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]:

```
doolittle = pd.read_csv('../data/sean-doolittle.csv')
doolittle.drop(columns = ['Unnamed: 0'], inplace = True)
doolittle.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(doolittle.shape)
doolittle.head(2)
```

(866, 84)

Out[2]:

	pitch_type	game_date	release_speed	release_pos_x	release_pos_z	player_name	batter	pitcher	events	description	zone
0	FF	2021-07-27	94.7	2.07	5.79	Doolittle, Sean	519203	448281	out	hit_into_play	14
1	FF	2021-07-27	95.3	2.09	5.83	Doolittle, Sean	592178	448281	walk	ball	12

In [3]:

```
gen_data = doolittle[['pitch_type', 'release_speed', 'release_spin_rate',
                      'true_spin', 'spin_eff', 'spin_axis', 'pfx_x', 'pfx_z', 'vb',
                      '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_side', 'release_pos_z': 'r_height'
}
gen_data.rename(columns = col_dict, inplace = True)

hit_labels = [1, 2, 3, 4, 5]
doolittle['hard_hit_summary'] = pd.qcut(doolittle['launch_speed'], [0, .5262, .617, .7283, .8278, 1],
                                        labels = hit_labels)
```

In [4]:

```
# doolittle.pitch_type.value_counts(normalize=True)
r_doolittle = doolittle.loc[doolittle['stand'] == 'R']
l_doolittle = doolittle.loc[doolittle['stand'] == 'L']
# all hitters
ff = doolittle.loc[doolittle['pitch_type'] == 'FF']
cu = doolittle.loc[doolittle['pitch_type'] == 'CU']
sl = doolittle.loc[doolittle['pitch_type'] == 'SL']
fs = doolittle.loc[doolittle['pitch_type'] == 'FS']
# RHH
r_ff = r_doolittle.loc[r_doolittle['pitch_type'] == 'FF']
r_cu = r_doolittle.loc[r_doolittle['pitch_type'] == 'CU']
r_sl = r_doolittle.loc[r_doolittle['pitch_type'] == 'SL']
r_fs = r_doolittle.loc[r_doolittle['pitch_type'] == 'FS']
# LHH
l_ff = l_doolittle.loc[l_doolittle['pitch_type'] == 'FF']
l_cu = l_doolittle.loc[l_doolittle['pitch_type'] == 'CU']
l_sl = l_doolittle.loc[l_doolittle['pitch_type'] == 'SL']
l_fs = l_doolittle.loc[l_doolittle['pitch_type'] == 'FS']
order = ['FF', 'CU', 'SL', 'FS']

ff_tilt = ff['phi'].mean()
cu_tilt = cu['phi'].mean()
sl_tilt = sl['phi'].mean()
fs_tilt = fs['phi'].mean()
```

Pitcher Overview

General Pitch Data

In [5]:

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

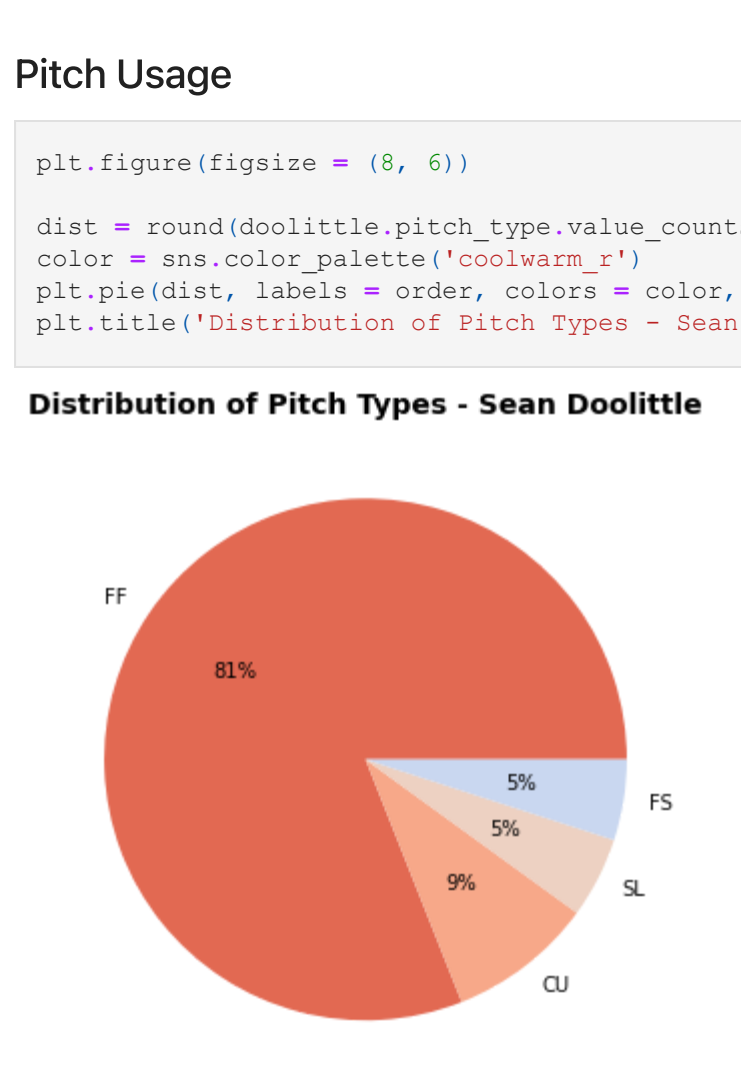
Out[5]:

pitch_type	velo	spin	true_spin	spin_eff	spin_axis	hb	vb	strike	r_side	r_height	bauer_
FF	93.082454	2260.757489	1836.341299	0.810455	165.846154	20.023248	-3.623077	0.703704	1.757863	6.082806	24.28
SL	84.180851	2260.042553	319.597273	0.143030	218.484848	5.956596	2.872340	0.574468	2.021915	6.078085	26.86
CU	79.728205	2405.307692	356.355789	0.149474	257.315789	0.967692	7.756923	0.551282	2.093462	5.903590	30.10
FS	80.420513	982.230769	669.579310	0.702414	149.968750	9.603077	-6.750769	0.461538	1.972821	5.942821	12.19

Pitch Usage

In [6]:

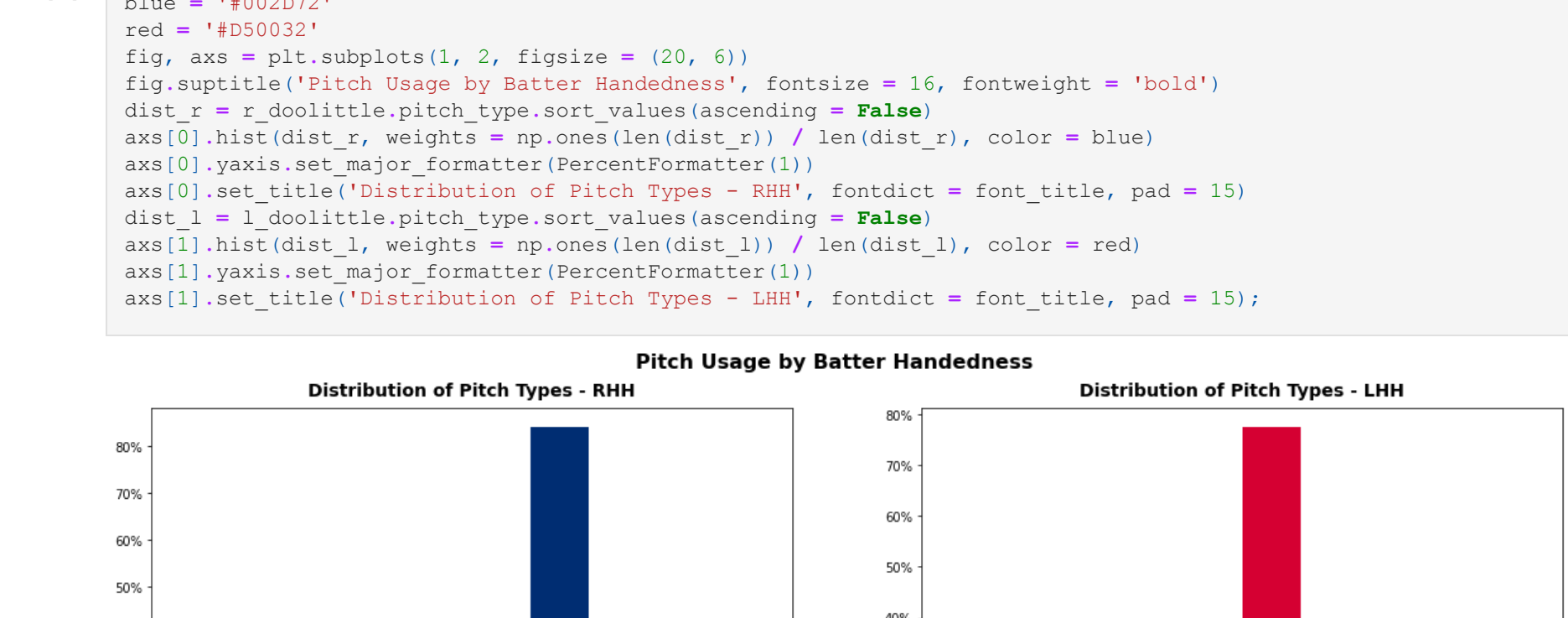
```
plt.figure(figsize = (8, 6))

dist = round(doolittle.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 - Sean Doolittle', fontdict = font_title, pad = 15);
```



Pitch Usage by Batter Handedness

In [7]:

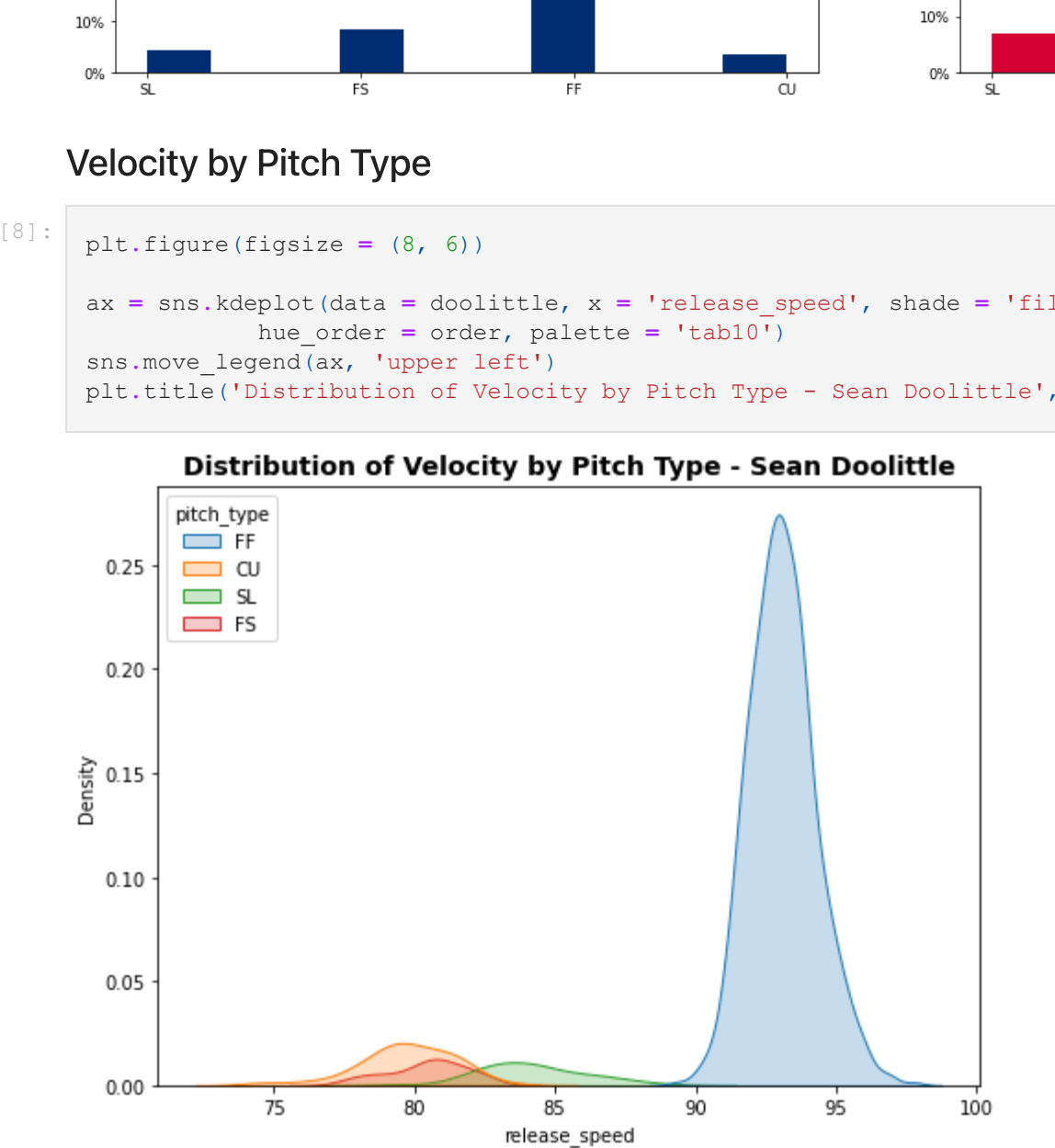
```
blue = '#002D72'
red = '#D50032'
fig, axs = plt.subplots(1, 2, figsize = (20, 6))
fig.suptitle('Pitch Usage by Batter Handedness', fontsize = 16, fontweight = 'bold')
dist_r = r_doolittle.pitch_type.sort_values(ascending = False)
axs[0].hist(dist_r, weights = np.ones(len(dist_r)) / len(dist_r), color = 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_doolittle.pitch_type.sort_values(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 [8]:

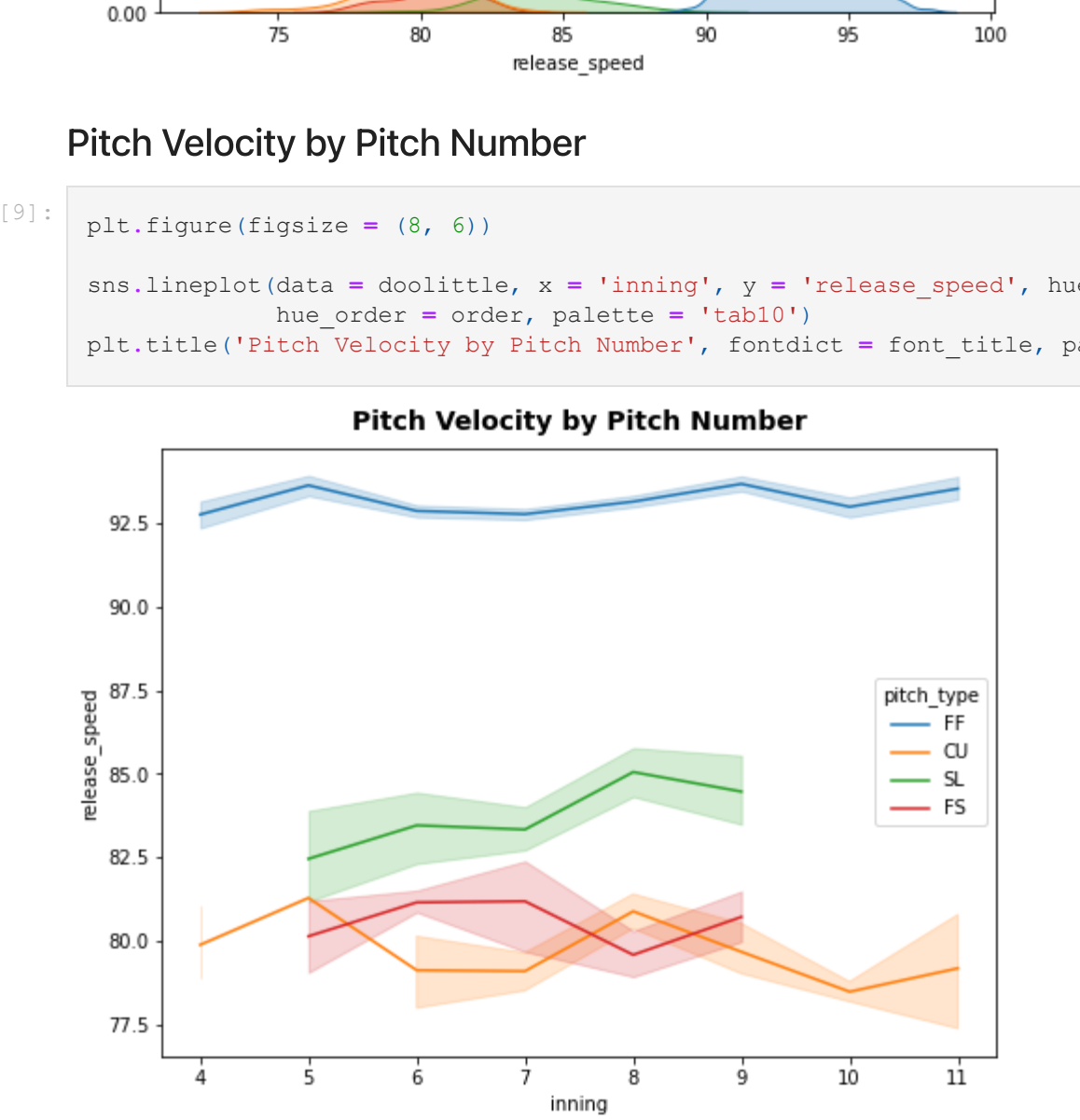
```
plt.figure(figsize = (8, 6))

ax = sns.kdeplot(data = doolittle, x = 'release_speed', shade = 'fill', hue = 'pitch_type',
                 hue_order = order, palette = 'tab10')
sns.move_legend(ax, 'upper left')
plt.title('Distribution of Velocity by Pitch Type - Sean Doolittle', fontdict = font_title, pad = 12);
```



Pitch Velocity by Pitch Number

In [9]:

```
plt.figure(figsize = (8, 6))

sns.lineplot(data = doolittle, 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);
```

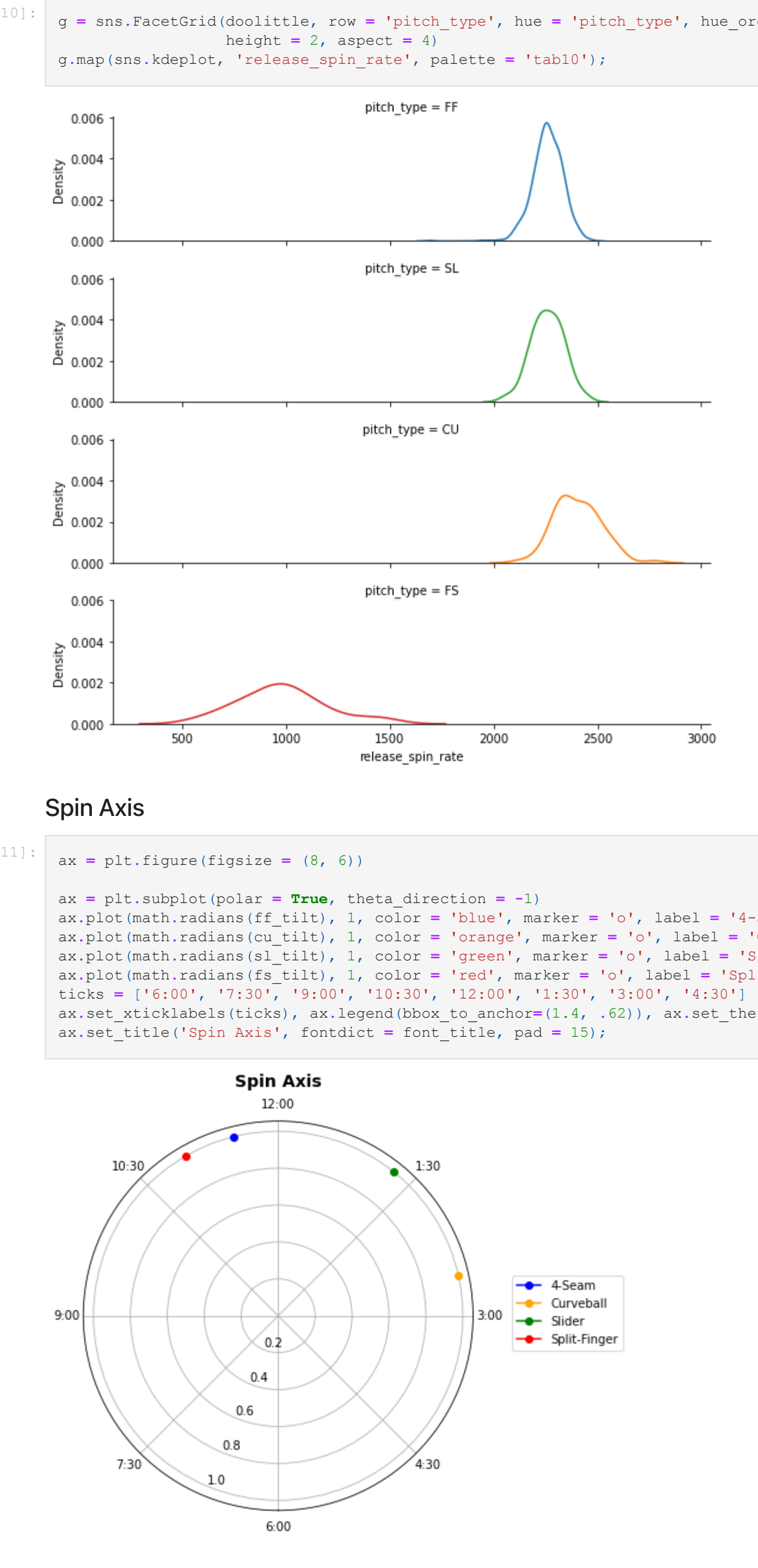


Pitcher Stuff

Spin Rate by Pitch Type

In [10]:

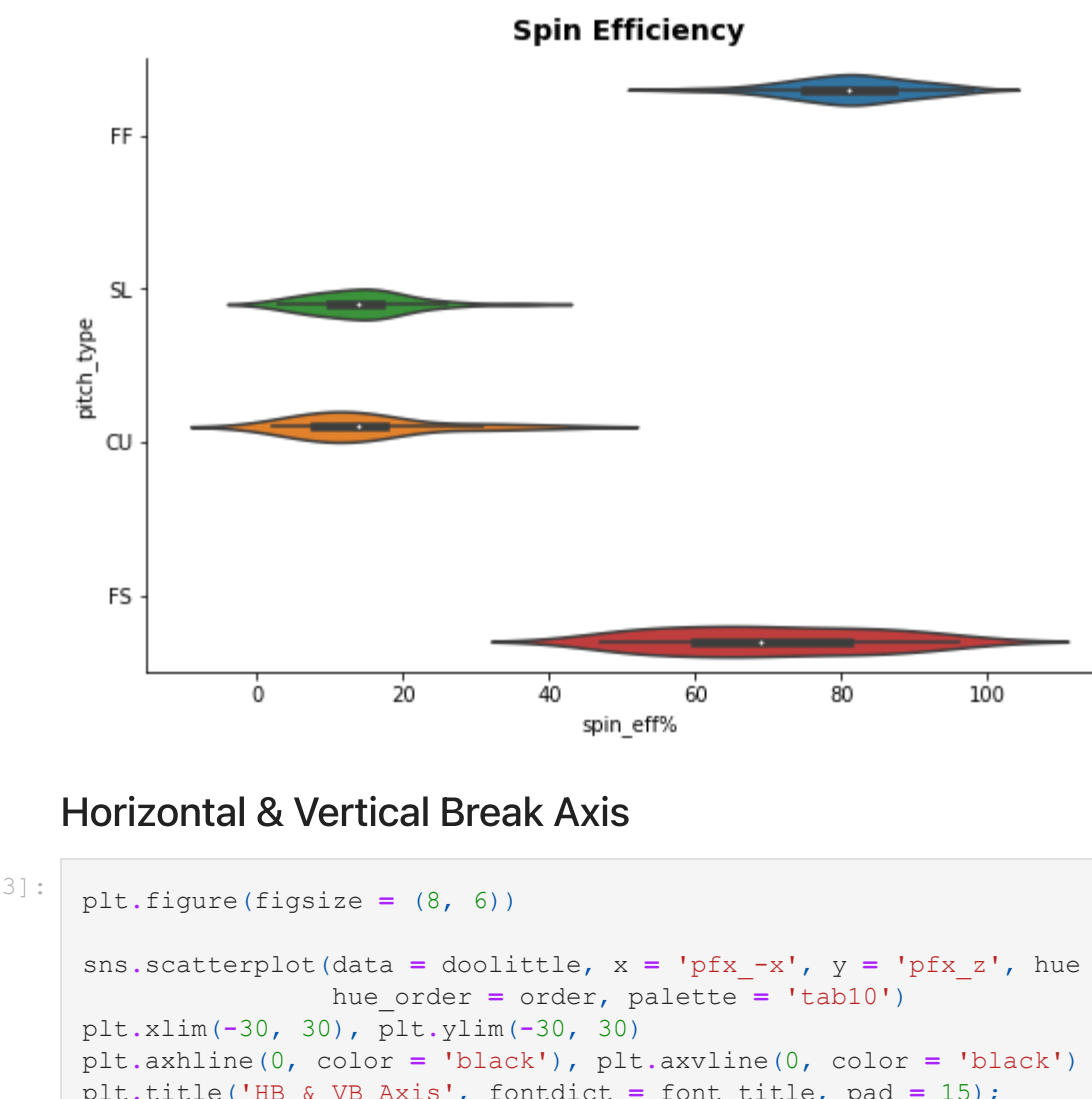
```
g = sns.FacetGrid(doolittle, row = 'pitch_type', hue = 'pitch_type', hue_order = order, palette = 'tab10',
                  height = 2, aspect = 4)
g.map(sns.kdeplot, 'release_spin_rate', palette = 'tab10');
```



Spin Axis

In [11]:

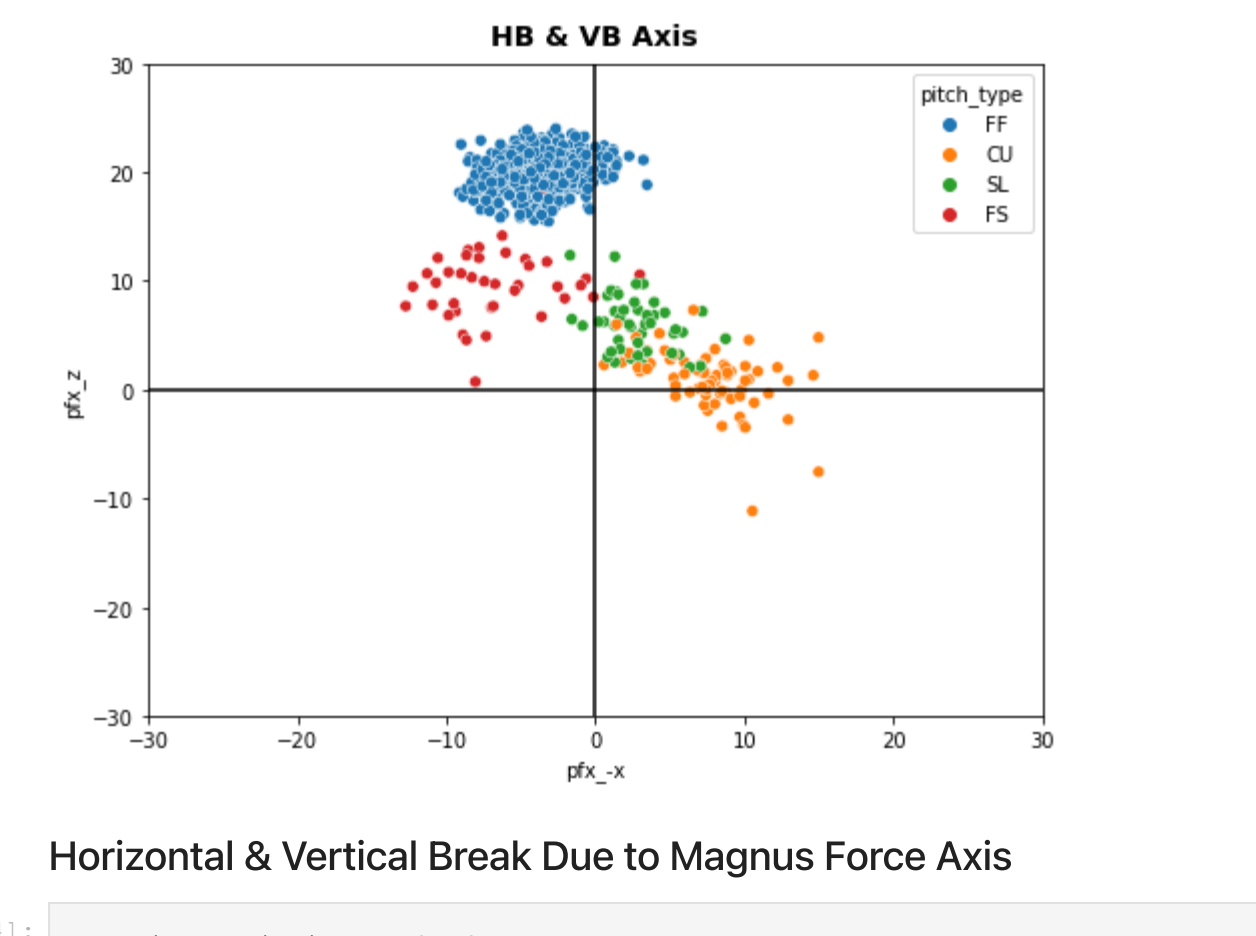
```
ax = plt.subplot(polar = True, theta_direction = -1)
ax.plot(math.radians(ff_tilt), 1, color = 'blue', marker = 'o', label = '4-Seam')
ax.plot(math.radians(cu_tilt), 1, color = 'orange', marker = 'o', label = 'Curveball')
