

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
 - Avg. Bauer Units by Pitch Type
- Count Breakdown
 - Pitch Usage by Count
- Heatmaps
 - Pitch Location by Pitch Type
 - Quality Contact Heatmap - All Pitches
 - 4-Seam Heatmaps
 - Knuckle Curve Heatmaps

Imports

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

Data

```
In [2]: kimbrl = pd.read_csv("../data/craig-kimbrl.csv")
kimbrl.drop(columns = ['Unnamed: 0'], inplace = True)
kimbrl.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(kimbrl.shape)
kimbrl.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	Kimbrl, Craig	607732	518886	out	hit_into_play	3.
1	KC	2021-08-31	86.2	-2.98	5.00	Kimbrl, Craig	592567	518886	strikeout	swinging_strike	14.

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

```
Out [3]:
```

	release_speed	release_pos_x	release_pos_z	batter	pitcher	zone	hit_location	balls	strikes	gam
pitch_type										
FF	96.517436	-2.949983	4.779795	596628	495470	518886.0	8.163846	4.572650	1.088989	0.979487
KC	86.078559	-2.891172	4.919277	604564	184539	518886.0	-11.336658	3.924051	0.675810	0.962594

```
In [4]: gen_data = kimbrl[['pitch_type', 'release_speed', 'release_spin_rate',
                        'true_spin', 'spin_eff', 'phi', 'pfx_x', 'pfx_z', 'pfx_y',
                        'is_strike', 'release_pos_x', 'release_pos_z', 'batter_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)
```

```
In [5]: #kimbrl.pitch_type.value_counts(normalize = True)
r_kimbrl = kimbrl.loc[kimbrl['stand'] == 'R']
# all hitters
ff = kimbrl.loc[kimbrl['pitch_type'] == 'FF']
l_kimbrl = kimbrl.loc[kimbrl['pitch_type'] == 'L']
r_ff = r_kimbrl.loc[r_kimbrl['pitch_type'] == 'FF']
l_kimbrl = kimbrl.loc[l_kimbrl['pitch_type'] == 'L']
l_ff = l_kimbrl.loc[l_kimbrl['pitch_type'] == 'FF']
l_l = l_kimbrl.loc[l_kimbrl['pitch_type'] == 'L']
order = ['FF', 'KC']
```

```
In [6]: ff_tilt = ff['phi'].mean()
ff_tilt = kc['phi'].mean()
```

Pitcher Overview

General Pitch Data

```
In [7]: gen_data.groupby(['pitch_type'], sort = False).mean()
```

```
Out [7]:
```

	velo	spin	true_spin	spin_eff	spin_axis	hb	vb	strike	r_side	r_height	bauer
pitch_type											
FF	96.517436	2301477094	1627.904587	0.670275	217770642	14.643487	10.275077	0.644444	-2.949983	4.779795	23.8
KC	86.078559	2587793017	952.416745	0.369894	412.457447	-5.713616	-10.410973	0.613466	-2.891172	4.919277	29.3

Pitch Usage

```
In [8]: plt.figure(figsize = (8, 6))

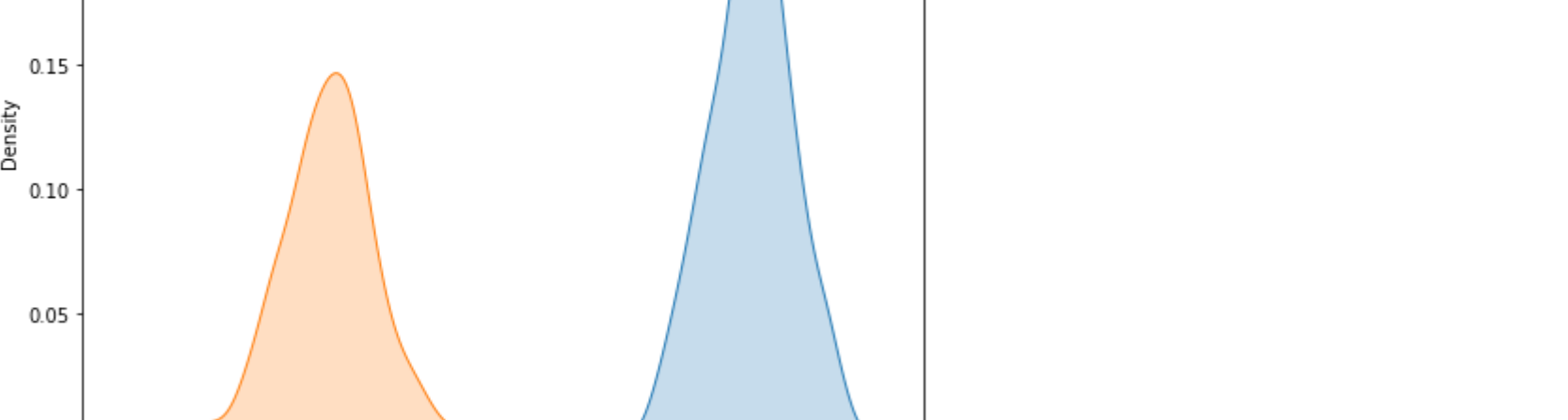
dist = round(kimbrl.pitch_type.value_counts(normalize = True), 2)
color = sns.color_palette('coolwarm')
plt.title('Distribution of Pitch Types - Craig Kimbrl', fontdict = font_title, pad = 15);
```



Pitch Usage by Batter Handedness

