Craig Kimbrel

Notebook Contents

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]:
```
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)
```

(986, 84)

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]:

pitch_type	release_speed	release_pos_x	release_pos_z	batter	pitcher	zone	hit_location	balls	strikes	gam
FF	96.517436	-2.949983	4.779795	596628.485470	518886.0	8.153846	4.572650	1.088889	0.979487	
KC	86.076559	-2.891172	4.919277	604564.184539	518886.0	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_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_z': 'r_height', 'release_pos_x': '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']
# all hitters
ff = kimbrel.loc[kimbrel['pitch_type'] == 'FF']
kc = kimbrel.loc[kimbrel['pitch_type'] == 'KC']
# RHH
r_ff = r_kimbrel.loc[r_kimbrel['pitch_type'] == 'FF']
r_kc = r_kimbrel.loc[r_kimbrel['pitch_type'] == 'KC']
# LHH
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]:

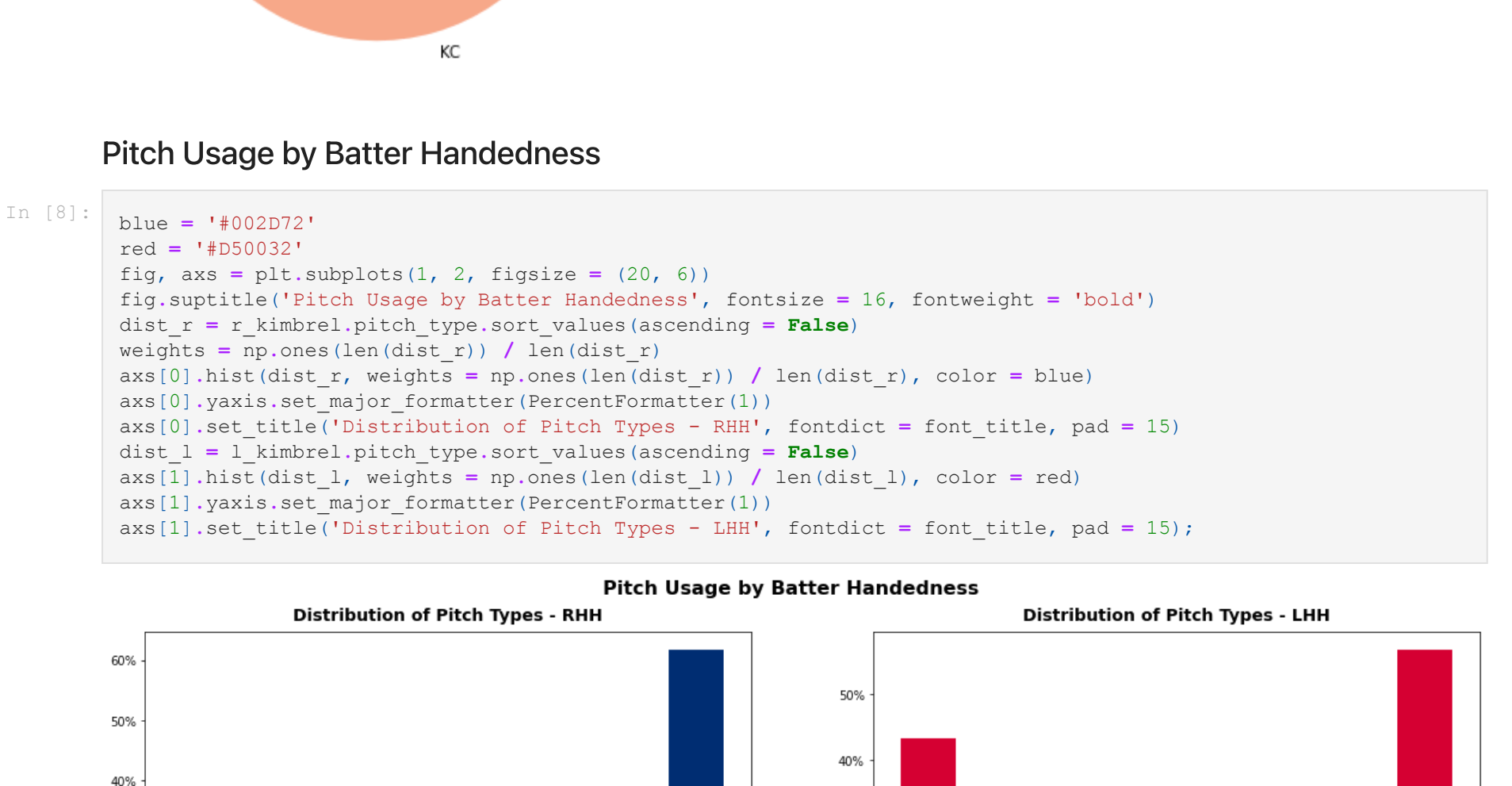
pitch_type	velo	spin	true_spin	spin_eff	spin_axis	hb	vb	strike	r_side	r_height	bauer_
FF	96.517436	2301.417094	1527.904587	0.670275	217.770642	14.643487	10.275077	0.644444	-2.949983	4.779795	23.8
KC	86.076559	2557.793017	952.415745	0.369894	412.457447	-5.713616	-10.410973	0.613466	-2.891172	4.919277	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%%')
plt.title('Distribution of Pitch Types - Craig Kimbrel', fontdict = font_title, pad = 15);
```

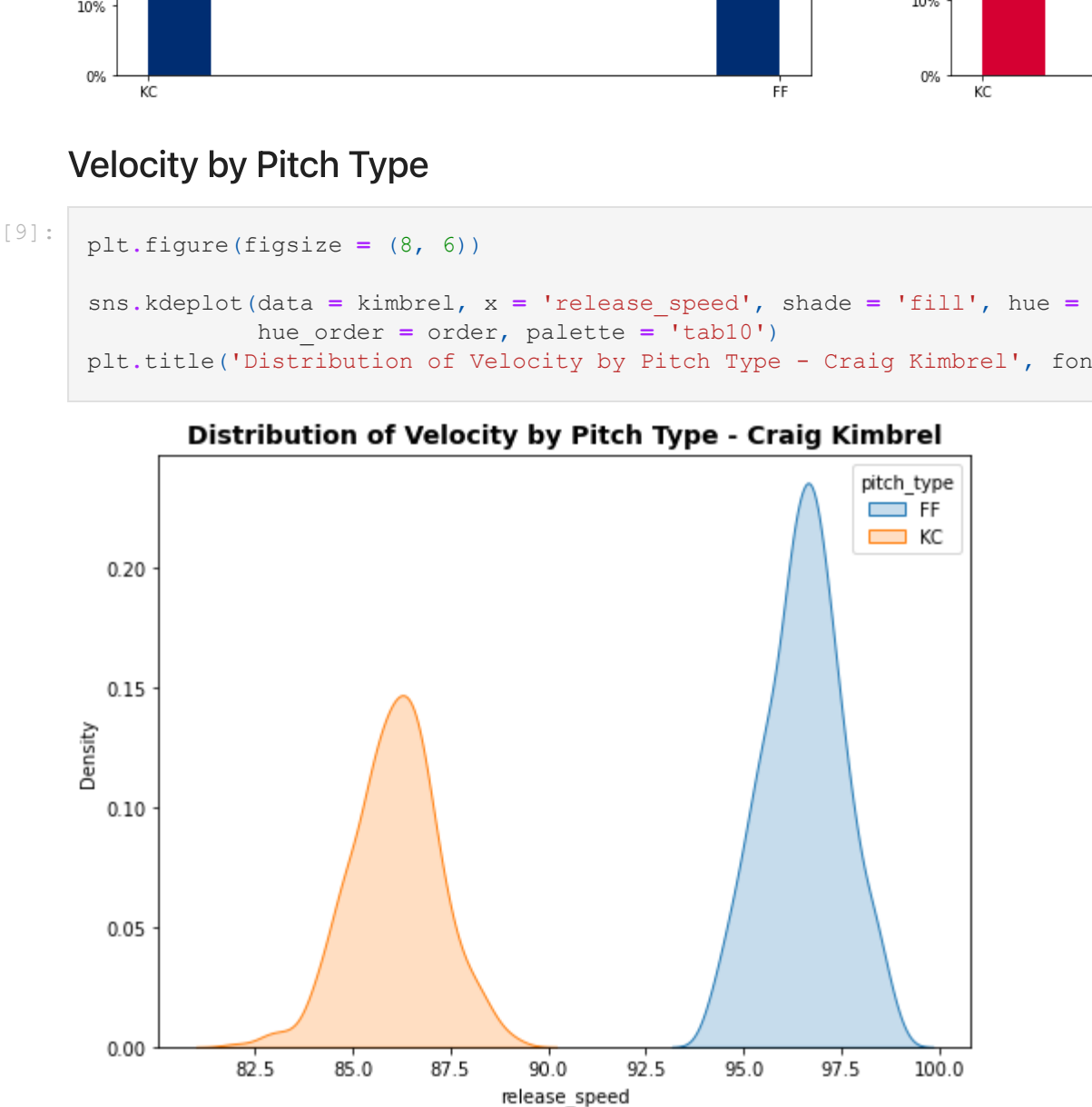
Pitch Usage by Batter Handedness

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.value_counts(ascending = False)
weights = np.ones(len(dist_r)) / len(dist_r)
axs[0].hist(dist_r, weights = np.ones(len(dist_r)) / len(dist_r), color = blue)
axs[0].yaxis.set_major_formatter(PercentFormatter(1))
axs[0].set_title('Distribution of Pitch Types - RHH', fontdict = font_title, pad = 15)
dist_l = l_kimbrel.pitch_type.value_counts(ascending = False)
axs[1].hist(dist_l, weights = np.ones(len(dist_l)) / len(dist_l), color = red)
axs[1].yaxis.set_major_formatter(PercentFormatter(1))
axs[1].set_title('Distribution of Pitch Types - LHH', fontdict = font_title, pad = 15);
```



