1. The Statcast revolution

This is Aaron Judge. Judge is one of the physically largest players in Major League Baseball standing 6 feet 7 inches (2.01 m) tall and weighing 282 pounds (128 kg). He also hit the [hardest home run]() ever recorded. How do we know this? **Statcast**.

Statcast is a state-of-the-art tracking system that uses high-resolution cameras and radar equipment to measure the precise location and movement of baseballs and baseball players. Introduced in 2015 to all 30 major league ballparks, Statcast data is revolutionizing the game. Teams are engaging in an "arms race" of data analysis, hiring analysts left and right in an attempt to gain an edge over their competition. This [video]() describing the system is incredible.

In this notebook, we're going to wrangle, analyze, and visualize Statcast data to compare Mr. Judge and another (extremely large) teammate of his. Let's start by loading the data into our Notebook. There are two CSV files, `judge.csv` and `stanton.csv`, both of which contain Statcast data for 2015-2017. We'll use pandas DataFrames to store this data. Let's also load our data visualization libraries, matplotlib and seaborn.

In [2]:

```
!pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefounda
Requirement already satisfied: pandas>=1.2 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.pyth
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefou
Requirement already satisfied: contourpy>=1.0.1 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation
Requirement already satisfied: cycler>=0.10 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.pyt
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundatic
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundatic
Requirement already satisfied: packaging>=20.0 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.
Requirement already satisfied: pillow>=8 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.pythor
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation
Requirement already satisfied: python-dateutil>=2.7 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefounda
Requirement already satisfied: pytz>=2020.1 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.pyt
Requirement already satisfied: tzdata>=2022.7 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.p
Requirement already satisfied: six>=1.5 in c:\users\lenovo\appdata\local\packages\pythonsoftwarefoundation.python.
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   - -------------------------------------- 10.2/294.9 kB ? eta -:--:--
   ---- ----------------------------------- 30.7/294.9 kB 435.7 kB/s eta 0:00:01
   ------------ --------------------------- 92.2/294.9 kB 751.6 kB/s eta 0:00:01
   ---------------------------------------- 294.9/294.9 kB 1.8 MB/s eta 0:00:00
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
```

```
[notice] A new release of pip is available: 24.0 -> 24.1
[notice] To update, run:
C:\Users\LENOVO\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe
-m pip install --upgrade pip
```

In [3]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Load Aaron Judge's Statcast data
judge = pd.read_csv('datasets/judge.csv')

# Load Giancarlo Stanton's Statcast data
stanton = pd.read_csv('datasets/stanton.csv')
```

2. What can Statcast measure?

The better question might be, what can't Statcast measure?

> Starting with the pitcher, Statcast can measure simple data points such as velocity. At the same time, Statcast digs a whole lot deeper, also measuring the release point and spin rate of every pitch.

Moving on to hitters, Statcast is capable of measuring the exit velocity, launch angle and vector of the ball as it comes off the bat. From there, Statcast can also track the hang time and projected distance that a ball travels.

Let's inspect the last five rows of the `judge` DataFrame. You'll see that each row represents one pitch thrown to a batter. You'll also see that some columns have esoteric names. If these don't make sense now, don't worry. The relevant ones will be explained as necessary.

In [4]:
```python
# Display all columns (pandas will collapse some columns if we don't set this option)
pd.set_option('display.max_columns', None)

# Display the last five rows of the Aaron Judge file

print(judge.tail())
```

```
     pitch_type   game_date  release_speed  release_pos_x  release_pos_z  \
3431         CH  2016-08-13           85.6        -1.9659         5.9113
3432         CH  2016-08-13           87.6        -1.9318         5.9349
3433         CH  2016-08-13           87.2        -2.0285         5.8656
3434         CU  2016-08-13           79.7        -1.7108         6.1926
3435         FF  2016-08-13           93.2        -1.8476         6.0063

      player_name  batter  pitcher    events          description  spin_dir  \
3431  Aaron Judge  592450   542882       NaN                 ball       NaN
3432  Aaron Judge  592450   542882  home_run  hit_into_play_score       NaN
3433  Aaron Judge  592450   542882       NaN                 ball       NaN
3434  Aaron Judge  592450   542882       NaN                 foul       NaN
3435  Aaron Judge  592450   542882       NaN        called_strike       NaN