```
In [9]: blue = '#00007F'
red = '#D50032'

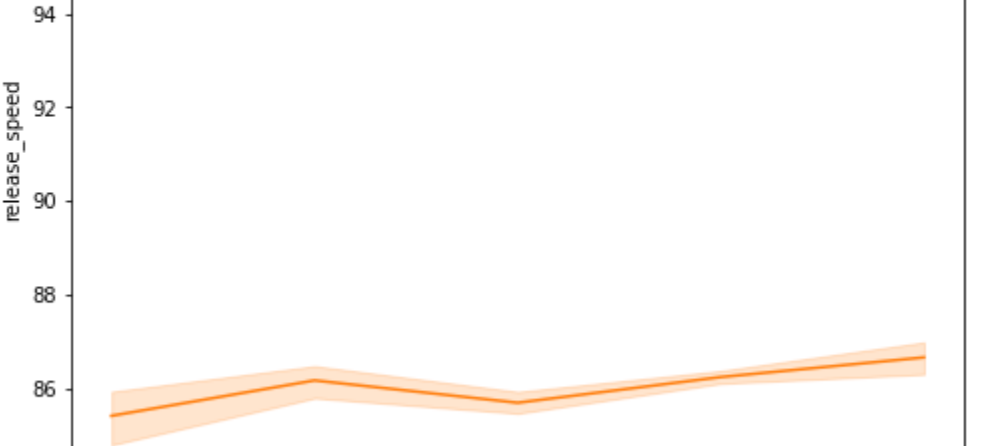
fig, ax = plt.subplots(1, 2, figsize = (20, 6))
fig.suptitle('Pitch Usage by Batter Handedness', fontsize = 16, fontweight = 'bold')
r = kimbrl.pitch_type.sort_values(ascending = False)
weights = np.ones(len(dist_r)) / len(dist_r)
ax[0].hist(dist_r, weights = np.ones(len(dist_r)) / len(dist_r), color = blue)
ax[0].axis(ax[0].get_major_formatter(PercentFormatter(1)))
ax[0].set_title('Distribution of Pitch Types - RHH', fontdict = font_title, pad = 15)
dist = kimbrl.pitch_type.sort_values(ascending = False)
ax[1].hist(dist_l, weights = np.ones(len(dist_l)) / len(dist_l), color = red)
ax[1].axis(ax[1].get_major_formatter(PercentFormatter(1)))
ax[1].set_title('Distribution of Pitch Types - LHH', fontdict = font_title, pad = 15);
```



Velocity by Pitch Type

```
In [10]: plt.figure(figsize = (8, 6))

sns.kdeplot(data = kimbrl, x = 'release_speed', shade = 'fill', hue = 'pitch_type',
            hue_order = order, palette = 'tab10')
plt.title('Distribution of Velocity by Pitch Type - Craig Kimbrl', fontdict = font_title, pad = 12);
```



Pitch Velocity by Pitch Number

```
In [11]: plt.figure(figsize = (8, 6))