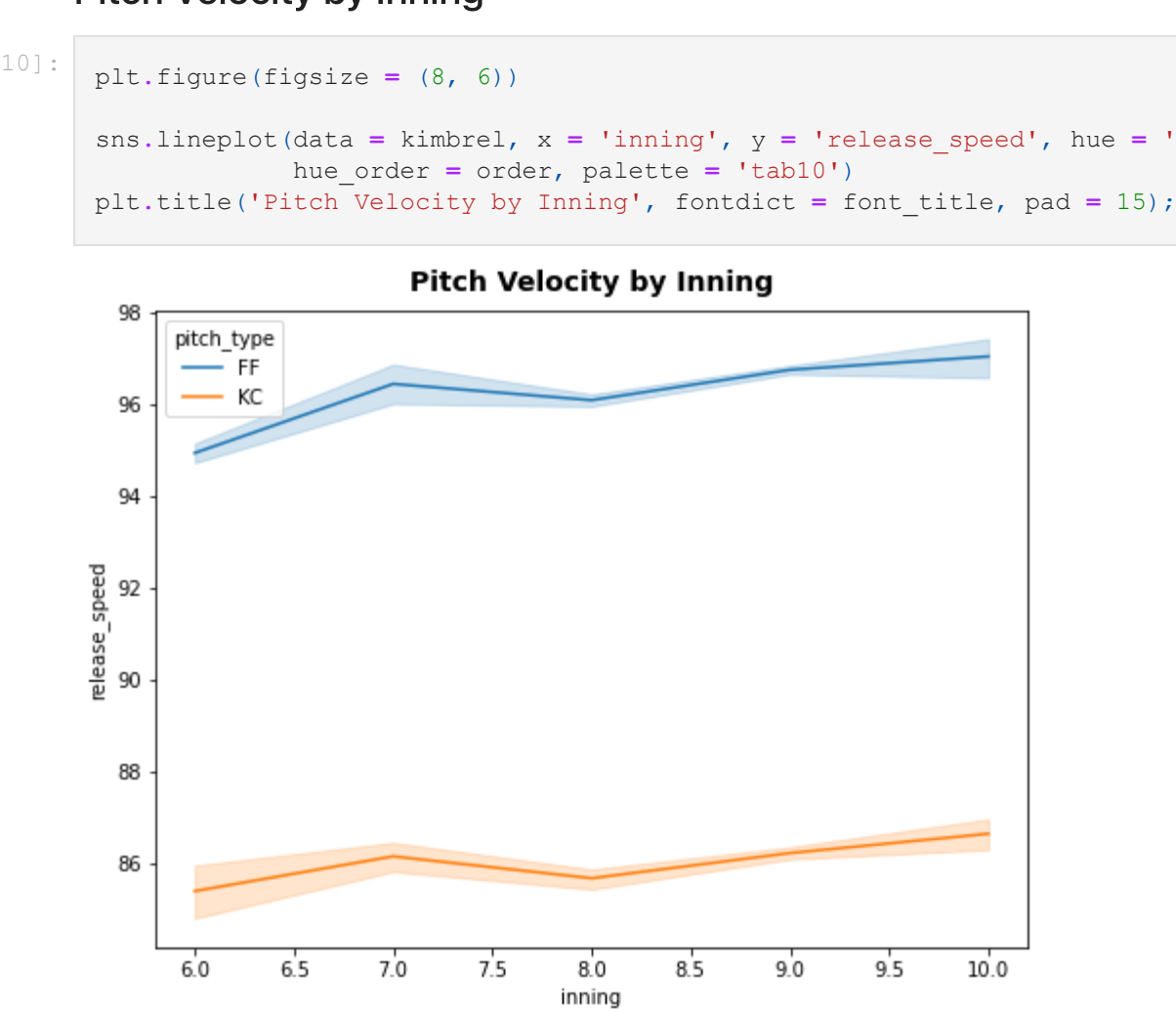
Velocity by Pitch Type

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

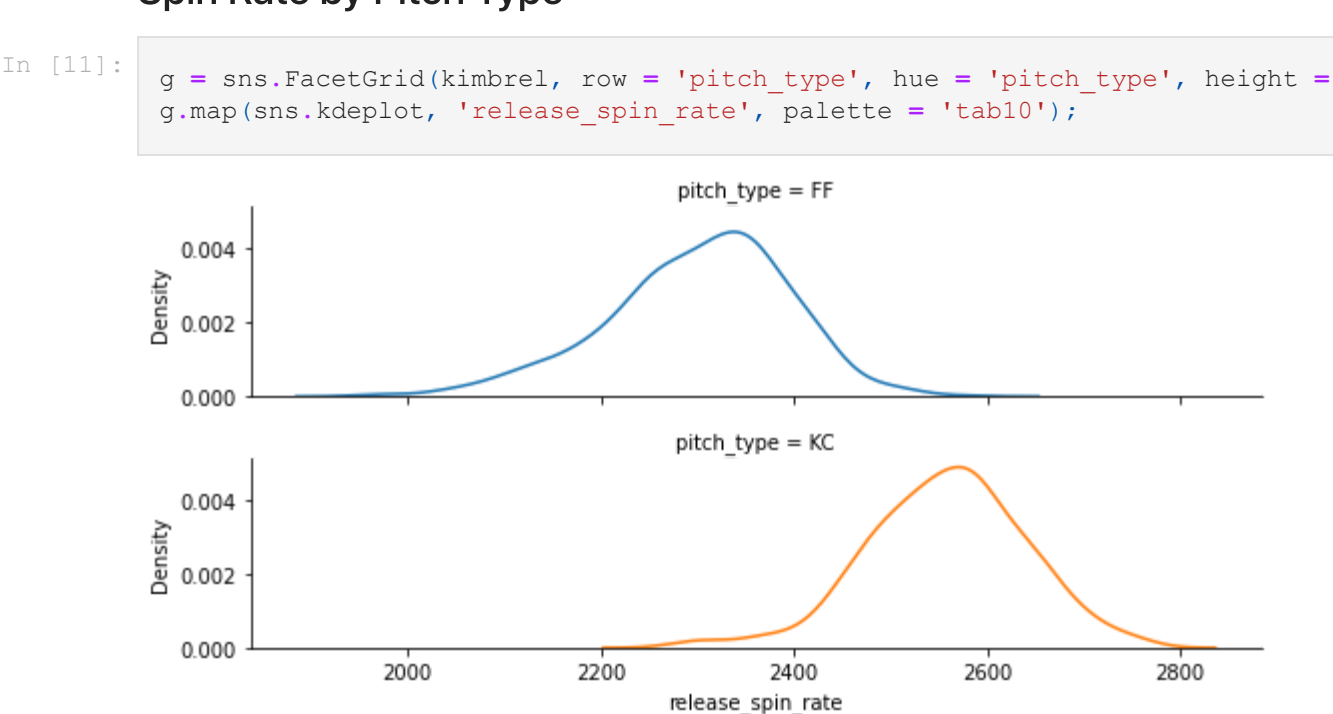
In [10]:
```
plt.figure(figsize = (8, 6))

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

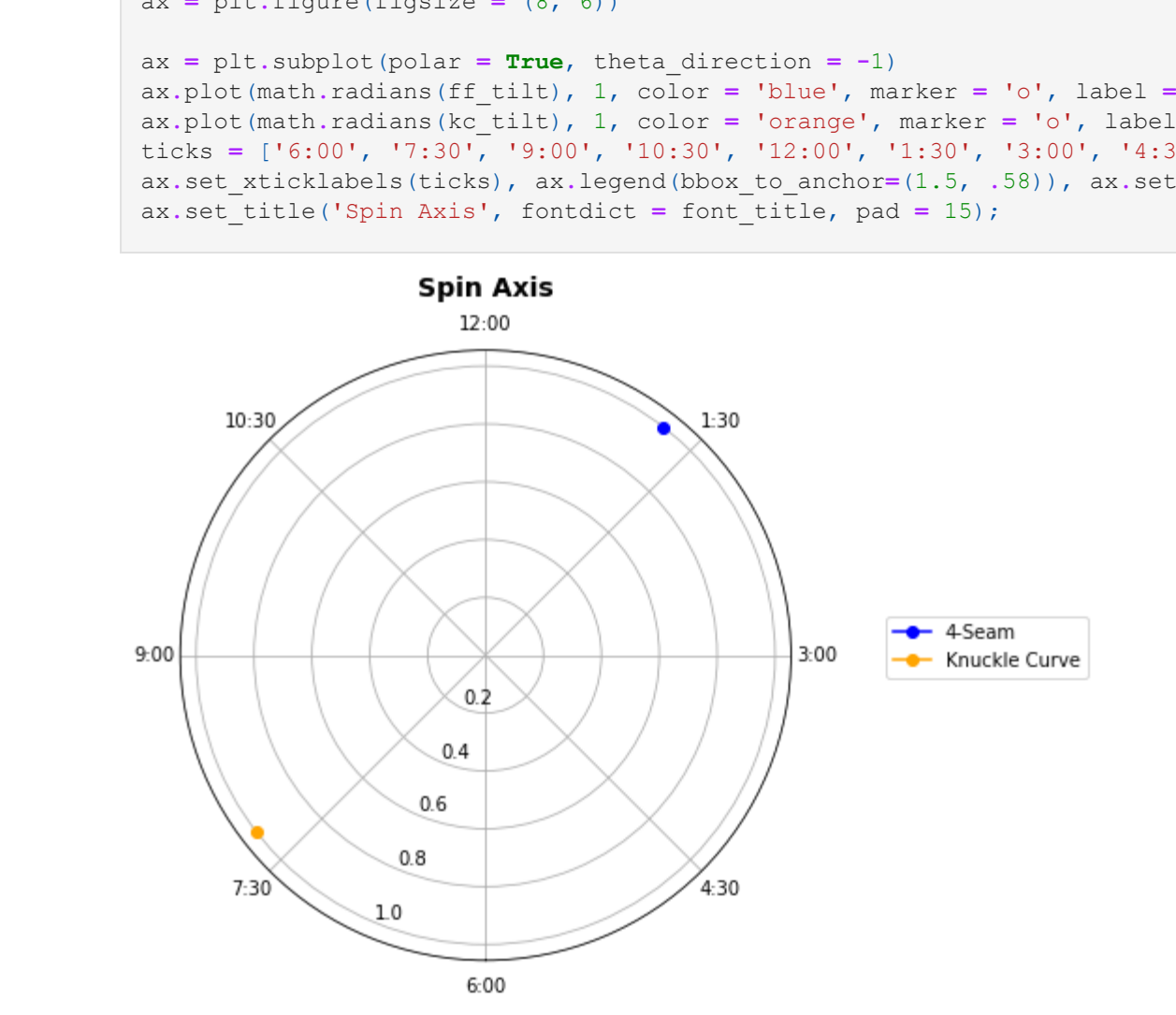
Spin Rate by Pitch Type