ax.plot(math.radians(sl_tilt), 1, color = 'green', marker = 'o', label = 'Slider')
ax.plot(math.radians(fs_tilt), 1, color = 'red', marker = 'o', label = 'Split-Finger')
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=(1.4, .62)), ax.set_theta_zero_location("S")
ax.set_title('Spin Axis', fontdict = font_title, pad = 15);
```



Spin Efficiency

In [12]:

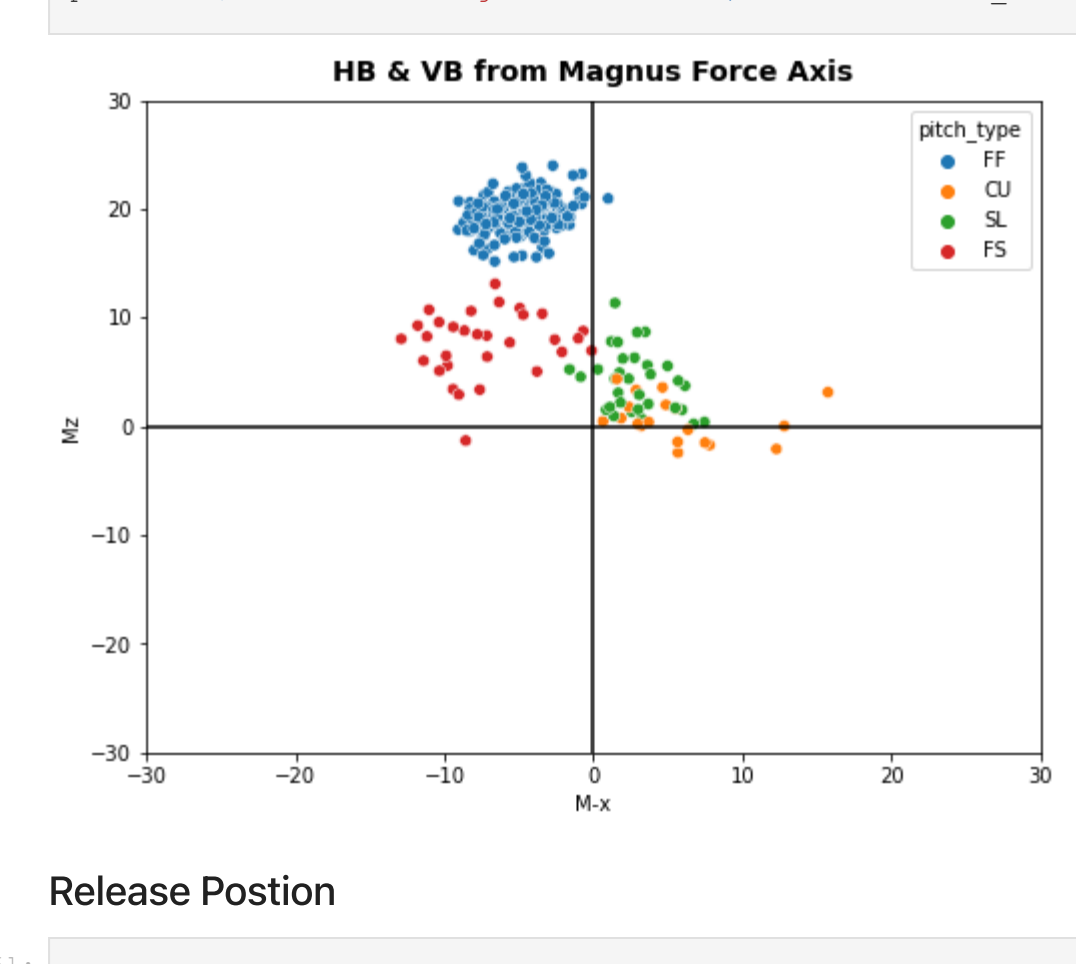
```
sns.catplot(data = doolittle, x = 'spin_eff', y = 'pitch_type', kind = 'violin', hue = 'pitch_type',
            hue_order = order, scale = 1.5, palette = 'tab10')
plt.title('Spin Efficiency', fontdict = font_title, pad = 15);
```



Horizontal & Vertical Break Axis

In [13]:

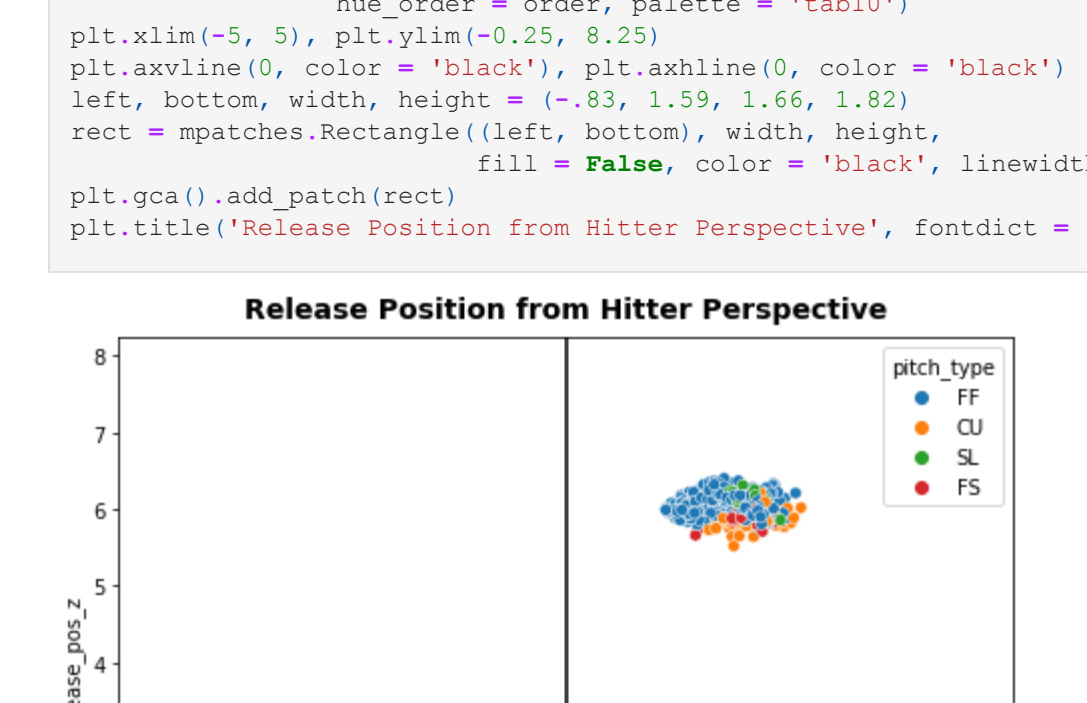
```
plt.figure(figsize = (8, 6))