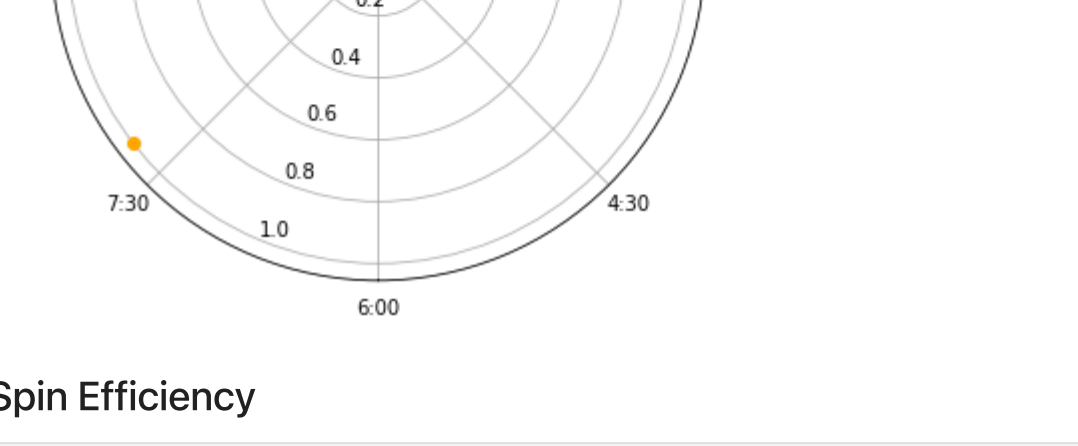
sns.lineplot(data = kimbrl, x = 'inning', y = 'release_speed', hue = 'pitch_type',
             hue_order = order, palette = 'tab10')
plt.title('Pitch Velocity by Pitch Number', fontdict = font_title, pad = 15);
```



Pitcher Stuff

Spin Rate by Pitch Type

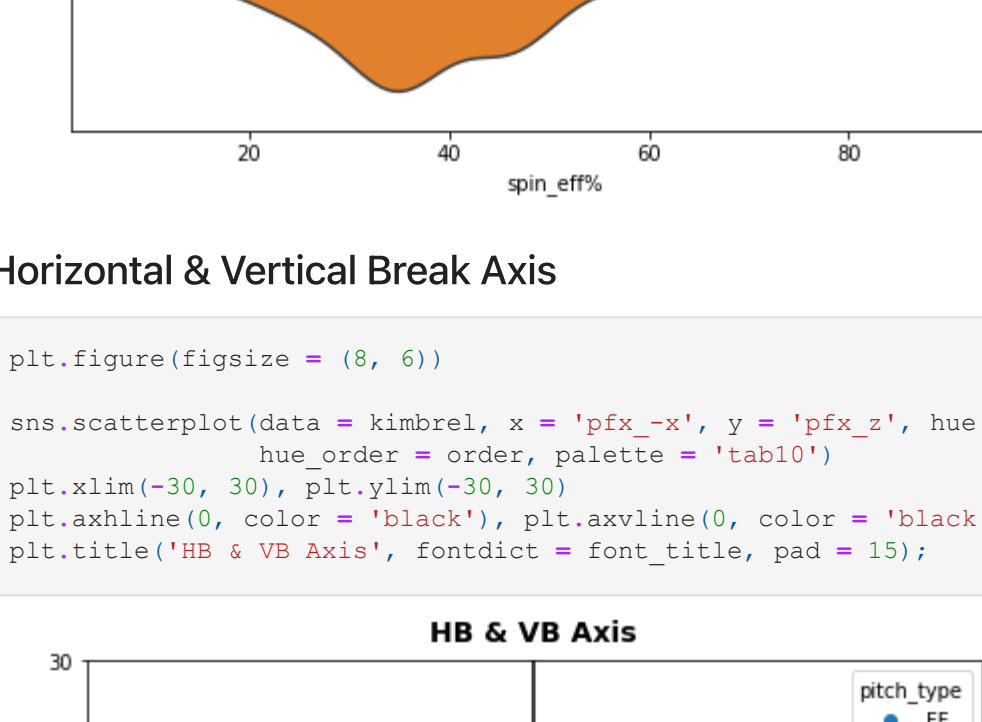
```
In [12]: g = sns.FacetGrid(kimbrl, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4)
g.map(sns.kdeplot, 'release_spin_rate', palette = 'tab10');
```



Spin Axis

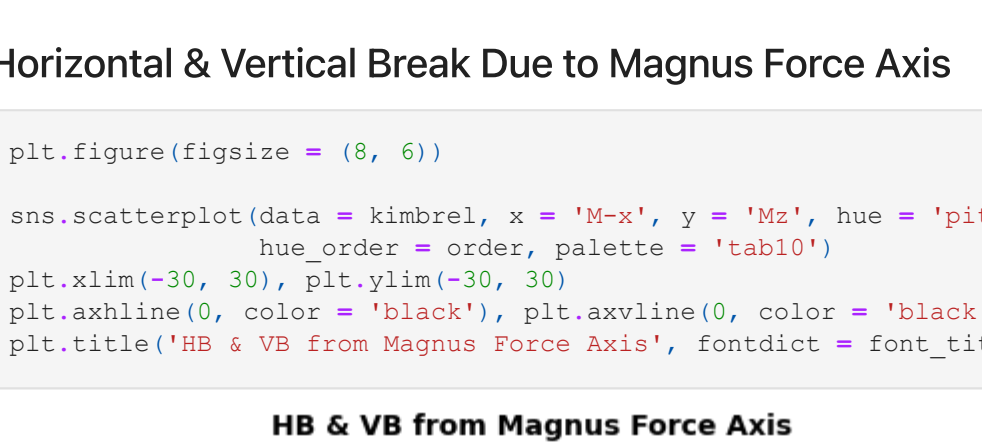
```
In [13]: ax = plt.figure(figsize = (8, 6))