      spin_rate_deprecated  break_angle_deprecated  break_length_deprecated  \
3431                   NaN                     NaN                      NaN
3432                   NaN                     NaN                      NaN
3433                   NaN                     NaN                      NaN
3434                   NaN                     NaN                      NaN
3435                   NaN                     NaN                      NaN

      zone                                                des game_type stand  \
3431  14.0                                                NaN         R     R
3432   4.0  Aaron Judge homers (1) on a fly ball to center...         R     R
3433  14.0                                                NaN         R     R
3434   4.0                                                NaN         R     R
3435   8.0                                                NaN         R     R

     p_throws home_team away_team type  hit_location   bb_type  balls  \
3431        R       NYY        TB    B           NaN       NaN      0
3432        R       NYY        TB    X           NaN  fly_ball      1
3433        R       NYY        TB    B           NaN       NaN      0
3434        R       NYY        TB    S           NaN       NaN      0
3435        R       NYY        TB    S           NaN       NaN      0

      strikes  game_year     pfx_x     pfx_z  plate_x  plate_z  on_3b  on_2b  \
3431        0       2016 -0.379108  0.370567    0.739    1.442    NaN    NaN
3432        2       2016 -0.295608  0.320400   -0.419    3.273    NaN    NaN
3433        2       2016 -0.668575  0.198567    0.561    0.960    NaN    NaN
3434        1       2016  0.397442 -0.614133   -0.803    2.742    NaN    NaN
3435        0       2016 -0.823050  1.623300   -0.273    2.471    NaN    NaN

      on_1b  outs_when_up  inning inning_topbot    hc_x   hc_y  \
3431    NaN             0       5           Bot     NaN    NaN
3432    NaN             2       2           Bot  130.45  14.58
3433    NaN             2       2           Bot     NaN    NaN
3434    NaN             2       2           Bot     NaN    NaN
3435    NaN             2       2           Bot     NaN    NaN

      tfs_deprecated  tfs_zulu_deprecated  pos2_person_id  umpire  \
3431             NaN                  NaN        571912.0     NaN
3432             NaN                  NaN        571912.0     NaN
3433             NaN                  NaN        571912.0     NaN
3434             NaN                  NaN        571912.0     NaN
3435             NaN                  NaN        571912.0     NaN

              sv_id    vx0      vy0    vz0     ax      ay      az  sz_top  \
3431  160813_144259  6.960 -124.371 -4.756 -2.821  23.634 -30.220    3.93
3432  160813_135833  4.287 -127.452 -0.882 -1.972  24.694 -30.705    4.01
3433  160813_135815  7.491 -126.665 -5.862 -6.393  21.952 -32.121    4.01
3434  160813_135752  1.254 -116.062  0.439  5.184  21.328 -39.866    4.01
3435  160813_135736  5.994 -135.497 -6.736 -9.360  26.782 -13.446    4.01

      sz_bot  hit_distance_sc  launch_speed  launch_angle  effective_speed  \
3431    1.82              NaN           NaN           NaN           84.459
3432    1.82            446.0         108.8        27.410           86.412
3433    1.82              NaN           NaN           NaN           86.368
3434    1.82              9.0          55.8       -24.973           77.723
3435    1.82              NaN           NaN           NaN           92.696

      release_spin_rate  release_extension  game_pk  pos1_person_id  \
3431             1552.0              5.683   448611        542882.0
3432             1947.0              5.691   448611        542882.0
3433             1761.0              5.721   448611        542882.0
```

```
3434            2640.0             5.022   448611        542882.0
3435            2271.0             6.068   448611        542882.0

      pos2_person_id.1  pos3_person_id  pos4_person_id  pos5_person_id  \
3431          571912.0        543543.0        523253.0        446334.0
3432          571912.0        543543.0        523253.0        446334.0
3433          571912.0        543543.0        523253.0        446334.0
3434          571912.0        543543.0        523253.0        446334.0
3435          571912.0        543543.0        523253.0        446334.0

      pos6_person_id  pos7_person_id  pos8_person_id  pos9_person_id  \
3431        622110.0        545338.0        595281.0        543484.0
3432        622110.0        545338.0        595281.0        543484.0
3433        622110.0        545338.0        595281.0        543484.0
3434        622110.0        545338.0        595281.0        543484.0
3435        622110.0        545338.0        595281.0        543484.0

      release_pos_y  estimated_ba_using_speedangle  \
3431        54.8144                           0.00
3432        54.8064                           0.98
3433        54.7770                           0.00
3434        55.4756                           0.00
3435        54.4299                           0.00