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');
```



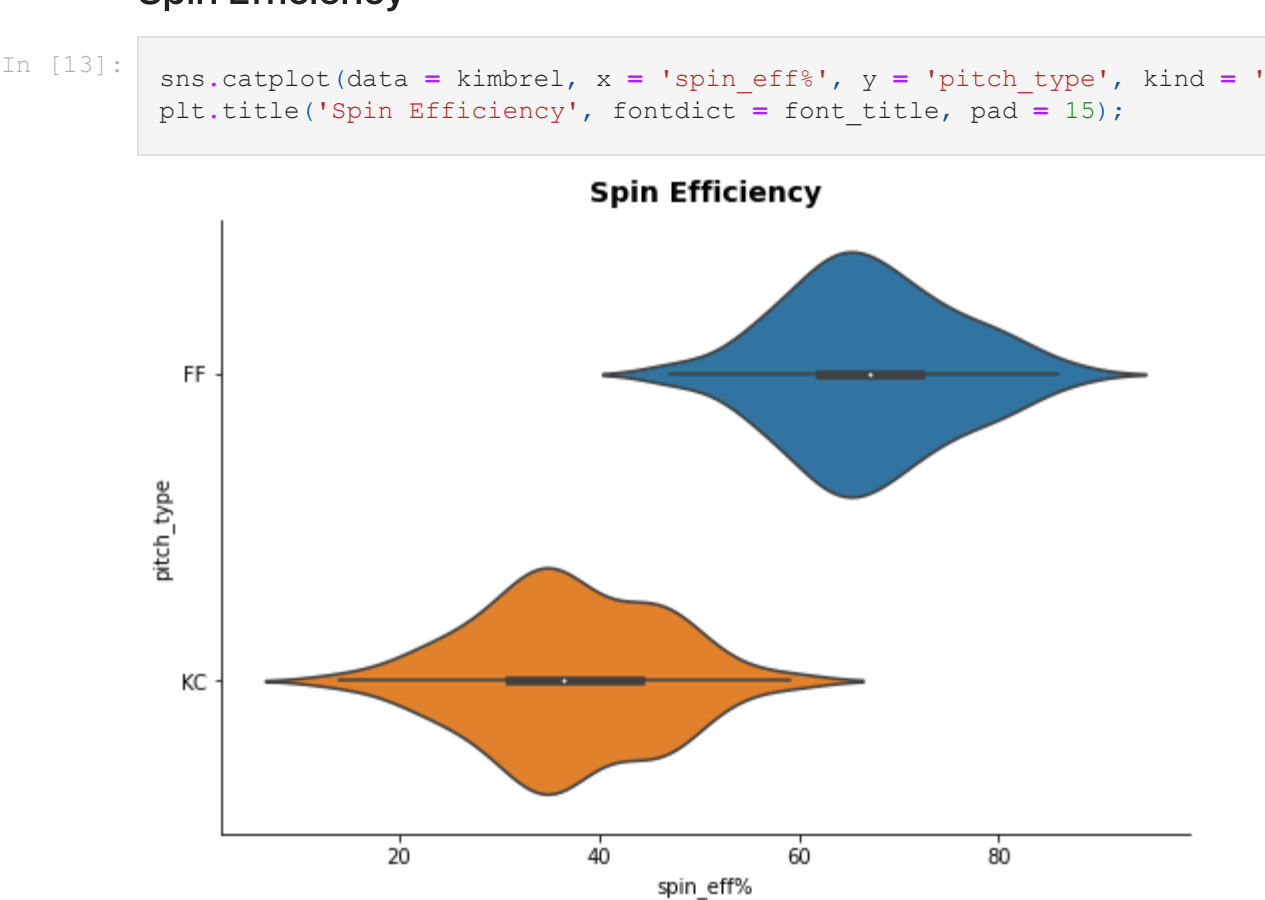
Spin Axis

In [12]:
```
ax = plt.figure(figsize = (8, 6))

ax = plt.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 = ['6:00', '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);
```