ax = plt.subplot(polar = True, theta_direction = 1)
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')
ticks = ['16:00', '7:30', '9:00', '10:30', '12:00', '1:30', '3:00', '4:30']
ax.set_xticklabels(ticks), ax.legend(bbox_to_anchor=(1.5, .58)), ax.set_theta_zero_location('S')
ax.set_title('Spin Axis', fontdict = font_title, pad = 15);
```



Spin Efficiency

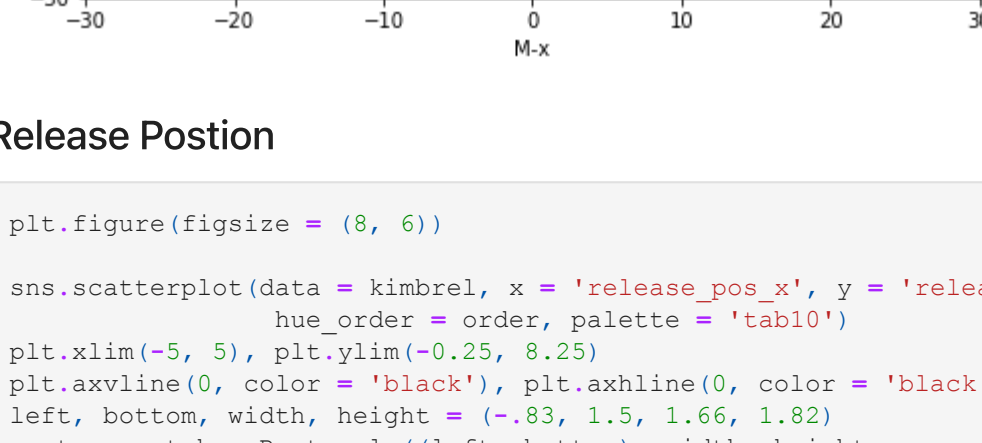
```
In [14]: sns.kdeplot(data = kimbrl, x = 'spin_eff', y = 'pitch_type', kind = 'violin', aspect = 1.5, palette = 'tab10')
sns.kdeplot(ax = ax[1][1], data = ff, x = 'spin_eff', y = 'pitch_type', fill = True, hue = 'pitch_type', palette = 'tab10');
```



Horizontal & Vertical Break Axis

```
In [15]: plt.figure(figsize = (8, 6))

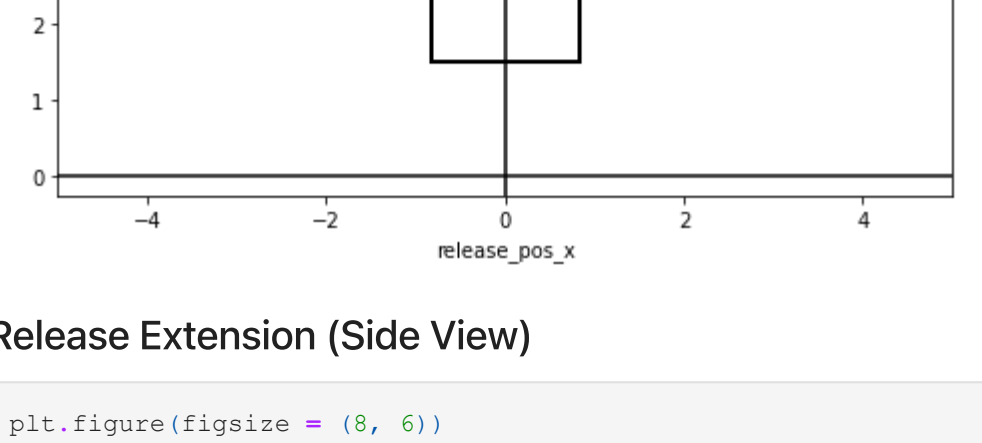
sns.scatterplot(data = kimbrl, x = 'pfx_x', y = 'pfx_z', hue = 'pitch_type',
               hue_order = order, palette = 'tab10')
plt.xlim(-30, 30), plt.ylim(-30, 30)
plt.axline(0, color = 'black'), plt.axline(0, color = 'black')
plt.title('HB & VB from Magnitude', fontdict = font_title, pad = 15);
```



Horizontal & Vertical Break Due to Magnus Force Axis

```
In [16]: plt.figure(figsize = (8, 6))