      estimated_woba_using_speedangle  woba_value  woba_denom  babip_value  \
3431                            0.000         NaN         NaN          NaN
3432                            1.937         2.0         1.0          0.0
3433                            0.000         NaN         NaN          NaN
3434                            0.000         NaN         NaN          NaN
3435                            0.000         NaN         NaN          NaN

      iso_value  launch_speed_angle  at_bat_number  pitch_number
3431        NaN                 NaN             36             1
3432        3.0                 6.0             14             4
3433        NaN                 NaN             14             3
3434        NaN                 1.0             14             2
3435        NaN                 NaN             14             1
```

3. Aaron Judge and Giancarlo Stanton, prolific sluggers

This is Giancarlo Stanton. He is also a very large human being, standing 6 feet 6 inches tall and weighing 245 pounds. Despite not wearing the same jersey as Judge in the pictures provided, in 2018 they will be teammates on the New York Yankees. They are similar in a lot of ways, one being that they hit a lot of home runs. Stanton and Judge led baseball in home runs in 2017, with 59 and 52, respectively. These are exceptional totals - the player in third "only" had 45 home runs.

Stanton and Judge are also different in many ways. One is batted ball events, which is any batted ball that produces a result. This includes outs, hits, and errors. Next, you'll find the counts of batted ball events for each player in 2017. The frequencies of other events are quite different.

In [5]:

```python
# All of Aaron Judge's batted ball events in 2017
judge_events_2017 = judge.loc[judge['game_year'] ==
print("Aaron Judge batted ball event totals, 2017:")
print(judge_events_2017.value_counts())

# All of Giancarlo Stanton's batted ball events in 2
stanton_events_2017 = stanton.loc[stanton['game_year
print("\nGiancarlo Stanton batted ball event totals,
print(stanton_events_2017.value_counts())
```

```
Aaron Judge batted ball event totals, 2017:
events
strikeout                    207
field_out                    146
walk                         116
single                        75
home_run                      52
double                        24
grounded_into_double_play     15
intent_walk                   11
force_out                     11
hit_by_pitch                   5
sac_fly                        4
fielders_choice_out            4
field_error                    4
triple                         3
strikeout_double_play          1
Name: count, dtype: int64
```