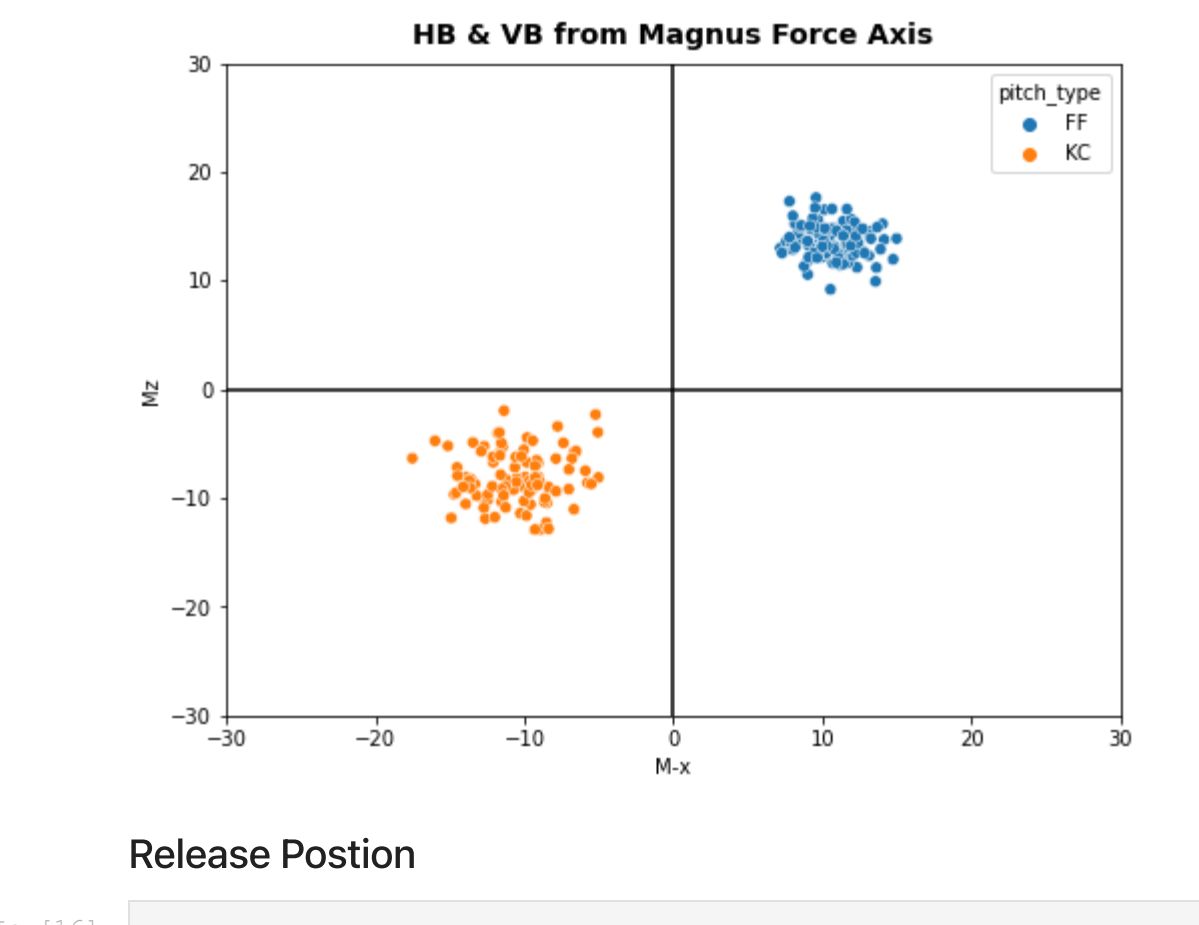
Spin Efficiency

In [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);
```



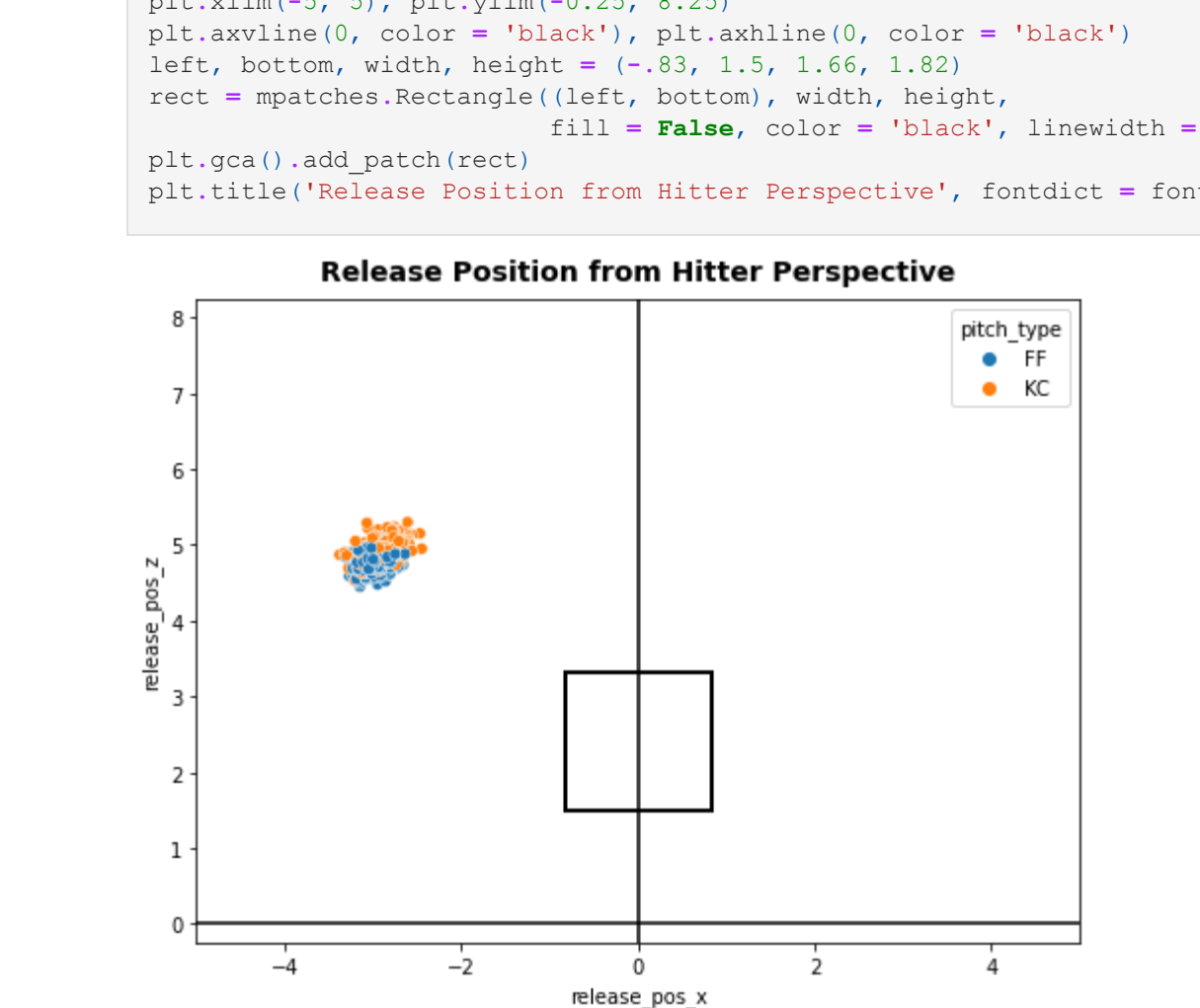
Horizontal & Vertical Break Axis

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.axhline(0, color = 'black'), plt.axvline(0, color = 'black')
plt.title('HB & VB Axis', fontdict = font_title, pad = 15);
```