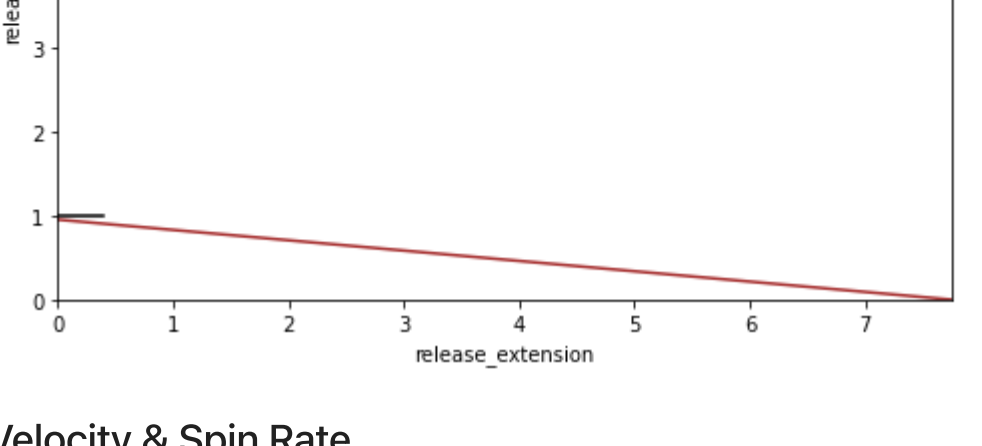
sns.scatterplot(data = kimbrl, x = 'Mx', y = 'Mz', hue = 'pitch_type',
               hue_order = order, palette = 'tab10')
plt.xlim(-30, 30), plt.ylim(-30, 30)
plt.axline(0, color = 'black'), plt.axline(0, color = 'black')
ax[1][0].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[1][1].set_title('HB & VB from Magnus Force Axis', fontdict = font_title, pad = 15);
```



Release Postion

```
In [17]: plt.figure(figsize = (8, 6))

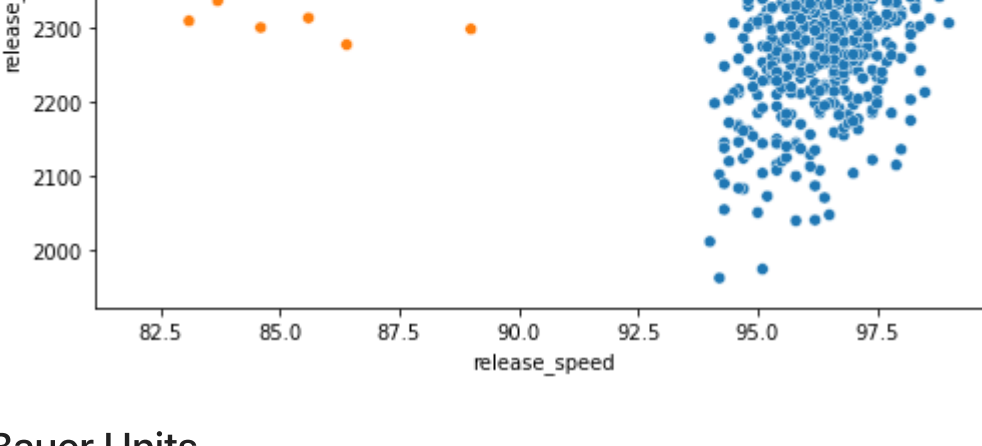
sns.scatterplot(data = kimbrl, x = 'release_pos_x', y = 'release_pos_z', hue = 'pitch_type',
               hue_order = order, palette = 'tab10')
plt.xlim(-5, 5), plt.ylim(-0.25, 0.25)
plt.axline(0, color = 'black'), plt.axline(0, color = 'black')
left, bottom, width, height = (-8.5, 1.5, 1.66, 1.62)
rect = mpatches.Rectangle(left, bottom, width, height,
                        fill = False, color = 'black', linewidth = 2)
plt.gca().add_patch(rect)
plt.title('Release Position From Hitter Perspective', fontdict = font_title, pad = 15);
```



Release Extension (Side View)

```
In [18]: plt.figure(figsize = (8, 6))