sns.scatterplot(data = doolittle, 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 [14]:

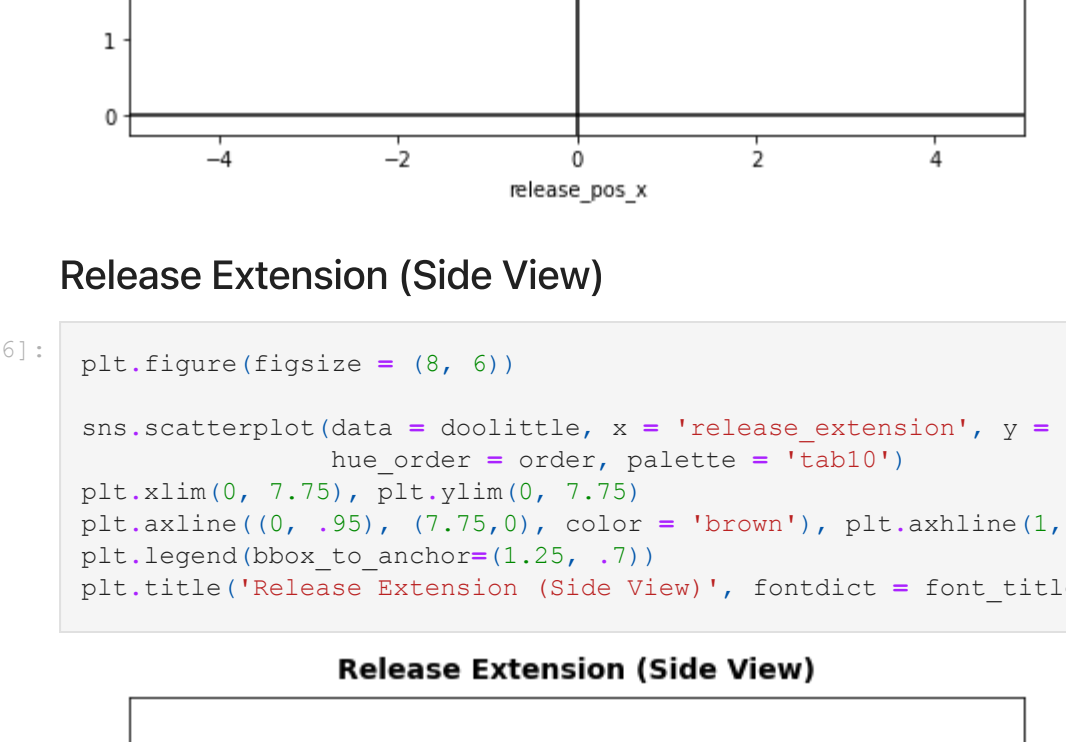
```
plt.figure(figsize = (8, 6))

sns.scatterplot(data = doolittle, 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 [15]:

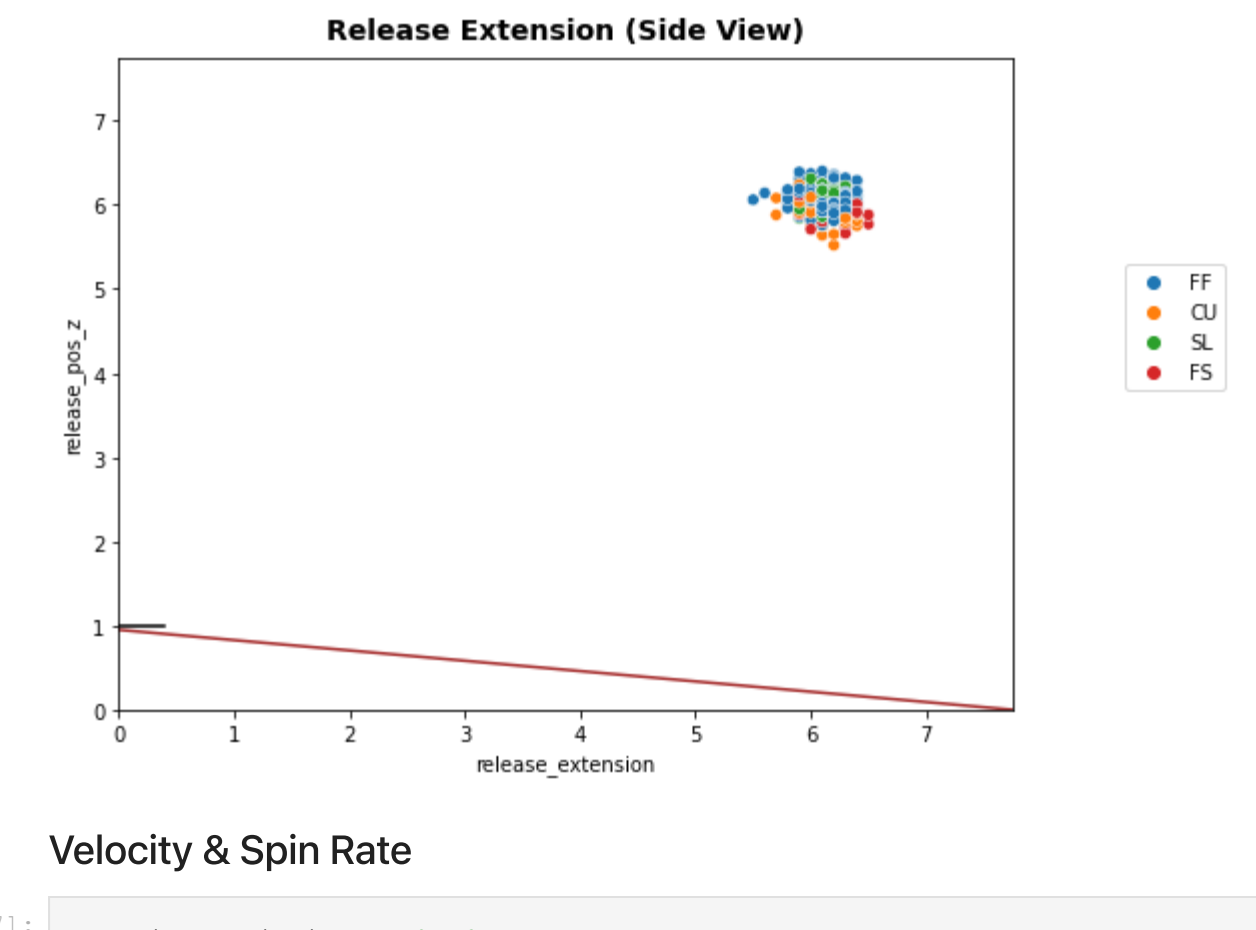
```
plt.figure(figsize = (8, 6))

sns.scatterplot(data = doolittle, 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.axhline(0, color = 'black')
left, bottom, width, height = (-.83, 1.59, 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 [16]:

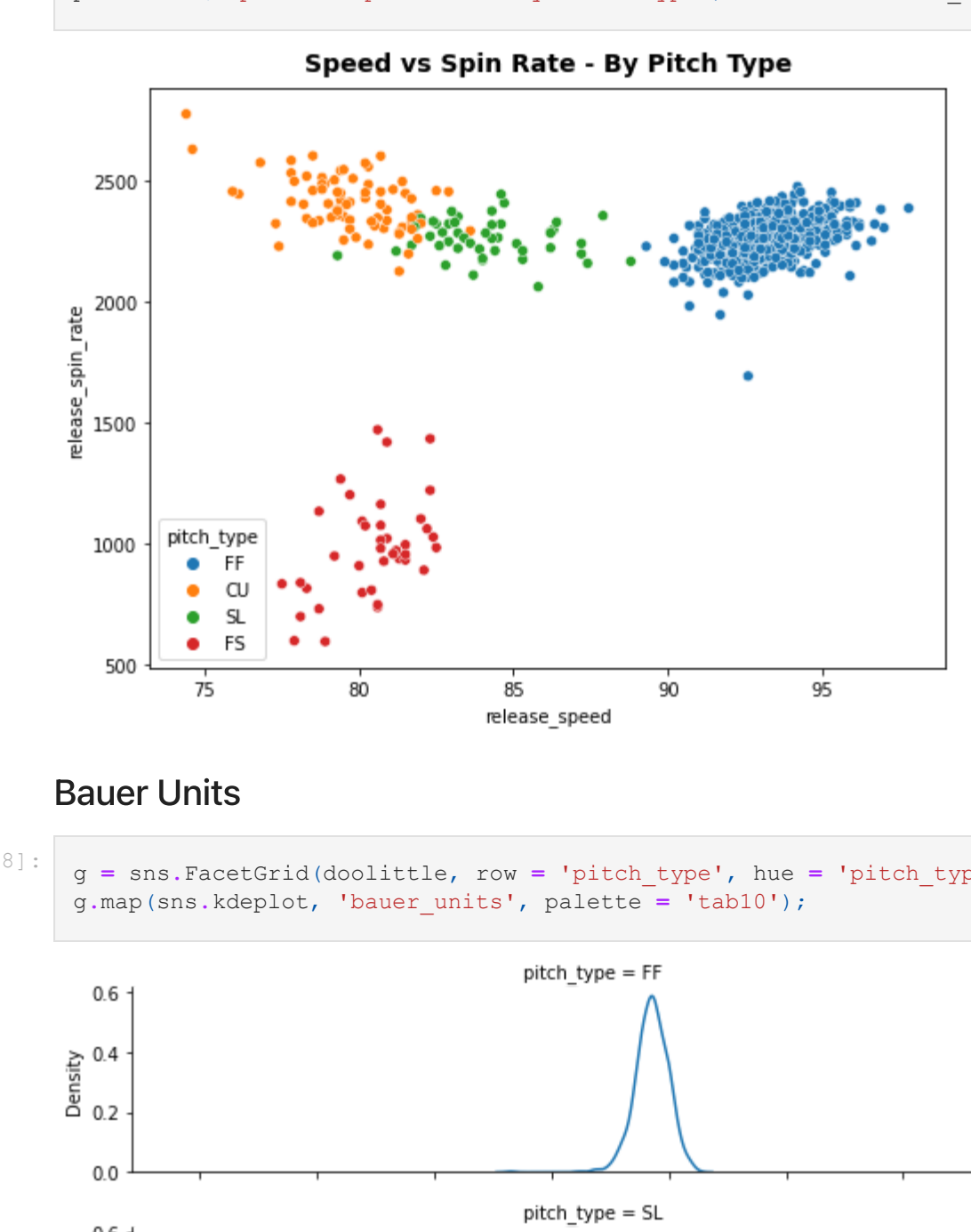
```
plt.figure(figsize = (8, 6))

sns.scatterplot(data = doolittle, 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.axline((0, .95), (7.75,0), color = 'brown'), plt.axhline(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 [17]:

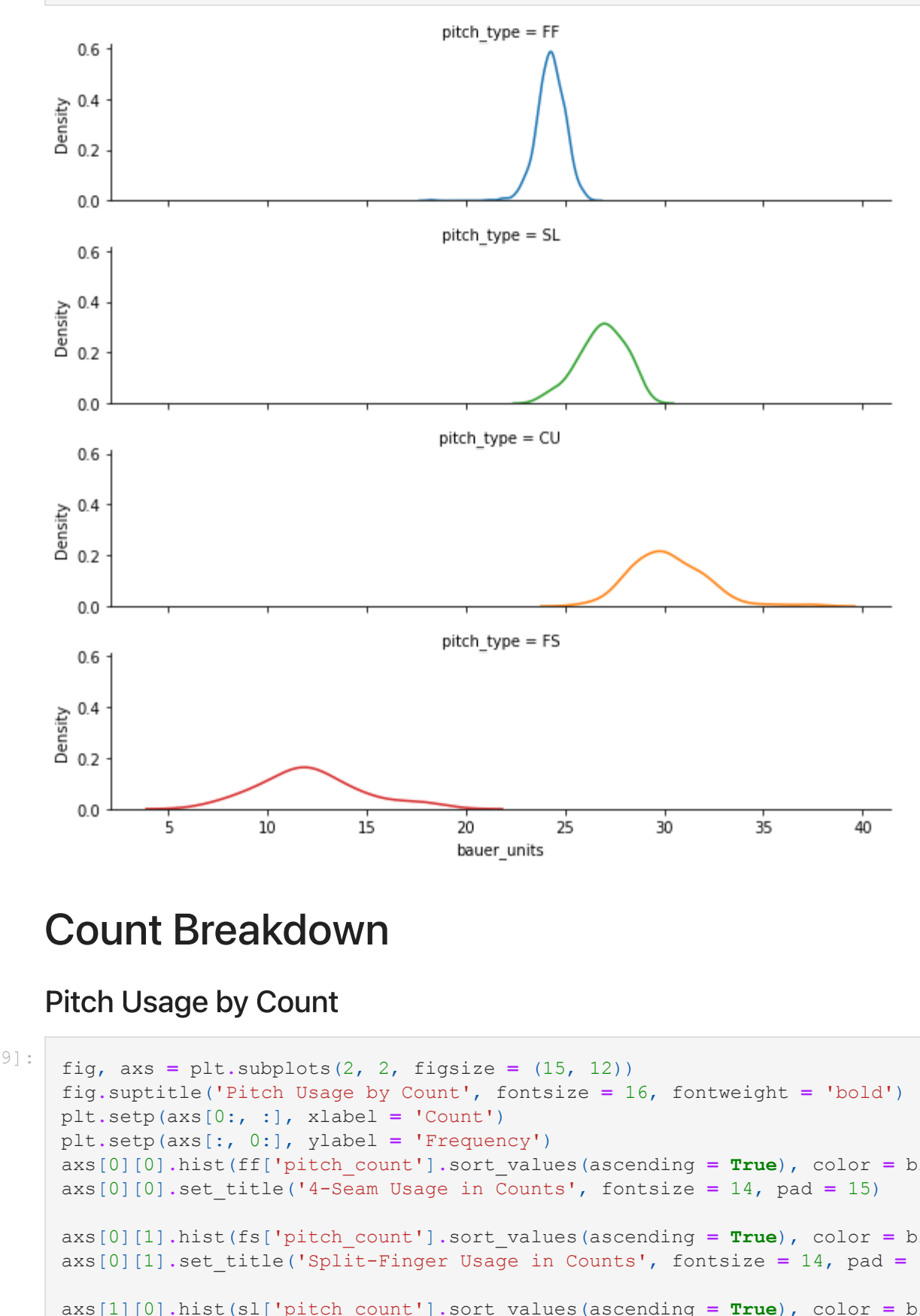
```
plt.figure(figsize = (8, 6))