```
Giancarlo Stanton batted ball event totals, 2017:
events
field_out                    239
strikeout                    161
single                        77
walk                          72
home_run                      59
double                        32
intent_walk                   13
grounded_into_double_play     13
force_out                      7
hit_by_pitch                   7
field_error                    5
sac_fly                        3
fielders_choice_out            2
strikeout_double_play          2
pickoff_1b                     1
Name: count, dtype: int64
```

4. Analyzing home runs with Statcast data

So Judge walks and strikes out more than Stanton. Stanton flies out more than Judge. But let's get into their hitting profiles in more detail. Two of the most groundbreaking Statcast metrics are launch angle and exit velocity:

- Launch angle: the vertical angle at which the ball leaves a player's bat
- Exit velocity: the speed of the baseball as it comes off the bat

This new data has changed the way teams value both hitters and pitchers. Why? As per the Washington Post:

> Balls hit with a high launch angle are more likely to result in a hit. Hit fast enough and at the right angle, they become home runs.

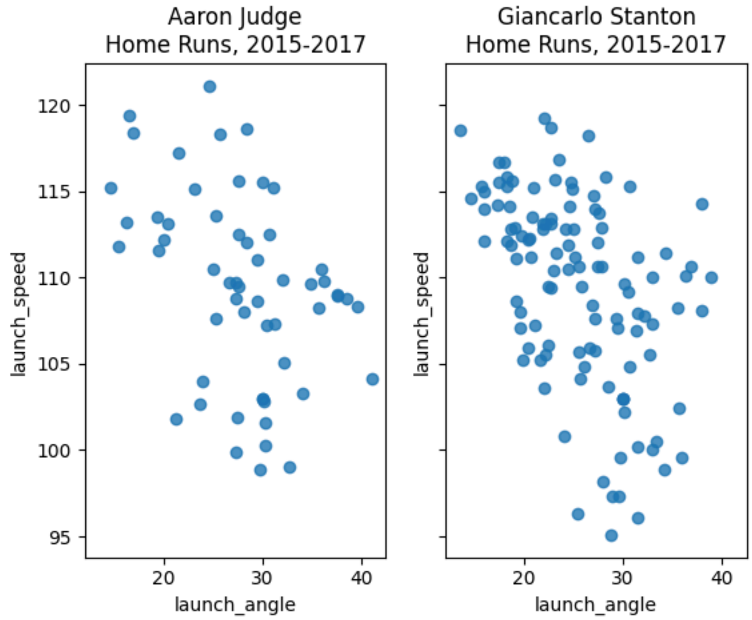
Let's look at exit velocity vs. launch angle and let's focus on home runs only (2015-2017). The first two plots show data points. The second two show smoothed contours to represent density.

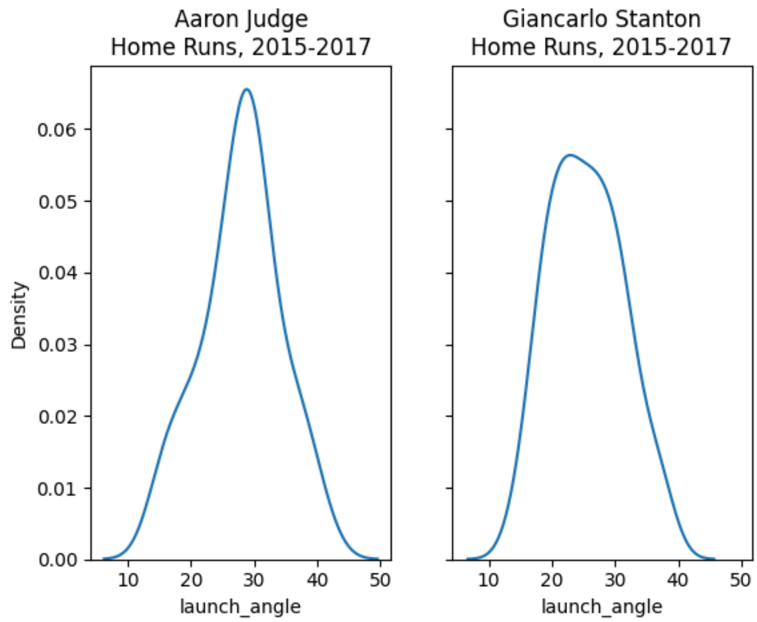
In [22]:
```
# Filter to include home runs only
judge_hr = judge[judge['events']=='home_run']
stanton_hr = stanton[stanton['events']=='home_run']

# Create a figure with two scatter plots of launch speed vs. launch angle, one for ea
fig1, axs1 = plt.subplots(ncols=2, sharex=True, sharey=True)
sns.regplot(x='launch_angle', y='launch_speed', fit_reg=False, color='tab:blue', data
sns.regplot(x='launch_angle', y='launch_speed', fit_reg=False, color='tab:blue', data

# Create a figure with two KDE plots of launch speed vs. launch angle, one for each p
fig2, axs2 = plt.subplots(ncols=2, sharex=True, sharey=True)
sns.kdeplot(judge_hr.launch_angle, ax=axs2[0]).set_title('Aaron Judge\nHome Runs, 201
sns.kdeplot(stanton_hr.launch_angle,ax=axs2[1]).set_title('Giancarlo Stanton\nHome Ru
```

Out [22]:
```
Text(0.5, 1.0, 'Giancarlo Stanton\nHome Runs, 2015-2017')
```





5. Home runs by pitch velocity

It appears that Stanton hits his home runs slightly lower and slightly harder than Judge, though this needs to be taken with a grain of salt given the small sample size of home runs.

Not only does Statcast measure the velocity of the ball coming off of the bat, it measures the velocity of the ball coming out of the pitcher's hand and begins its journey towards the plate. We can use this data to compare Stanton and Judge's home runs in terms of pitch velocity. Next you'll find box plots displaying the five-number summaries for each player: minimum, first quartile, median, third quartile, and maximum.

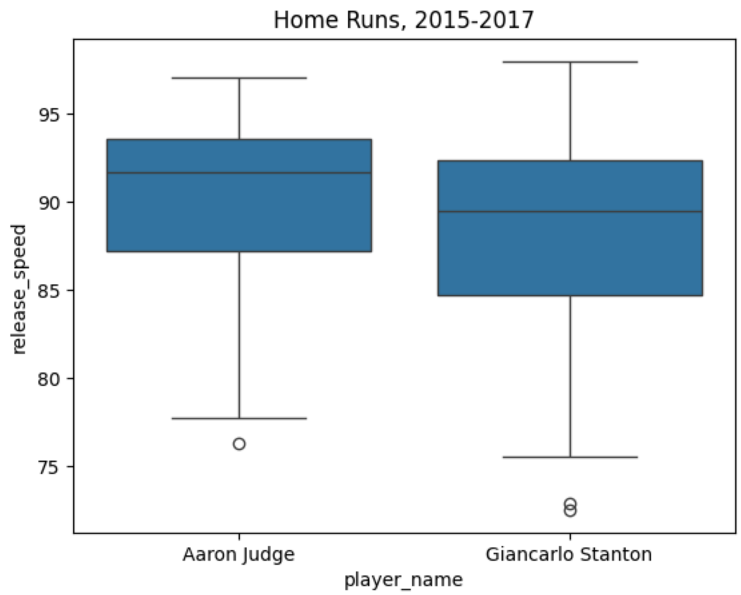
In [23]:

```
# Combine the Judge and Stanton home run DataFrames for easy boxplot plotting
judge_stanton_hr = pd.concat([judge_hr, stanton_hr])

# Create a boxplot that describes the pitch velocity of each player's home runs
sns.boxplot(x='player_name',y='release_speed', color='tab:blue', data=judge_stanton_h
```

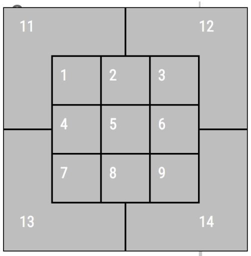
Out [23]:

```
Text(0.5, 1.0, 'Home Runs, 2015-2017')
```



6. Home runs by pitch location (I)

So Judge appears to hit his home runs off of faster pitches than Stanton. We might call Judge a fastball hitter. Stanton appears agnostic to pitch speed and likely pitch movement since slower pitches (e.g. curveballs, sliders, and changeups) tend to have more break. Statcast *does* track pitch movement and type but let's move on to something else: **pitch location**. Statcast tracks the zone the pitch is in when it crosses the plate. The zone numbering looks like this (from the catcher's point of view):



We can plot this using a 2D histogram. For simplicity, let's only look at strikes, which gives us a 9x9 grid. We can view each zone as coordinates on a 2D plot, the bottom left corner being (1,1) and the top right corner being (3,3). Let's set up a function to assign x-coordinates to each pitch.

In [24]:
```
def assign_x_coord(row):
    """
    Assigns an x-coordinate to Statcast's strike zone numbers. Zones 11, 12, 13,
    and 14 are ignored for plotting simplicity.
    """
    # Left third of strike zone
    if row.zone in [1, 4, 7]:
        return 1
    # Middle third of strike zone
    if row.zone in [2, 5, 8]:
        return 2
    # Right third of strike zone
    if row.zone in [3, 6, 9]:
        return 3
```

7. Home runs by pitch location (II)

And let's do the same but for y-coordinates.

In [25]:
```
def assign_y_coord(row):
    """
    Assigns a y-coordinate to Statcast's strike zone numbers. Zones 11, 12, 13,
    and 14 are ignored for plotting simplicity.
    """
    # Upper third of strike zone
    if row.zone in [1, 2, 3]:
        return 3
    # Middle third of strike zone
    if row.zone in [4, 5, 6]:
        return 2
    # Lower third of strike zone
    if row.zone in [7, 8, 9]:
        return 1
```

8. Aaron Judge's home run zone

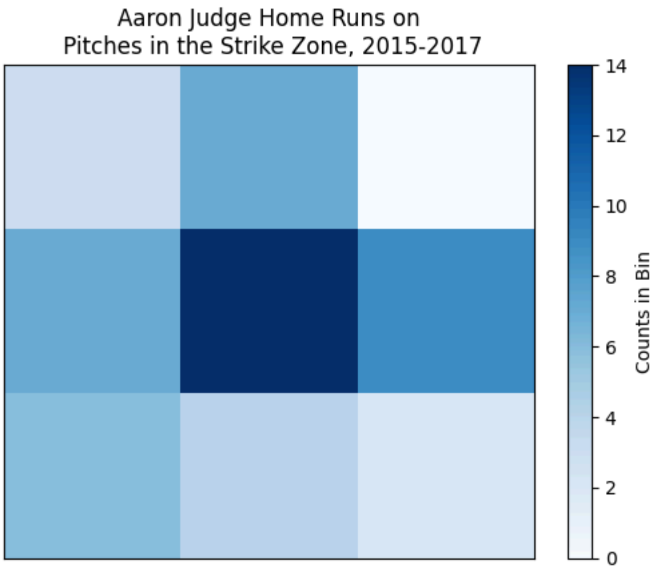
Now we can apply the functions we've created then construct our 2D histograms. First, for Aaron Judge (again, for pitches in the strike zone that resulted in home runs).

In [26]:
```
# Zones 11, 12, 13, and 14 are to be ignored for plotting simplicity
judge_strike_hr = judge_hr.copy().loc[judge_hr.zone <= 9]