sns.scatterplot(data = kimbrl, x = 'release_extension', y = 'release_pos_z', hue = 'pitch_type',
               hue_order = order, palette = 'tab10')
plt.xlim(0, 7.75), plt.ylim(-0.75, 0.75)
plt.axline(0, .95), color = 'brown', plt.axline(1, 0, .05, color = 'black')
plt.legend(bbox_to_anchor=(1.25, .7))
plt.title('Release Extension (Side View)', fontdict = font_title, pad = 15);
```



Velocity & Spin Rate

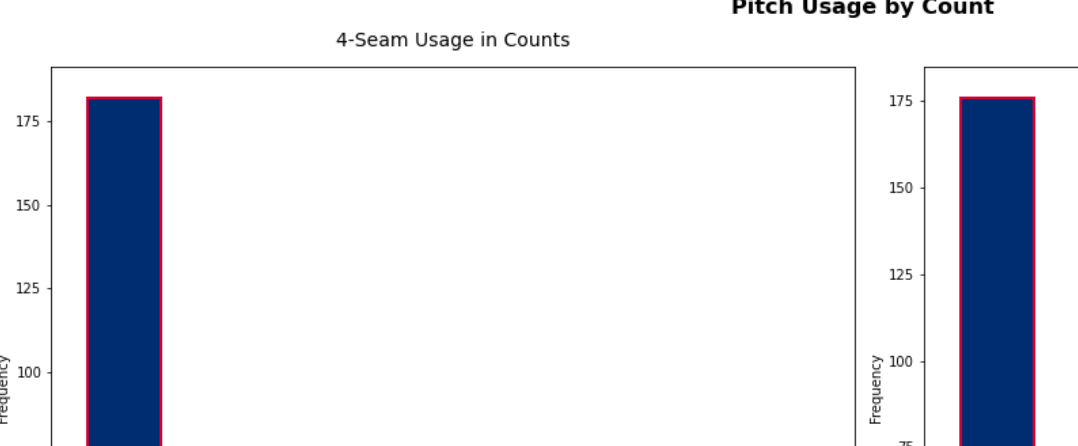
```
In [19]: plt.figure(figsize = (8, 6))

sns.scatterplot(data = kimbrl, x = 'release_speed', y = 'release_spin_rate', hue = 'pitch_type',
               hue_order = order, palette = 'tab10')
plt.title('Velo vs Spin Rate - By Pitch Type', fontdict = font_title, pad = 15);
```



Bauer Units

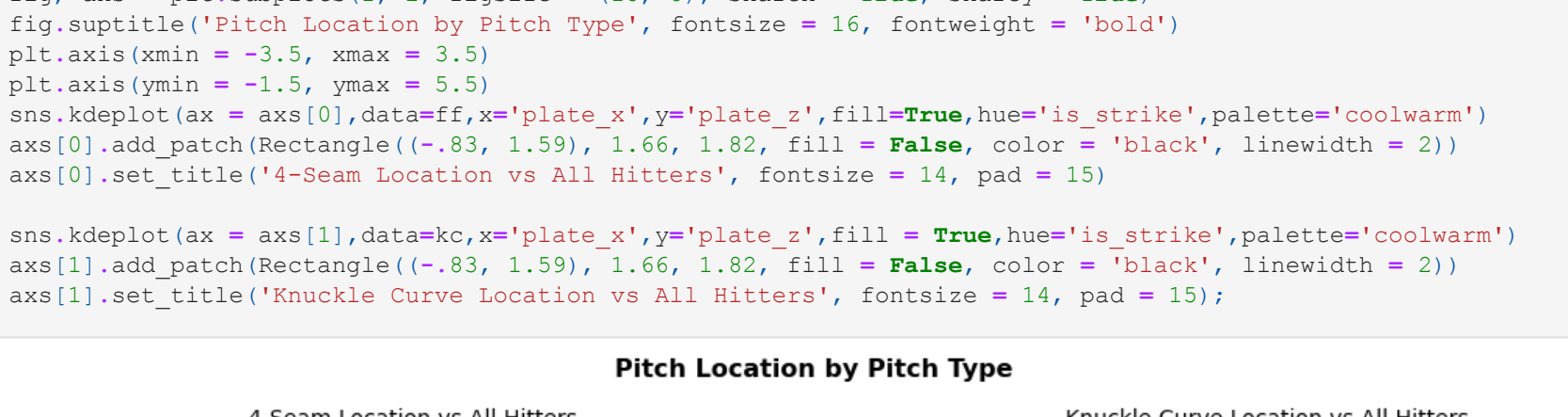
```
In [20]: g = sns.FacetGrid(kimbrl, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4)
g.map(sns.kdeplot, 'bauer_units', palette = 'tab10');
```



Count Breakdown

Pitch Usage by Count