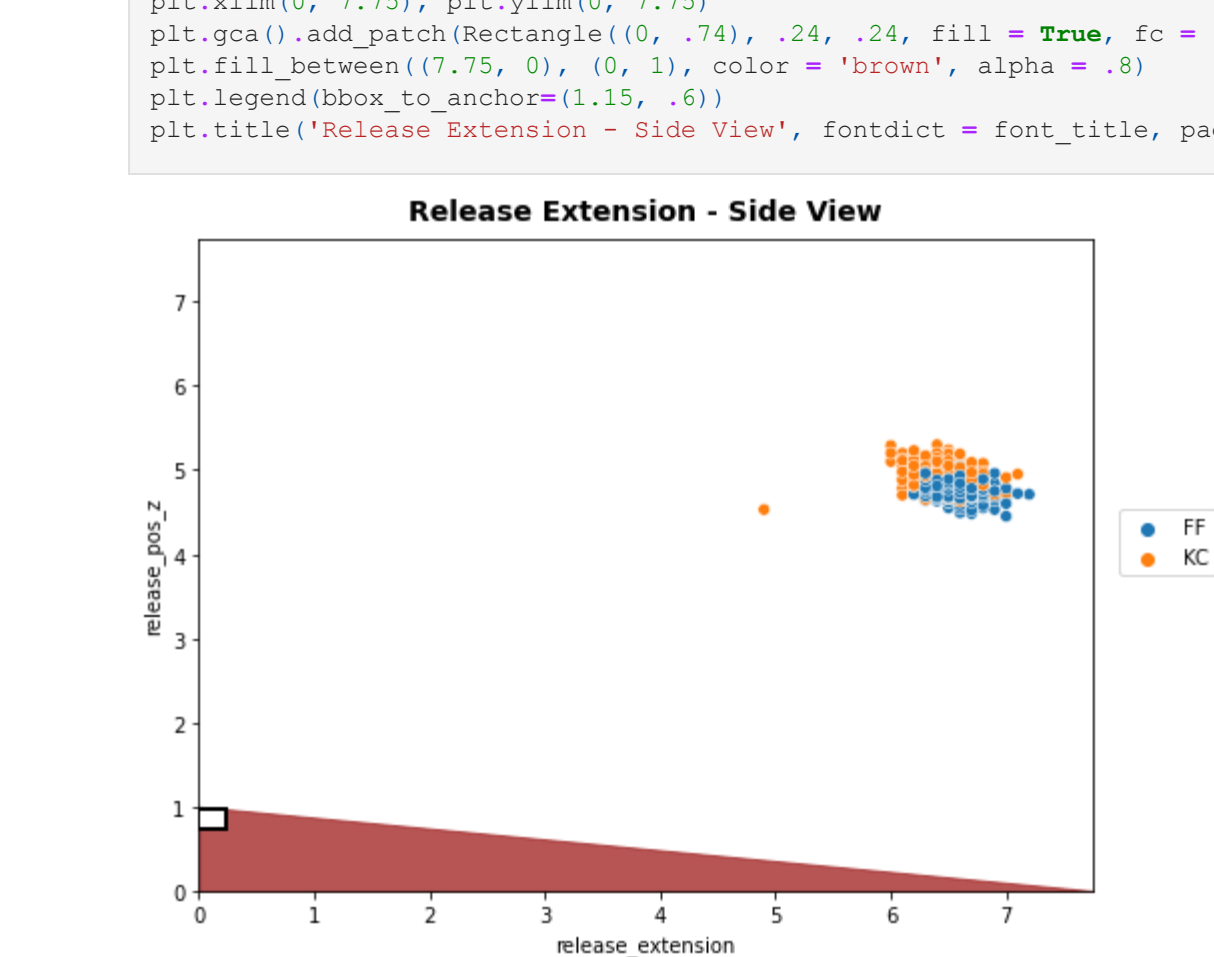
Horizontal & Vertical Break Due to Magnus Force Axis

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

sns.scatterplot(data = kimbrel, x = 'M-x', y = 'Mz', hue = 'pitch_type',
                hue_order = order, palette = 'tab10')
plt.xlim(-30, 30), plt.ylim(-30, 30)
plt.axhline(0, color = 'black'), plt.axvline(0, color = 'black')
plt.title('HB & VB from Magnus Force Axis', fontdict = font_title, pad = 15);
```



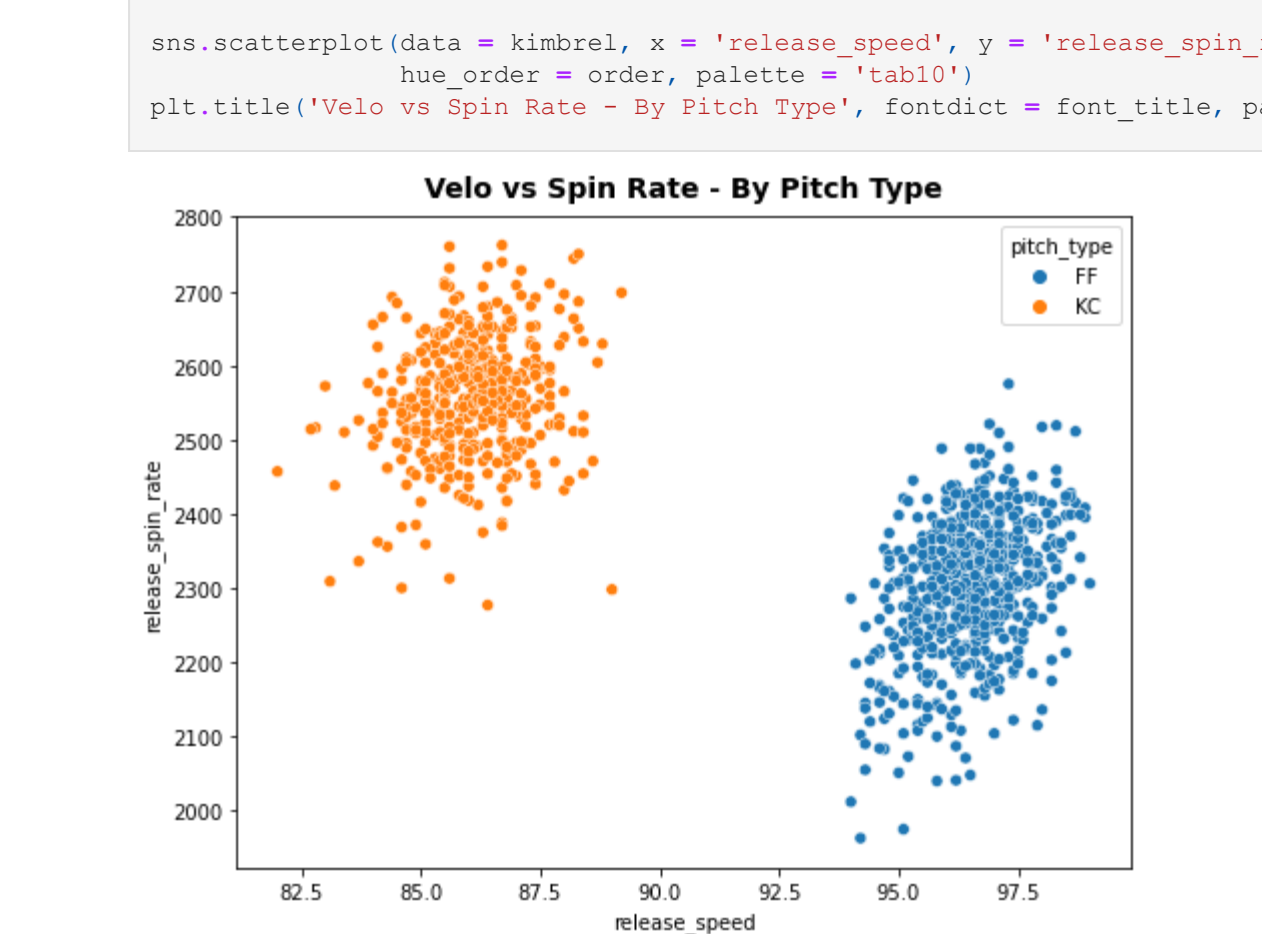
Release Postion

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, 8.25)
plt.axvline(0, color = 'black'), plt.axvline(0, color = 'black')
left, bottom, width, height = (-.83, 1.5, 1.66, 1.82)
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);
```