# Assign Cartesian coordinates to pitches in the strike zone for Judge home runs
judge_strike_hr['zone_x'] = judge_strike_hr.apply(assign_x_coord, axis=1)
judge_strike_hr['zone_y'] = judge_strike_hr.apply(assign_y_coord, axis=1)

# Plot Judge's home run zone as a 2D histogram with a colorbar
plt.hist2d(judge_strike_hr['zone_x'], judge_strike_hr['zone_y'], bins = 3, cmap='Blue
```

```
plt.title('Aaron Judge Home Runs on\n Pitches in the Strike Zone, 2015-2017')
plt.gca().get_xaxis().set_visible(False)
plt.gca().get_yaxis().set_visible(False)
cb = plt.colorbar()
cb.set_label('Counts in Bin')
```



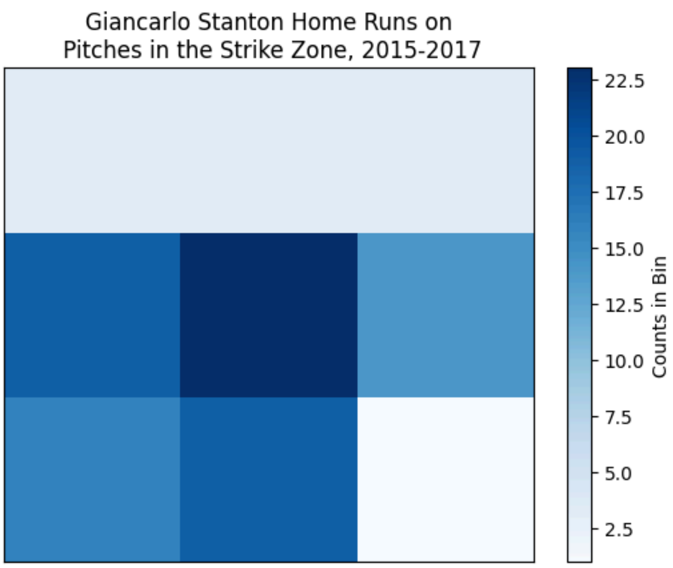
9. Giancarlo Stanton's home run zone

And now for Giancarlo Stanton.

In [27]:
```
# Zones 11, 12, 13, and 14 are to be ignored for plotting simplicity
stanton_strike_hr = stanton_hr.copy().loc[stanton_hr.zone <= 9]

# Assign Cartesian coordinates to pitches in the strike zone for Stanton home runs
stanton_strike_hr['zone_x'] = stanton_strike_hr.apply(assign_x_coord, axis=1)
stanton_strike_hr['zone_y'] = stanton_strike_hr.apply(assign_y_coord, axis=1)

# Plot Stanton's home run zone as a 2D histogram with a colorbar
plt.hist2d(stanton_strike_hr['zone_x'], stanton_strike_hr['zone_y'], bins = 3, cmap='
plt.title('Giancarlo Stanton Home Runs on\n Pitches in the Strike Zone, 2015-2017')
plt.gca().get_xaxis().set_visible(False)
plt.gca().get_yaxis().set_visible(False)
cb = plt.colorbar()
cb.set_label('Counts in Bin')
```



10. Should opposing pitchers be scared?

A few takeaways:

- Stanton does not hit many home runs on pitches in the upper third of the strike zone.
- Like pretty much every hitter ever, both players love pitches in the horizontal and vertical middle of the plate.
- Judge's least favorite home run pitch appears to be high-away while Stanton's appears to be low-away.
- If we were to describe Stanton's home run zone, it'd be middle-inside. Judge's home run zone is much more spread out.

The grand takeaway from this whole exercise: Aaron Judge and Giancarlo Stanton are not identical despite their superficial similarities. In terms of home runs, their launch profiles, as well as their pitch speed and location preferences, are different.

Should opposing pitchers still be scared?

In [28]:
```
# Should opposing pitchers be wary of Aaron Judge and Giancarlo Stanton
should_pitchers_be_scared = True
```