```
In [21]: fig, ax = plt.subplots(1, 2, figsize = (18, 8))
fig.suptitle('Pitch Usage by Count', fontsize = 16, fontweight = 'bold')
plt.subplot(ax[0], xlabel = 'Count')
ax[0].hist(ff['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
ax[0].set_title('4-Seam Usage in Counts', fontdict = font_title, pad = 15)
ax[1].hist(kc['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
plt.tight_layout();
```

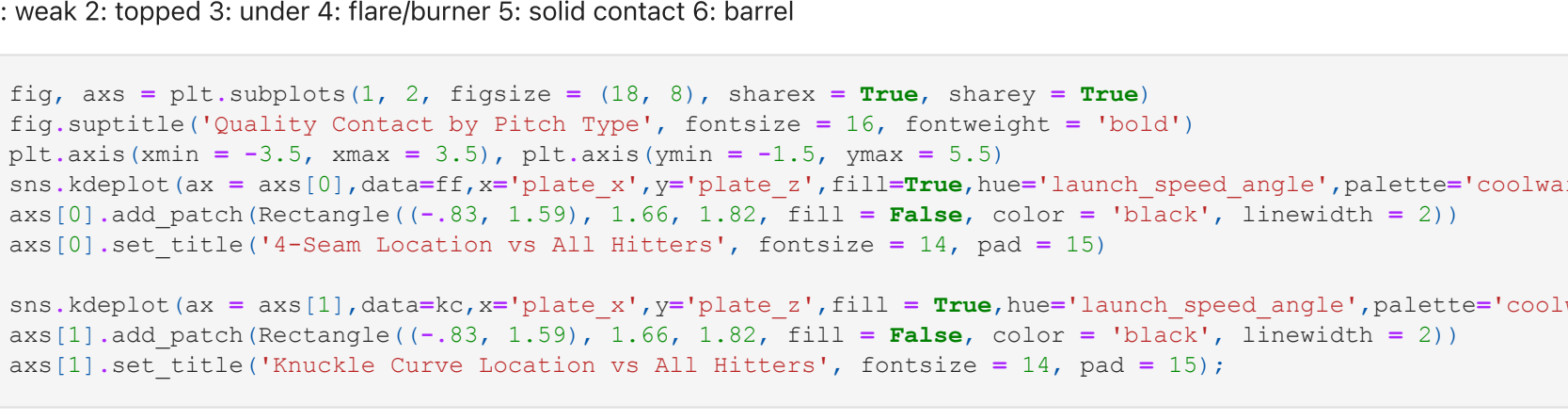


Heatmaps

'All From Hitters' Perspective

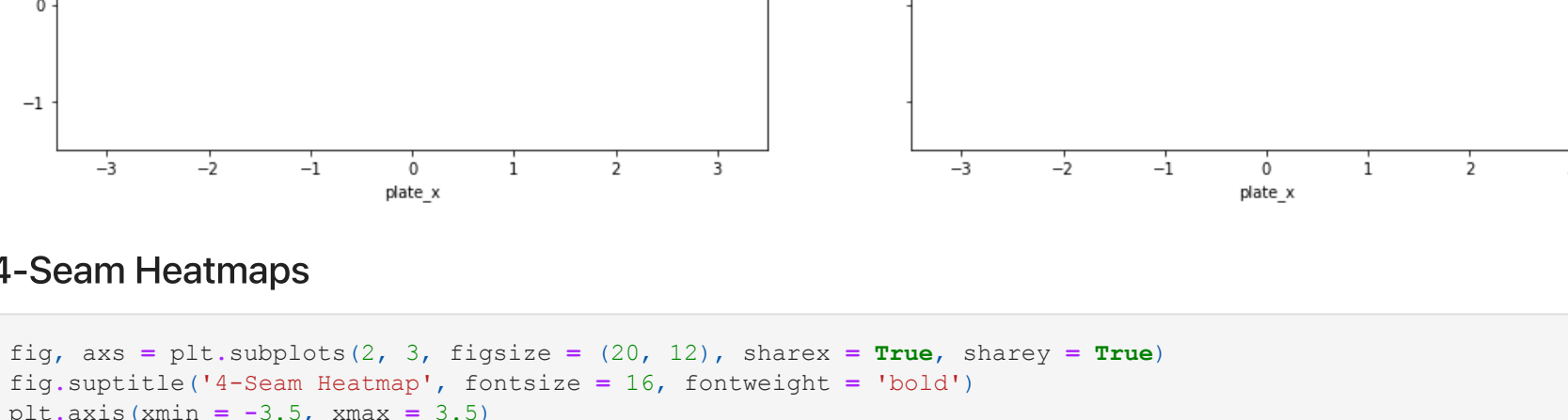
Pitch Location by Pitch Type