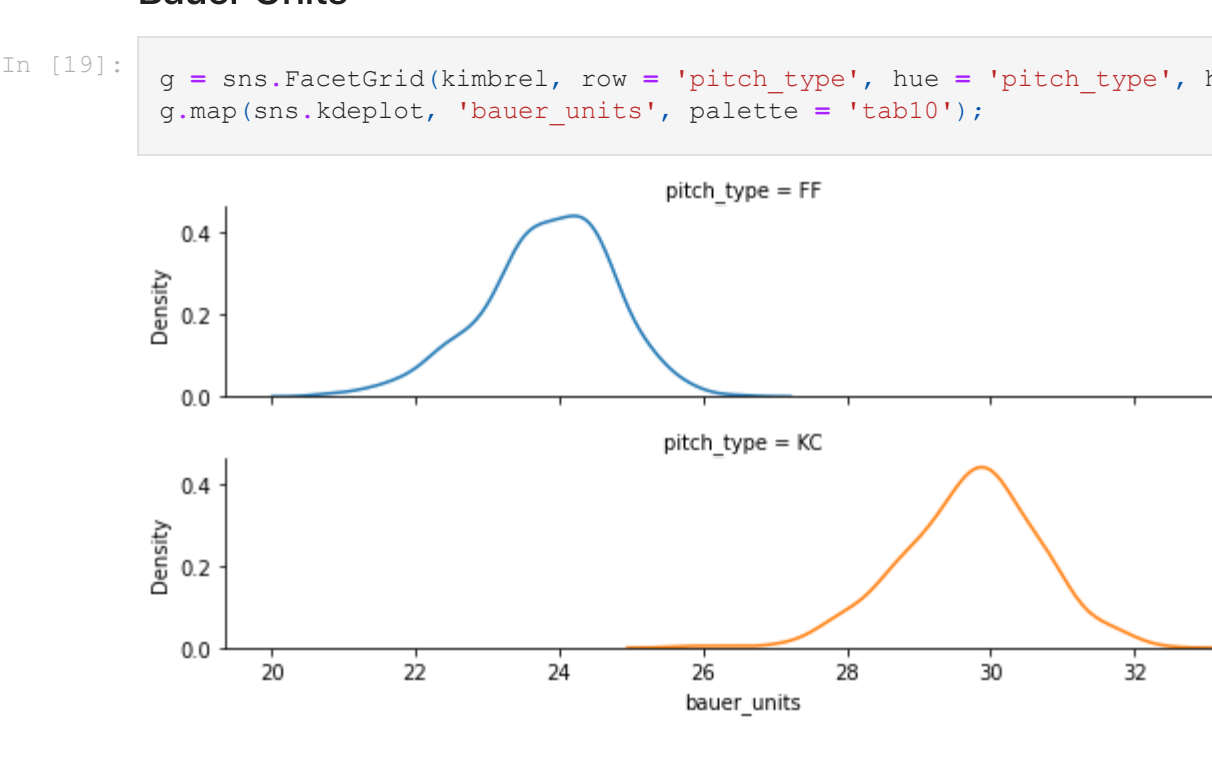
Release Extension - Side View

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

sns.scatterplot(data = kimbrel, x = 'release_extension', y = 'release_pos_z', hue = 'pitch_type',
                hue_order = order, palette = 'tab10')
plt.xlim(0, 7.75), plt.ylim(0, 7.75)
plt.gca().add_patch(Rectangle((0, .74), .24, .24, fill = True, fc = 'white', ec = 'black', linewidth = 2))
plt.fill_between([3.75, 0], [0, 1], color = 'brown', alpha = .8)
plt.legend(bbox_to_anchor=(1.15, .5))
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('Velo vs Spin Rate - By Pitch Type', fontdict = font_title, pad = 15);
```



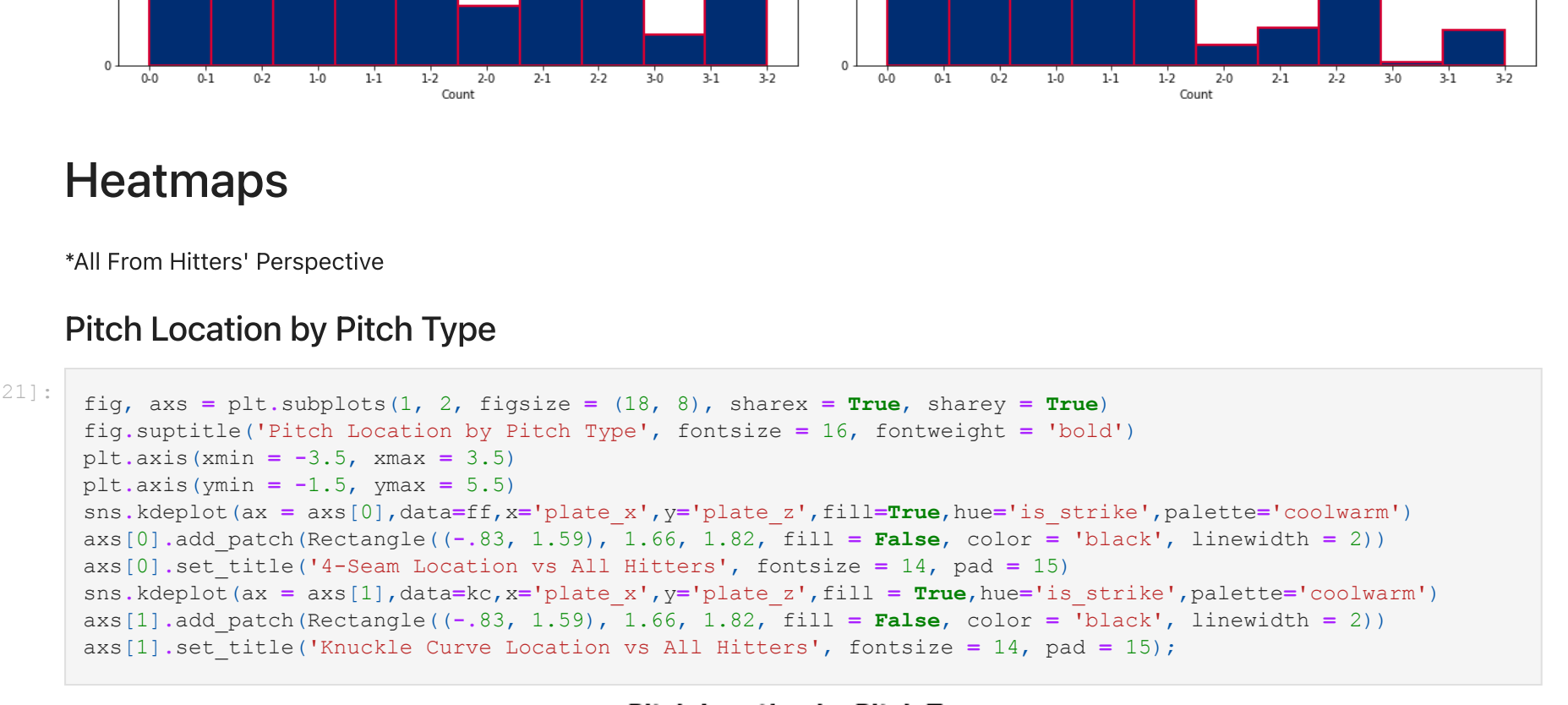
Bauer Units

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, 6))
fig.suptitle('Pitch Usage by Count', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0], xlabel = 'Count')
plt.setp(axs[0], ylabel = 'Frequency')
axs[0].hist(ff['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].hist(kc['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
axs[1].set_title('Knuckle Curve Usage in Counts', fontsize = 14, pad = 15)
plt.tight_layout();
```

