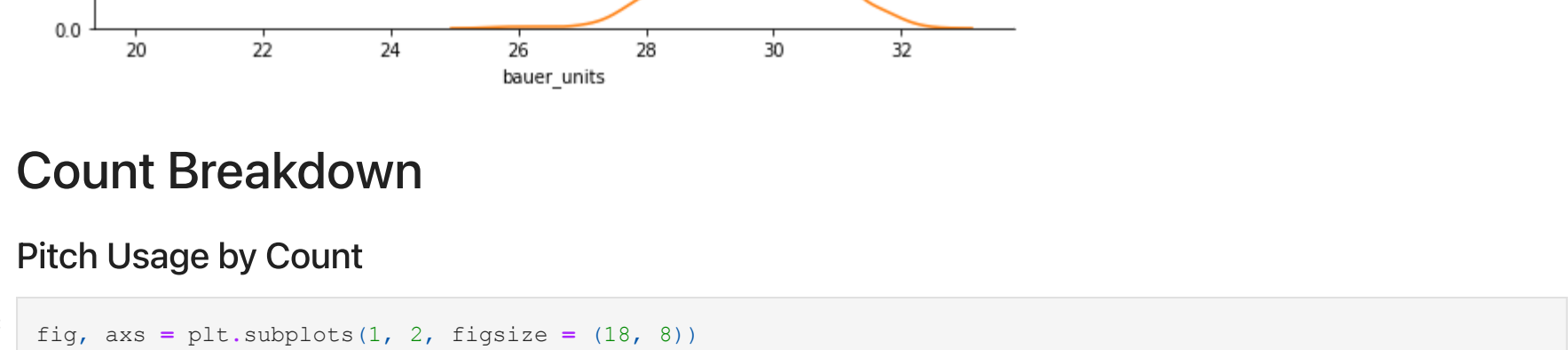
Velocity & Spin Rate

```
In [18]: plt.figure(figsize = (8, 6))
sns.scatterplot(data = kimbrel, x = 'release_speed', y = 'release_spin_rate', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.title('Velocity & Spin Rate - By Pitch Type', fontdict = font_title, pad = 15);
```



Bauer Units

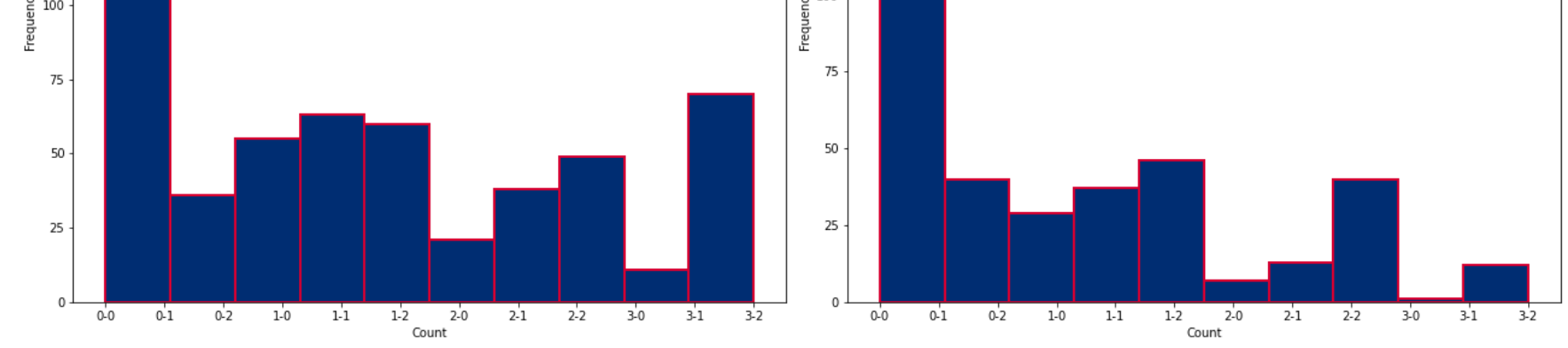
```
In [19]: g = sns.FacetGrid(kimbrel, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4, )
g.map(sns.kdeplot, 'bauer_units', palette = 'tab10');
```