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



Bauer Units

In [18]:

```
g = sns.FacetGrid(doolittle, row = 'pitch_type', hue = 'pitch_type', height = 2, aspect = 4, hue_order = order)
g.map(sns.kdeplot, 'bauer_units', palette = 'tab10');
```



Count Breakdown

Pitch Usage by Count

In [19]:

```
fig, axs = plt.subplots(2, 2, figsize = (15, 12))
fig.suptitle('Pitch Usage by Count', fontsize = 16, fontweight = 'bold')
plt.setp(axs[0:, :], xlabel = 'Count')
plt.setp(axs[:, 0:], ylabel = 'Frequency')
axs[0][0].hist(ff['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
axs[0][0].set_title('4-Seam Usage in Counts', fontsize = 14, pad = 15)

axs[0][1].hist(fs['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
axs[0][1].set_title('Split-Finger Usage in Counts', fontsize = 14, pad = 15)

axs[1][0].hist(sl['pitch_count'].sort_values(ascending = True), color = blue, edgecolor = red, linewidth = 2)
axs[1][0].set_title('Slider Usage in Counts', fontsize = 14, pad = 15)
plt.tight_layout();
```

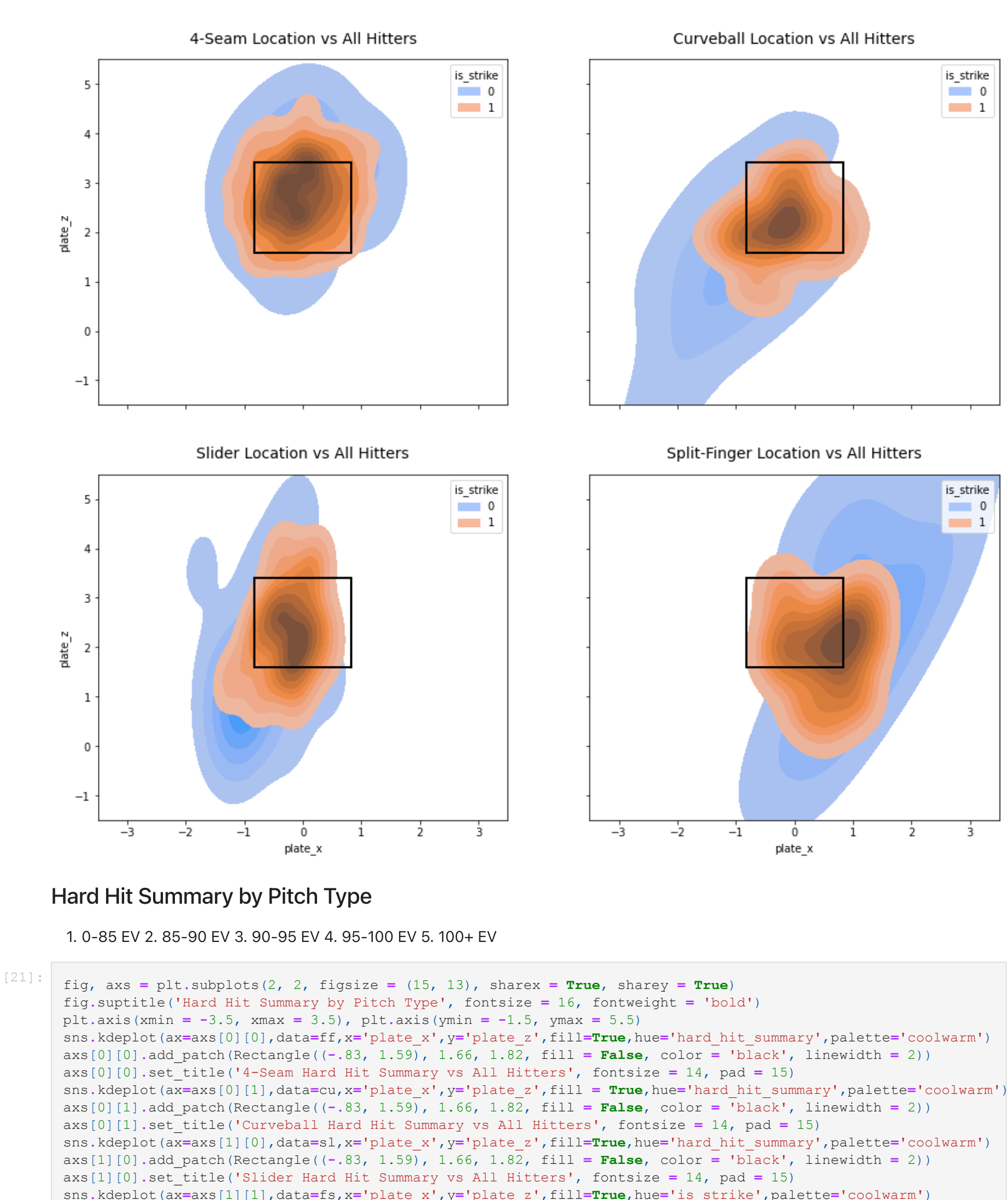
Heatmaps

*All From Hitters' Perspective

Pitch Location by Pitch Type

In [20]:

```
fig, axs = plt.subplots(2, 2, figsize = (15, 13), 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][0],data=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('4-Seam Location vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[0][1],data=cu,x='plate_x',y='plate_z',fill = True,hue='is_strike',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('Curveball Location vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][0],data=sl,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('Slider Location vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][1],data=fs,x='plate_x',y='plate_z',fill = True,hue='is_strike',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('Split-Finger Location vs All Hitters', fontsize = 14, pad = 15);
```