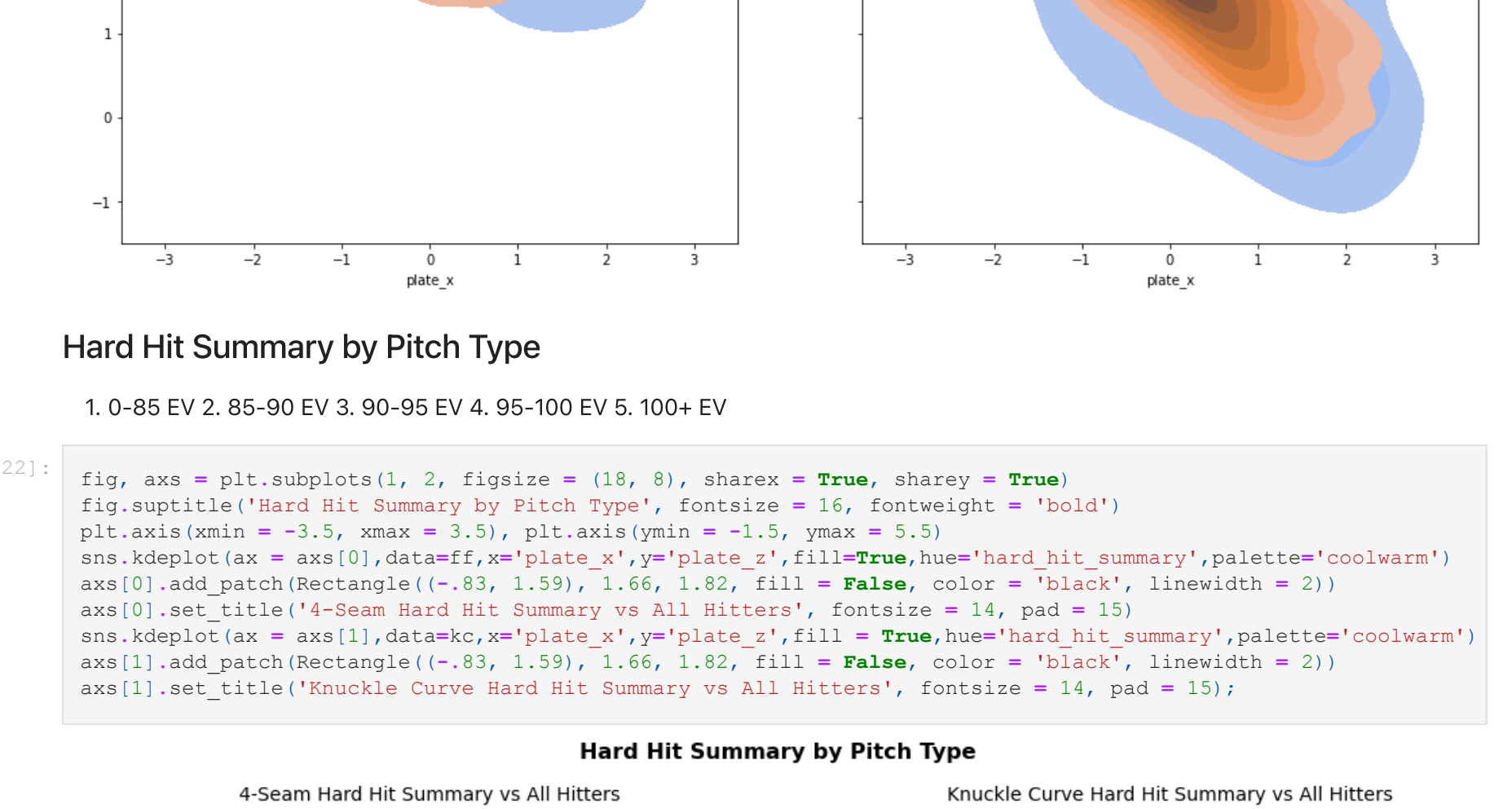


Heatmaps

*All From Hitters' Perspective

Pitch Location by Pitch Type

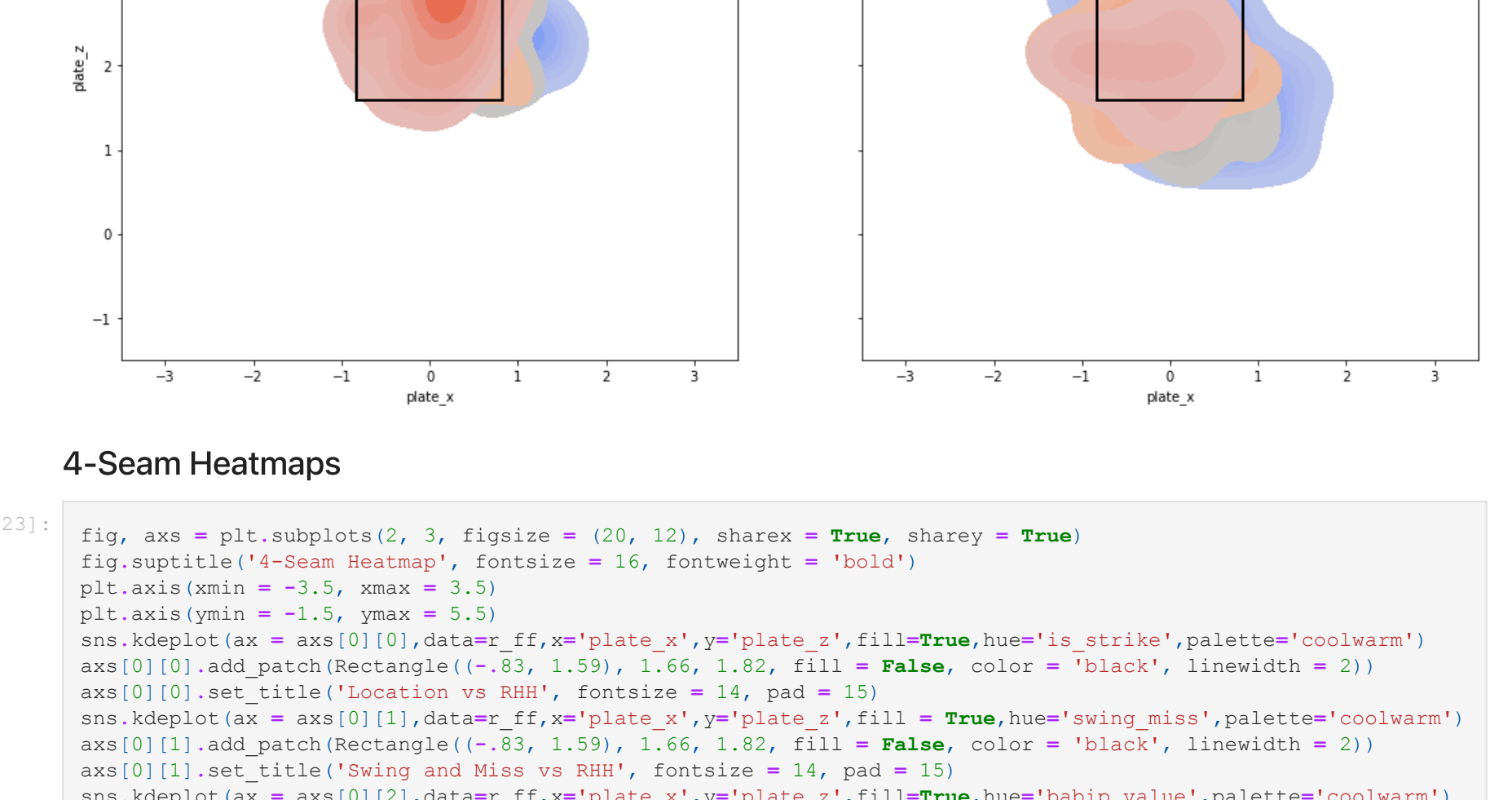
In [21]:
```
fig, axs = plt.subplots(1, 2, figsize = (18, 8), sharex = True, sharey = True)
fig.suptitle('Pitch Location by Pitch Type', fontsize = 16, fontweight = 'bold')
plt.axis(xmin = -3.5, xmax = 3.5)
plt.axis(ymin = -1.5, ymax = 5.5)
sns.kdeplot(ax = axs[0], data=ff,x='plate_x',y='plate_z',fill=True,hue='is_strike',palette='coolwarm')
axs[0].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0].set_title('4-Seam Location vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1], data=kc,x='plate_x',y='plate_z',fill = True,hue='is_strike',palette='coolwarm')
axs[1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1].set_title('Knuckle Curve Location vs All Hitters', fontsize = 14, pad = 15);
```