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 = 'Frequency')
axis[0].hist(ff['pitch_count'], sort_values(ascending = True), color = blue, edgecolor = 'black', linewidth = 2)
axis[0].set_title('4-Seam Usage in Counts', fontdict = font_title, pad = 15)
axis[1].hist(kc['pitch_count'], sort_values(ascending = True), color = blue, edgecolor = 'black', linewidth = 2)
axis[1].set_title('Knuckle Curve Usage in Counts', fontdict = font_title, pad = 15)
plt.tight_layout();
```

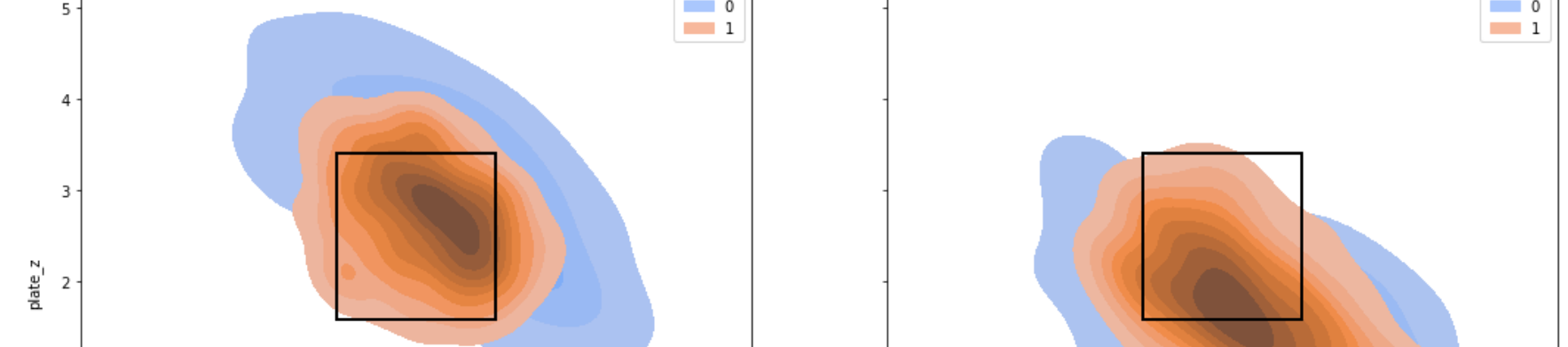


Heatmaps

*All From Hitters' Perspective

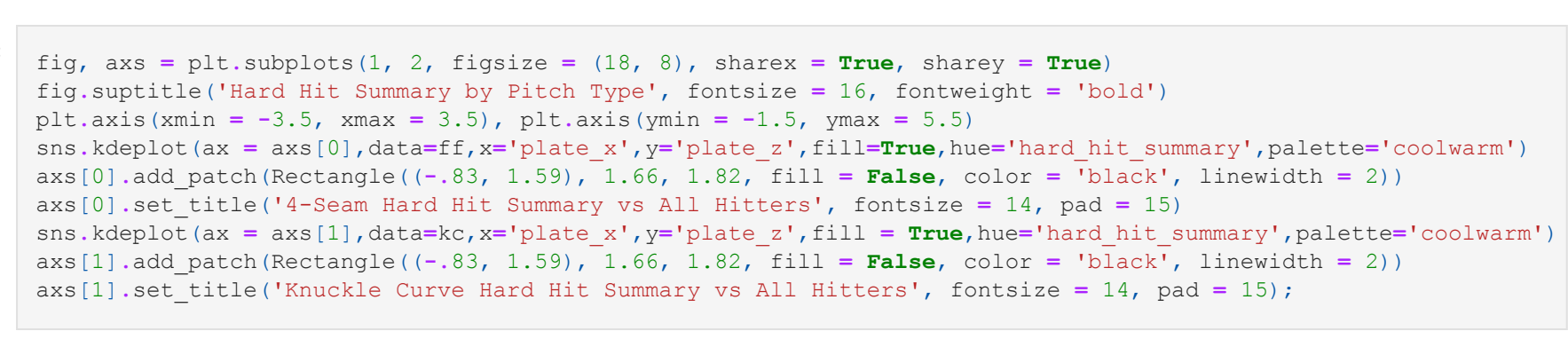
Pitch Location by Pitch Type