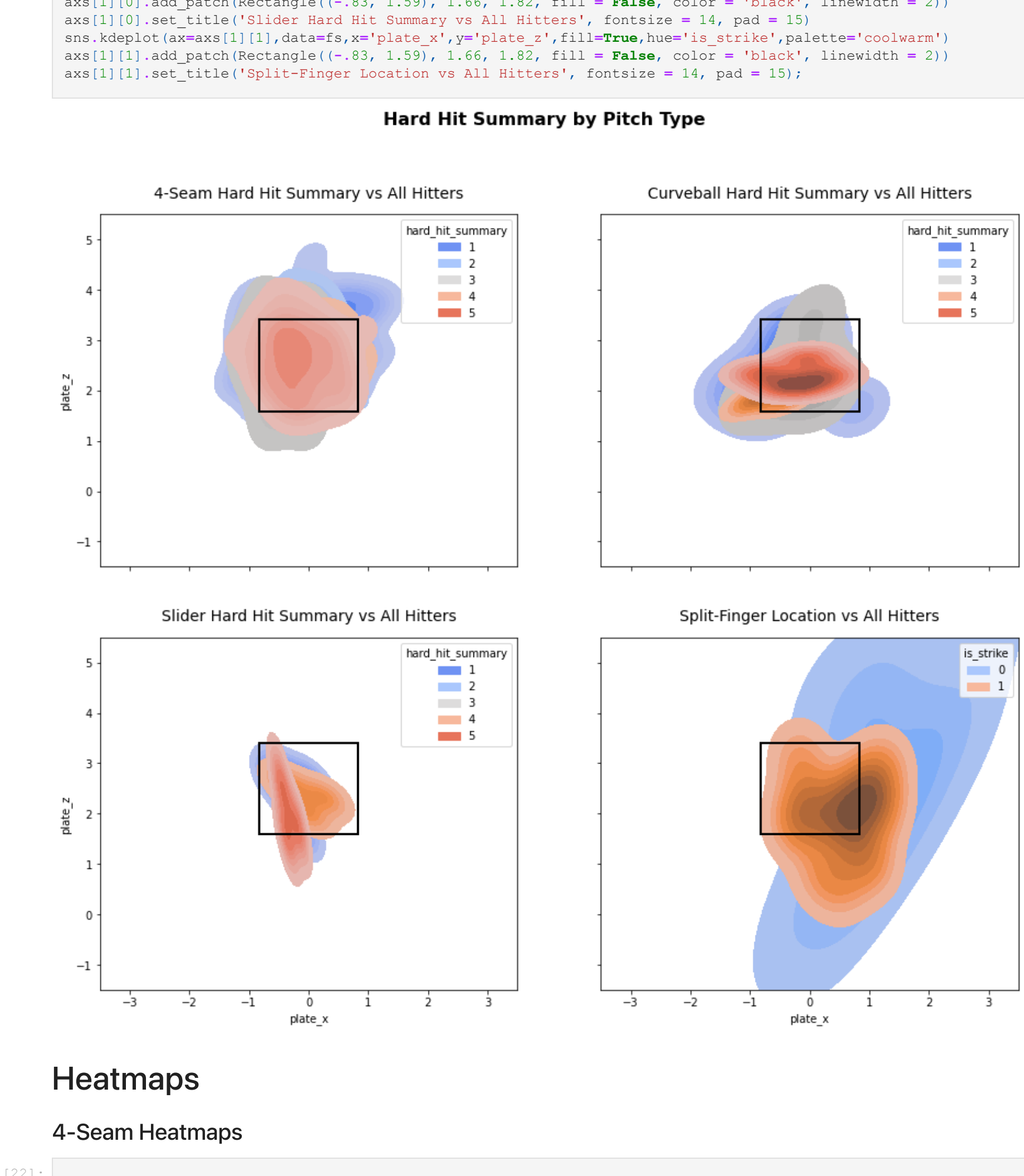


Hard Hit Summary by Pitch Type

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

In [21]:

```
fig, axs = plt.subplots(2, 2, figsize = (15, 13), 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][0],data=ff,x='plate_x',y='plate_z',fill=True,hue='hard_hit_summary',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('4-Seam Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax=axs[0][1],data=cu,x='plate_x',y='plate_z',fill = True,hue='hard_hit_summary',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('Curveball Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax=axs[1][0],data=sl,x='plate_x',y='plate_z',fill=True,hue='hard_hit_summary',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('Slider Hard Hit Summary vs All Hitters', fontsize = 14, pad = 15)
sns.kdeplot(ax=axs[1][1],data=fs,x='plate_x',y='plate_z',fill=True,hue='is_strike',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('Split-Finger Location vs All Hitters', fontsize = 14, pad = 15);
```

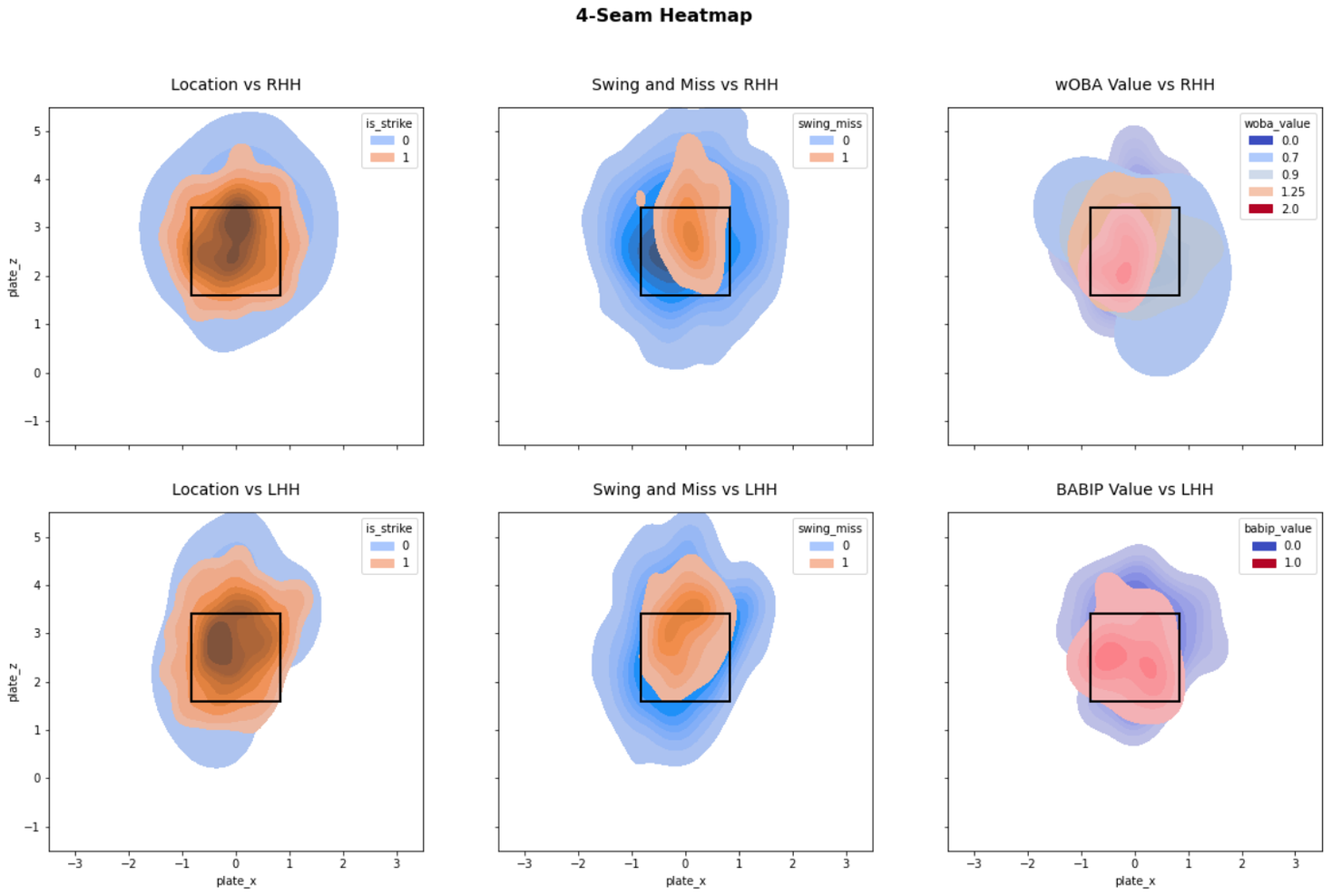


Heatmaps

4-Seam Heatmaps

In [22]:

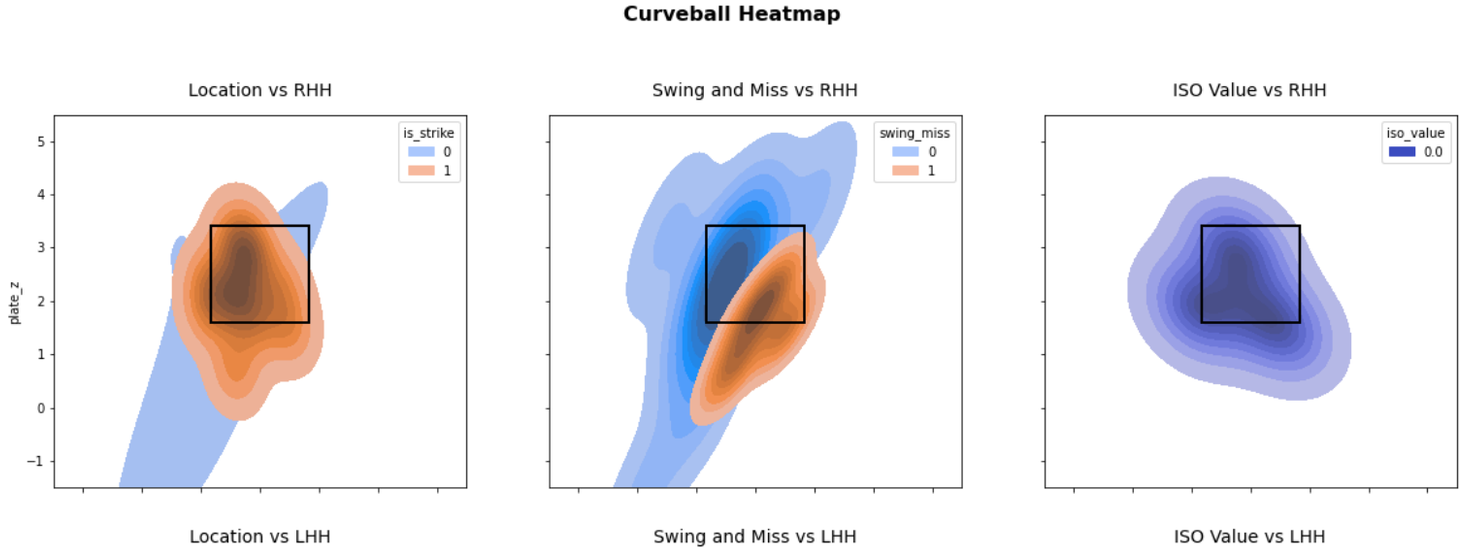
```
fig, axs = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('4-Seam Heatmap', fontsize = 16, fontweight = 'bold')
plt.axis(xmin = -3.5, xmax = 3.5), plt.axis(ymin = -1.5, ymax = 5.5)
sns.kdeplot(ax = axs[0][0],data=r_ff,x='plate_x',y='plate_z',fill=True,hue='is_strike',palette='coolwarm')
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='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_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);
```

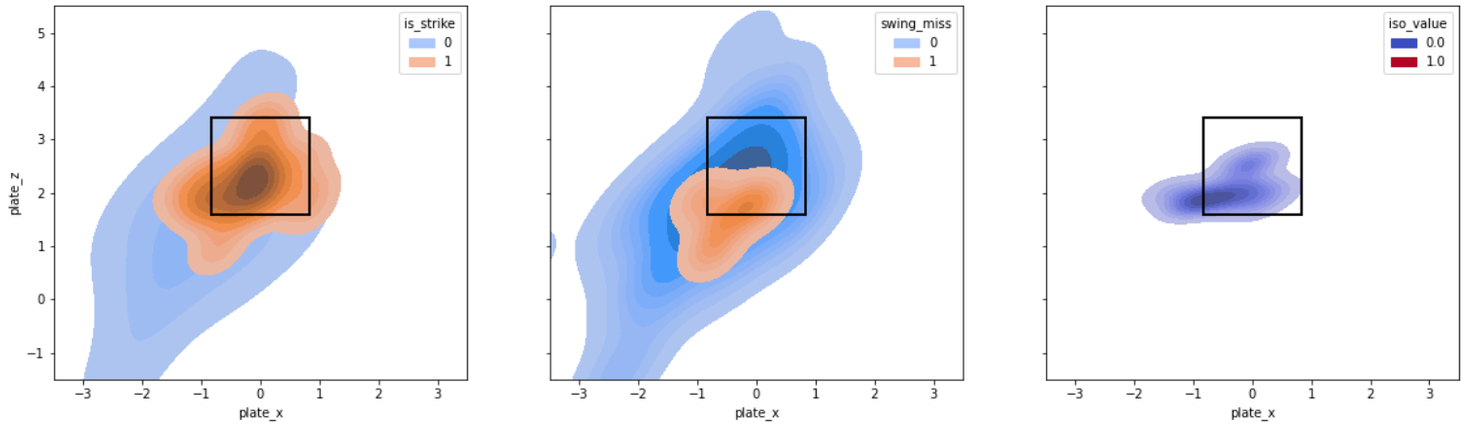


Curveball Heatmaps

In [23]:

```
fig, axs = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('Curveball 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_cu,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_cu,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_cu,x='plate_x',y='plate_z',fill=True,hue='iso_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('ISO Value vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][0],data=l_cu,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_cu,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_cu,x='plate_x',y='plate_z',fill=True,hue='iso_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('ISO Value vs LHH', fontsize = 14, pad = 15);
```

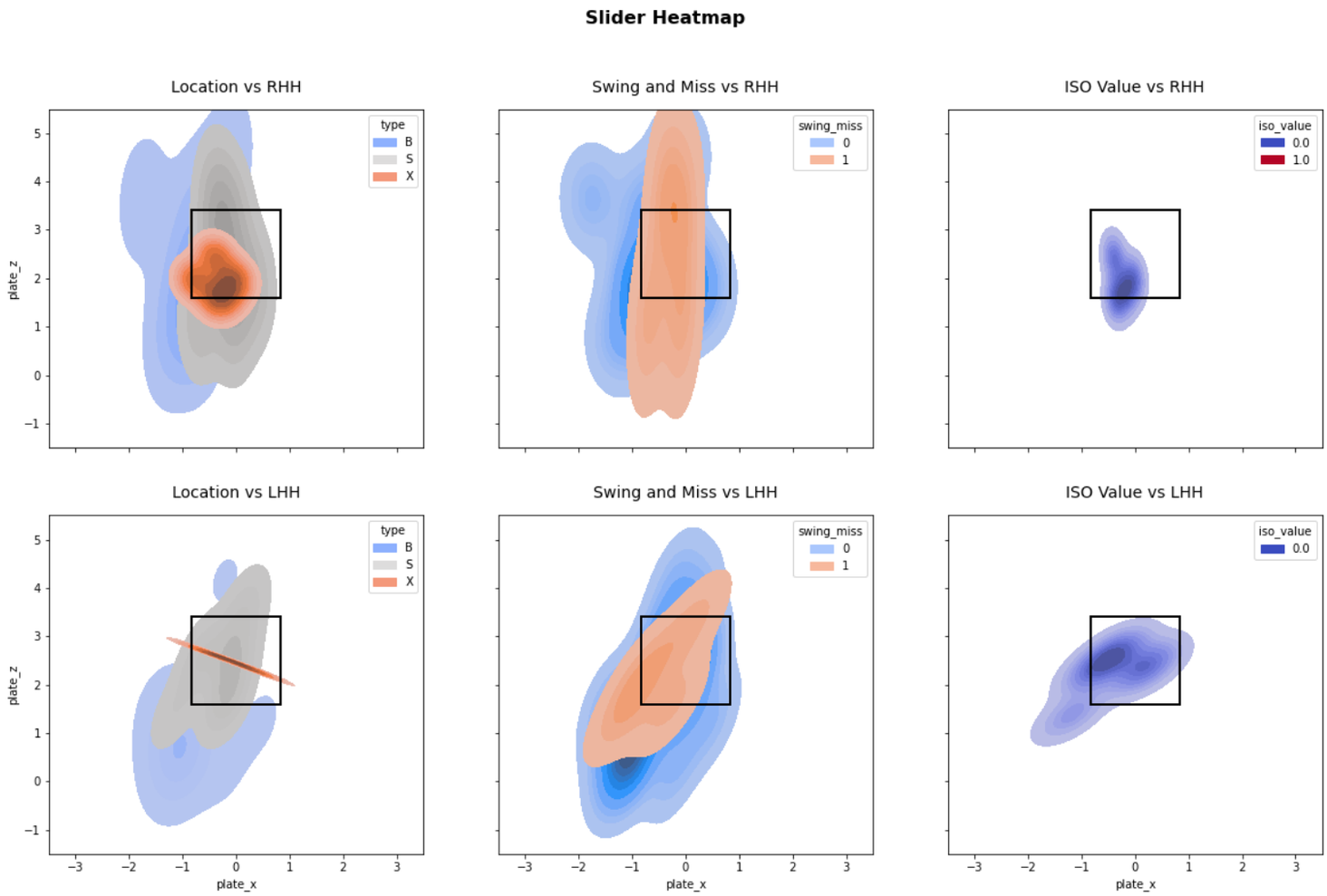




Slider Heatmaps

In [24]:

```
fig, axs = plt.subplots(2, 3, figsize = (20, 12), sharex = True, sharey = True)
fig.suptitle('Slider 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_sl,x='plate_x',y='plate_z',fill=True,hue='type',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_sl,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_sl,x='plate_x',y='plate_z',fill=True,hue='iso_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('ISO Value vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1][0],data=l_sl,x='plate_x',y='plate_z',fill=True,hue='type',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_sl,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_sl,x='plate_x',y='plate_z',fill=True,hue='iso_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('ISO Value vs LHH', fontsize = 14, pad = 15);
```



Split-Finger Heatmaps

In [25]:

```
fig, axs = plt.subplots(1, 3, figsize = (25, 8), sharex = True, sharey = True)
fig.suptitle('Split-Finger 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],data=r_fs,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('Location vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[1],data=r_fs,x='plate_x',y='plate_z',fill = True,hue='swing_miss',palette='coolwarm')
axs[1].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[1].set_title('Swing and Miss vs RHH', fontsize = 14, pad = 15)
sns.kdeplot(ax = axs[2],data=r_fs,x='plate_x',y='plate_z',fill=True,hue='iso_value',palette='coolwarm')
axs[2].add_patch(Rectangle((-.83, 1.59), 1.66, 1.82, fill = False, color = 'black', linewidth = 2))
axs[2].set_title('ISO Value vs RHH', fontsize = 14, pad = 15);
```