Hard Hit Summary by Pitch Type

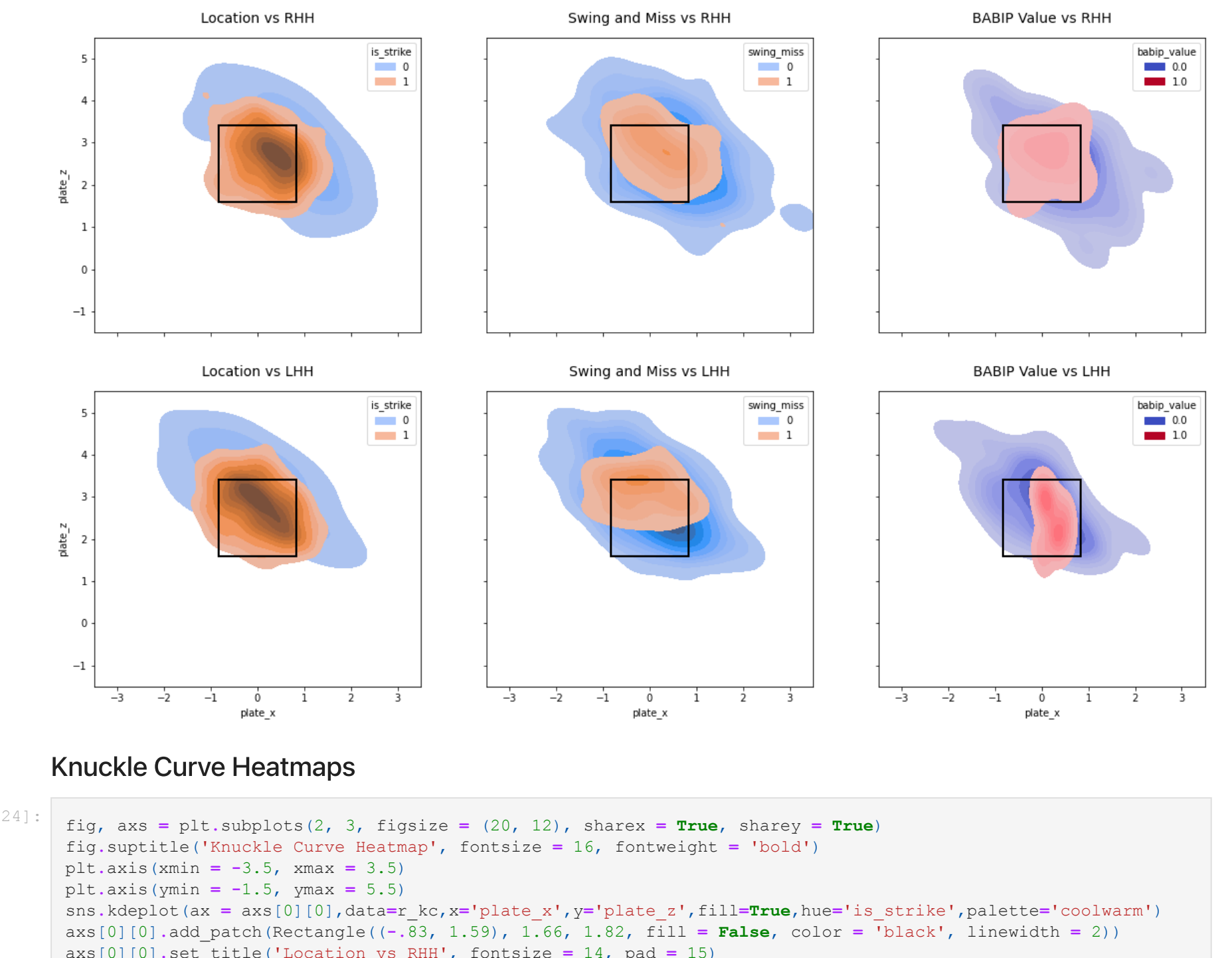
1. 0-85 EV 2. 85-90 EV 3. 90-95 EV 4. 95-100 EV 5. 100+ EV

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.axis(xmin = -3.5, xmax = 3.5), plt.axis(ymin = -1.5, ymax = 5.5)
sns.kdeplot(ax = axs[0], data=ff,x='plate_x',y='plate_z',fill=True,hue='hard_hit_summary',palette='coolwarm')
axs[0].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0].set_title('4-Seam Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1], data=kc,x='plate_x',y='plate_z',fill = True,hue='hard_hit_summary',palette='coolwarm')
axs[1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[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.axis(xmin = -3.5, xmax = 3.5)
plt.axis(ymin = -1.5, ymax = 5.5)
sns.kdeplot(ax = axs[0][0], data=r_ff,x='plate_x',y='plate_z',fill=True,hue='is_strike',palette='coolwarm')
axs[0][0].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0][0].set_title('Location vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[0][1], data=r_ff,x='plate_x',y='plate_z',fill = True,hue='swing_miss',palette='coolwarm')
axs[0][1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0][1].set_title('Swing and Miss vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[0][2], data=r_ff,x='plate_x',y='plate_z',fill=True,hue='babip_value',palette='coolwarm')
axs[0][2].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0][2].set_title('BABIP Value vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][0], data=l_ff,x='plate_x',y='plate_z',fill=True,hue='is_strike',palette='coolwarm')
axs[1][0].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1][0].set_title('Location vs LHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][1], data=l_ff,x='plate_x',y='plate_z',fill=True,hue='swing_miss',palette='coolwarm')
axs[1][1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1][1].set_title('Swing and Miss vs LHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][2], data=l_ff,x='plate_x',y='plate_z',fill=True,hue='babip_value',palette='coolwarm')
axs[1][2].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1][2].set_title('BABIP Value vs LHH', fontsize = 14, 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.axis(xmin = -3.5, xmax = 3.5)
plt.axis(ymin = -1.5, ymax = 5.5)
sns.kdeplot(ax = axs[0][0], data=r_kc,x='plate_x',y='plate_z',fill=True,hue='is_strike',palette='coolwarm')
axs[0][0].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0][0].set_title('Location vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[0][1], data=r_kc,x='plate_x',y='plate_z',fill=True,hue='swing_miss',palette='coolwarm')
axs[0][1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0][1].set_title('Swing and Miss vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[0][2], data=r_kc,x='plate_x',y='plate_z',fill=True,hue='woba_value',palette='coolwarm')
axs[0][2].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[0][2].set_title('wOBA Value vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][0], data=l_kc,x='plate_x',y='plate_z',fill=True,hue='is_strike',palette='coolwarm')
axs[1][0].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1][0].set_title('Location vs LHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][1], data=l_kc,x='plate_x',y='plate_z',fill=True,hue='swing_miss',palette='coolwarm')
axs[1][1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1][1].set_title('Swing and Miss vs LHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][2], data=l_kc,x='plate_x',y='plate_z',fill=True,hue='woba_value',palette='coolwarm')
axs[1][2].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1][2].set_title('wOBA Value vs LHH', fontsize = 14, pad = 15);
```