```
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 = 'pfx_x', ylabel = 'pfx_z')
axis[0].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[0].set_title('4-Seam Location vs All Hitters', fontsize = 14, pad = 15)
axis[1].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[1].set_title('Knuckle Curve Location vs All Hitters', fontsize = 14, pad = 15);
```



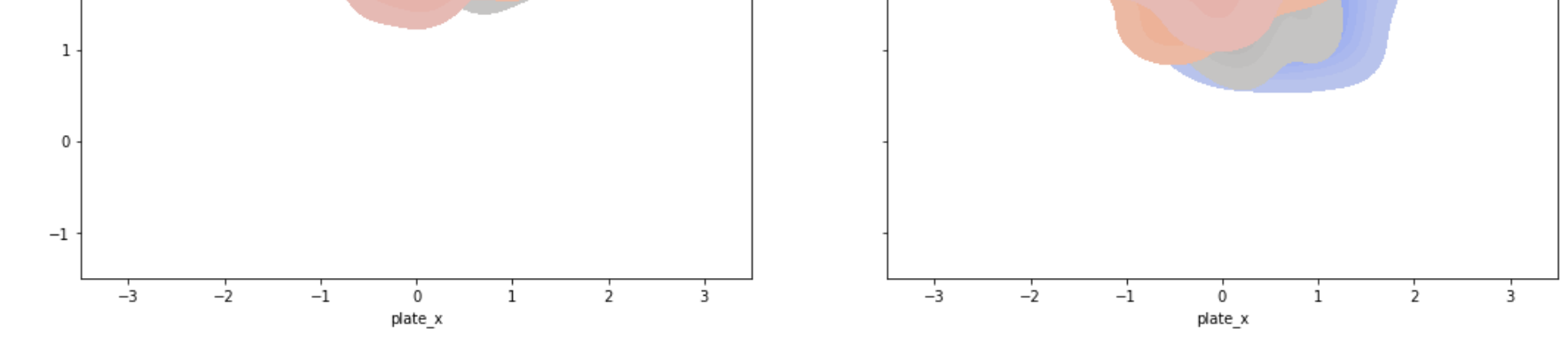
Hard Hit Summary by Pitch Type

```
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 = 'pfx_x', ylabel = 'pfx_z')
axis[0].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[0].set_title('4-Seam Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15)
axis[1].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[1].set_title('Knuckle Curve Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15);
```



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 = 'pfx_x', ylabel = 'pfx_z')
axis[0].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[0].set_title('Location vs RHH', fontdict = font_title, pad = 15)
axis[1].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[1].set_title('Swing and Miss vs RHH', fontdict = font_title, pad = 15)
axis[2].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[2].set_title('BABIP Value vs RHH', fontdict = font_title, pad = 15)
axis[3].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[3].set_title('Location vs LHH', fontdict = font_title, pad = 15)
axis[4].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[4].set_title('Swing and Miss vs LHH', fontdict = font_title, pad = 15)
axis[5].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[5].set_title('BABIP Value vs LHH', fontdict = font_title, pad = 15);
```



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 = 'pfx_x', ylabel = 'pfx_z')
axis[0].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[0].set_title('Location vs RHH', fontdict = font_title, pad = 15)
axis[1].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[1].set_title('Swing and Miss vs RHH', fontdict = font_title, pad = 15)
axis[2].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[2].set_title('BABIP Value vs RHH', fontdict = font_title, pad = 15)
axis[3].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[3].set_title('Location vs LHH', fontdict = font_title, pad = 15)
axis[4].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[4].set_title('Swing and Miss vs LHH', fontdict = font_title, pad = 15)
axis[5].add_patch(Rectangle((-83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axis[5].set_title('BABIP Value vs LHH', fontdict = font_title, pad = 15);
```

