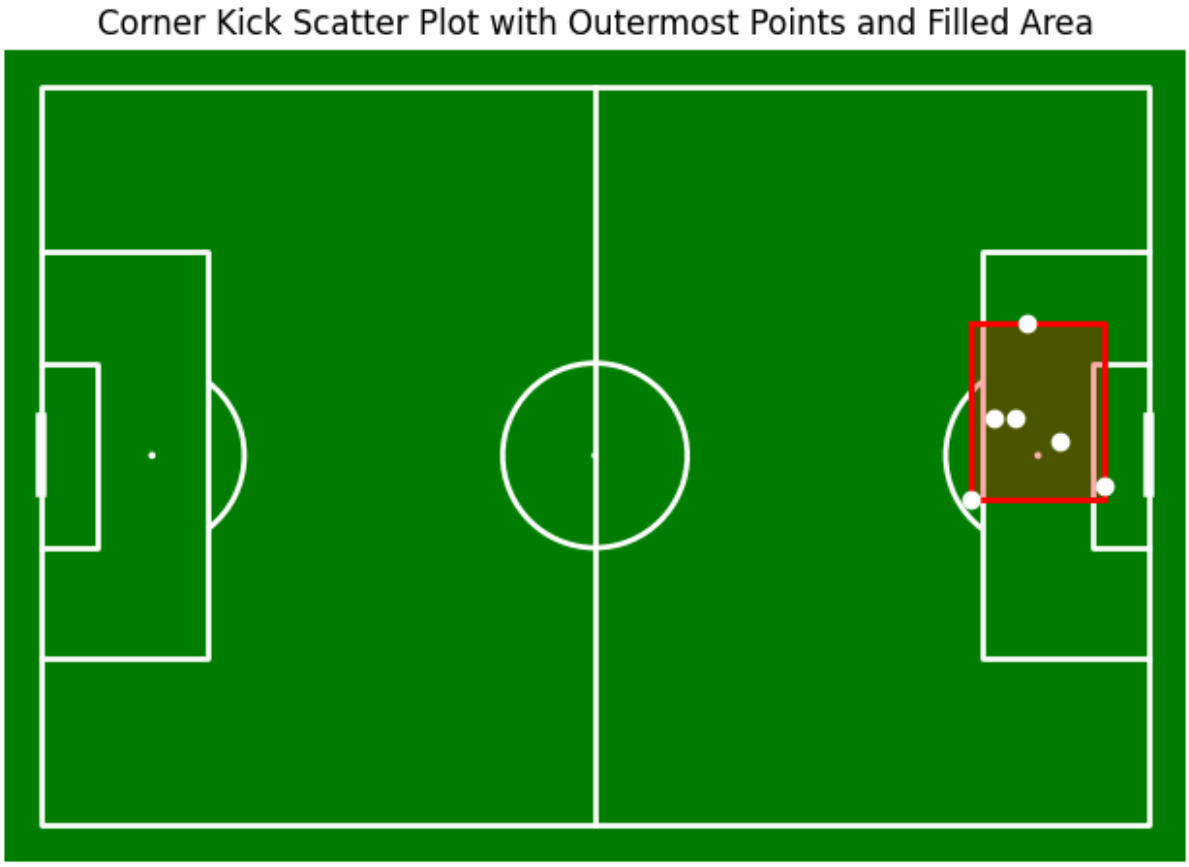
In [2]:

```
#jupyter notebook --no-browser --NotebookApp.token='' --NotebookApp.password='' --N
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from mplsoccer.pitch import Pitch

# Read corner kick coordinates from a CSV file
# Assuming the CSV file has columns named 'X2' and 'Y2'
# Replace with your actual file path
corner_kicks_df = pd.read_csv('F:/My Drive/Soccer Analytics Exports/CornerKick Map/
#Change to [X , Y, X2, Y2] Format from endX and Y2 from messibettis data
corner_kicks_df['X2'] *= 1.2
corner_kicks_df['Y2'] *= 0.8
# Can filter to those that are successful or not (scored or not) and if it landed o
#corner_kicks_df = corner_kicks_df[(corner_kicks_df['outcome'] == 'Successful') & (
# Create a pitch
corner_kicks_right = corner_kicks_df[corner_kicks_df['Y2'] < 40] # Left Half Ending
corner_kicks_left = corner_kicks_df[corner_kicks_df['Y2'] > 40] # Right Half Ending
pitch = Pitch(pitch_type='statsbomb', pitch_color='green', line_color='white')

# Draw the pitch
fig, ax = pitch.draw()

# Plot the corner kick locations - ADJUST FOR Left and Right or leave both for both
scatter = pitch.scatter(corner_kicks_df['X2'], corner_kicks_df['Y2'], ax=ax, color=

# Plot a line around the outermost points - Adjust for left or right or base
outermost_points_x = [corner_kicks_df['X2'].min(), corner_kicks_df['X2'].max(), cor
outermost_points_y = [corner_kicks_df['Y2'].min(), corner_kicks_df['Y2'].min(), cor
plt.plot(outermost_points_x, outermost_points_y, color='red', linestyle='-', linewi

# Fill the area between the outermost points
plt.fill(outermost_points_x, outermost_points_y, color='red', alpha=0.3)
plt.plot(outermost_points_x, outermost_points_y, color='red', linestyle='-', linewi
# Add title and legend
plt.title('Corner Kick Scatter Plot with Outermost Points and Filled Area')


# Display the plot
plt.show()
```



In [3]:

```
# Remake as a right side only
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from mplsoccer.pitch import Pitch

# Read corner kick coordinates from a CSV file
# Assuming the CSV file has columns named 'endX' and 'Y2'
# Replace with your actual file path
corner_kicks_df = pd.read_csv('F:/My Drive/Soccer Analytics Exports/CornerKick Map/

corner_kicks_df['X2'] *= 1.2
corner_kicks_df['Y2'] *= 0.8
# Can filter to those that are successful or not (scored or not) and if it landed o
#corner_kicks_df = corner_kicks_df[(corner_kicks_df['outcome'] == 'Successful') & (
# Create a pitch
corner_kicks_right = corner_kicks_df[corner_kicks_df['Y2'] < 40] # Left Half Ending
corner_kicks_left = corner_kicks_df[corner_kicks_df['Y2'] > 40] # Right Half Ending
pitch = Pitch(pitch_type='statsbomb', pitch_color='green', line_color='white')

# Draw the pitch
fig, ax = pitch.draw()

# Plot the corner kick locations - ADJUST FOR Left and Right or leave both for both
scatter = pitch.scatter(corner_kicks_right['X2'], corner_kicks_right['Y2'], ax=ax,

# Plot a line around the outermost points - Adjust for left or right or base !!!WOR
outermost_points_x = [corner_kicks_right['X2'].min(), corner_kicks_right['X2'].max(
outermost_points_y = [corner_kicks_df['Y2'].min(), corner_kicks_df['Y2'].min(), cor
```

```
plt.plot(outermost_points_x, outermost_points_y, color='red', linestyle='-', linewi

# Fill the area between the outermost points
plt.fill(outermost_points_x, outermost_points_y, color='red', alpha=0.3)
plt.plot(outermost_points_x, outermost_points_y, color='red', linestyle='-', linewi
# Add title and legend
plt.title('Corner Kick Scatter Plot with Outermost Points and Filled Area')


# Display the plot
plt.show()
```