```
In [22]: fig, ax = plt.subplots(1, 2, figsize = (18, 8), sharex = True, sharey = True)
fig.suptitle('Pitch Location by Pitch Type', fontsize = 16, fontweight = 'bold')
plt.subplot(ax[0], xmargin = 3.5, ymargin = 3.5, plt.ylim(-1.5, ymargin = 3.5))
sns.kdeplot(ax = ax[0], data = ff, x = 'plate_x', y = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
ax[0].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[0].set_title('4-Seam Location vs All Hitters', fontdict = font_title, pad = 15)
sns.kdeplot(ax = ax[1], data = kc, x = 'plate_x', y = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
ax[1].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[1].set_title('Knuckle Curve Location vs All Hitters', fontdict = font_title, pad = 15);
```



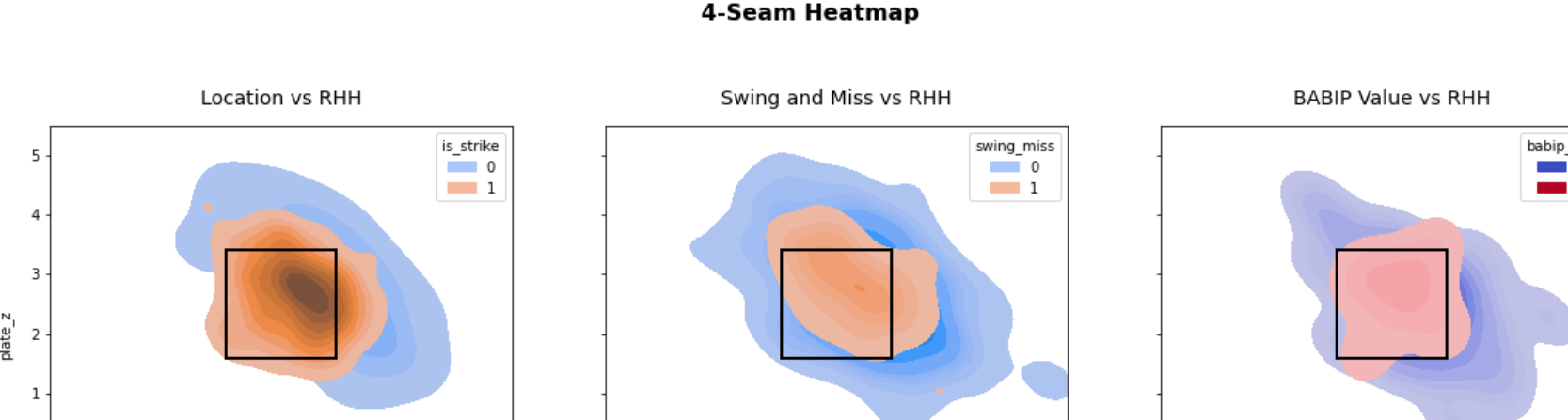
Quality Contact by Pitch Type

```
In [23]: fig, ax = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('Quality Contact by Pitch Type', fontsize = 16, fontweight = 'bold')
plt.subplot(ax[0], xmargin = 3.5, ymargin = 3.5)
sns.kdeplot(ax = ax[0], data = ff, x = 'plate_x', y = 'plate_z', fill = True, hue = 'launch_speed_angle', palette = 'coolwarm')
ax[0].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[0].set_title('4-Seam Location vs All Hitters', fontdict = font_title, pad = 15)
sns.kdeplot(ax = ax[1], data = kc, x = 'plate_x', y = 'plate_z', fill = True, hue = 'launch_speed_angle', palette = 'coolwarm')
ax[1].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[1].set_title('Knuckle Curve Location vs All Hitters', fontdict = font_title, pad = 15);
```



4-Seam Heatmaps

```
In [24]: fig, ax = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('4-Seam Heatmap', fontsize = 16, fontweight = 'bold')
plt.subplot(ax[0], xmargin = 3.5, ymargin = 3.5)
sns.kdeplot(ax = ax[0], data = ff, x = 'plate_x', y = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
ax[0].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[0].set_title('Location vs RHH', fontdict = font_title, pad = 15)
sns.kdeplot(ax = ax[1], data = ff, x = 'plate_x', y = 'plate_z', fill = True, hue = 'swing_miss', palette = 'coolwarm')
ax[1].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[1].set_title('Swing and Miss vs RHH', fontdict = font_title, pad = 15)
sns.kdeplot(ax = ax[2], data = ff, x = 'plate_x', y = 'plate_z', fill = True, hue = 'BABIP_value', palette = 'coolwarm')
ax[2].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[2].set_title('BABIP Value vs RHH', fontdict = font_title, pad = 15);
```



Knuckle Curve Heatmaps

```
In [25]: fig, ax = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('Knuckle Curve Heatmap', fontsize = 16, fontweight = 'bold')
plt.subplot(ax[0], xmargin = 3.5, ymargin = 3.5)
sns.kdeplot(ax = ax[0], data = kc, x = 'plate_x', y = 'plate_z', fill = True, hue = 'is_strike', palette = 'coolwarm')
ax[0].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[0].set_title('Location vs RHH', fontdict = font_title, pad = 15)
sns.kdeplot(ax = ax[1], data = kc, x = 'plate_x', y = 'plate_z', fill = True, hue = 'swing_miss', palette = 'coolwarm')
ax[1].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[1].set_title('Swing and Miss vs RHH', fontdict = font_title, pad = 15)
sns.kdeplot(ax = ax[2], data = kc, x = 'plate_x', y = 'plate_z', fill = True, hue = 'BABIP_value', palette = 'coolwarm')
ax[2].add_patch(Rectangle((-8.5, 1.59), 1.66, 1.62, fill = False, color = 'black', linewidth = 2))
ax[2].set_title('BABIP Value vs RHH', fontdict = font_title, pad = 15);
```

