Run Boy Run (An Analysis of Runs in Behind in Soccer)

Abstract

Off-the-ball runs in behind are a fundamental attacking mechanism in soccer, influencing both the creation of goal-scoring opportunities and the disruption of defensive structures. This study analyzes the characteristics of such runs and their relationship to shot creation using a large-scale spatial temporal tracking dataset provided by SkillCorner. By integrating player-tracking data with event level annotations of off-ball runs, we reconstructed possessions, identified whether each possession led to a shot, and extracted a comprehensive set of run level features: including spatial endpoints, run angle, speed, defensive line interactions, and positional relationships to the ball.

We first established that possessions containing at least one run in behind were **14% more likely** to result in a shot, confirming their attacking value. To identify which run characteristics predict shot outcomes, we conducted a series of one-variable logistic regression models, supported by distributional comparisons and correlation analysis to detect feature relevance and redundancy. Several features emerged as significant predictors of shot-producing possessions: end-location of the run, distance covered, width change during the run, diagonal run angle, change in defensive line height, and proximity of the runner to the ball carrier.

Analysis of shot-leading versus non-shot-leading runs revealed consistent patterns. Runs that travel longer distances, move centrally, end near the top of the penalty box, follow diagonal trajectories, create measurable defensive-line displacement, and occur closer to the player in possession all show increased likelihood of producing a shot. These findings align with tactical intuition around counterattacks, striker movement, and defensive manipulation.

While the linear models used cannot fully capture the complex interdependence between features, this work provides evidence-based insights into what constitutes an effective run in behind. The results offer a foundation for future modeling using more expressive nonlinear methods and provide practical guidance for coaching, player development, and data-driven tactical analysis.

Research Question

Are the features of off-the-ball runs in behind, identified through tracking data as player movements beyond the last line of defense without ball possession, significantly different in possessions that result in a shot compared to possessions that do not?

Background and Prior Work

The creation of goal-scoring opportunities in association football has become a central focus of applied analytics over the past decade, driven largely by the development of the expected goals (xG) framework. xG models estimate the probability that a given shot results in a goal based on measurable features such as shot distance, angle, and body part used. These models have allowed analysts to evaluate chance quality objectively rather than relying on subjective observation.[1] By providing a consistent probabilistic measure of shot quality, xG has reshaped how performance is analyzed, both at the team and individual player levels.

Traditional xG models rely primarily on event data, which record all on-ball actions such as passes, tackles, and shots, along with spatial and contextual metadata. This event-based structure allows for statistical modeling of goal probabilities from historical shot events. However, such models neglect important off-ball movements that precede or enable these shots. To evaluate the creation of attacking opportunities more holistically, researchers have begun integrating tracking data, which capture continuous player and ball positions at high frequency. Combining event and tracking data enables analysts to explore the influence of spatial organization, team shape, and player movement on the likelihood of creating high-xG opportunities.[2]

One of the leading frameworks for valuing in-possession actions is the Valuing Actions by Estimating Probabilities (VAEP) model, introduced by Decroos et al. (2020). VAEP assigns value to each on-ball event by measuring its effect on both the probability of scoring and conceding in the near future.[3] This framework provides a general approach for attributing value to game actions, but it is limited to on-ball events and cannot directly measure the contribution of off-ball player movements such as forward runs. Extending the VAEP concept to account for off-ball actions could help quantify how attacking movement changes the offensive state of a team's possession.

A major challenge is that off-ball runs are not explicitly recorded in most event datasets. To identify them, researchers have developed methods that rely on spatial-temporal tracking data and player trajectory analysis. Sam Gregory's Ready Player Run (2020) presented one of the first systematic methods to detect and classify off-ball runs, using positional data to cluster run types such as overlaps, penetrations, and decoy runs.[[4]](#cite_note-4) This approach provides a foundation for algorithmically recognizing when players make forward penetrating movements that stretch defenses or move beyond the back line—precisely the kind of runs that may precede higher-xG chances.

Building on this, William Spearman's Beyond Expected Goals (2018) introduced a probabilistic model of off-ball scoring opportunities (OBSO) that integrates pitch control and event-based goal probabilities to quantify spatial value across the field.[5] His work highlights how teams create and occupy valuable space off the ball, showing that controlling certain

areas of the pitch increases the probability of scoring. This framework can be adapted to assess whether forward runs penetrate into high-value spaces, thereby linking movement-based events to expected scoring outcomes.

By combining prior developments in xG modeling, VAEP valuation, and run detection, this project seeks to bridge the gap between off-ball player movement and on-ball outcome quality. Specifically, it will examine whether attacking sequences that include forward penetrating runs beyond the defensive back line produce significantly higher xG shots than sequences without such runs. This integration of spatial-temporal analysis, probabilistic modeling, and machine learning can provide new insight into how movement coordination and off-ball behavior contribute to efficient chance creation—an essential step toward a more comprehensive understanding of attacking effectiveness in soccer.

References

1. ^ The Football Analyst. (n.d.). Expected Goals (xG) – Football Statistics Explained. https://the-footballanalyst.com/expected-goals-xg-football-statistics-explained/
2. ^ StatsBomb. (2021). Event Data vs. Tracking Data in Football Analytics. https://statsbomb.com/articles/soccer/event-data-vs-tracking-data-in-football-analytics/
3. ^ Decroos, T., Bransen, L., Van Haaren, J., & Davis, J. (2020). VAEP: An Objective Approach to Valuing On-the-Ball Actions in Soccer. Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI). https://www.ijcai.org/Proceedings/2020/0648.pdf
4. ^ Gregory, S. (2020). Ready Player Run: Off-Ball Run Identification and Classification. FC Barcelona Sports Analytics Summit. https://static.capabiliaserver.com/frontend/clients/barca/wp_prod/wp-content/uploads/2020/01/19a5562d-ready-player-run-barcelona-sam-gregory.pdf
5. ^ Spearman, W. (2018). Beyond Expected Goals. MIT Sloan Sports Analytics Conference. https://www.researchgate.net/publication/327139841_Beyond_Expected_Goals

Hypothesis

Possessions that have runs in behind will lead to more shots and diagonal runs that break the defense's line will be the most dangerous; i.e. will lead to shots more often.

Data

Data overview

For each dataset include the following information

- Dataset 1
    - Dataset Name: Skill Corner Open Source Tracking Data
    - Link to the dataset:https://github.com/SkillCorner/opendata
    - Number of observations: 4600+ Off ball runs, 9 million associated tracking frames, 2700+ possessions
    - Number of variables: run features (67), tracking (18), possession (11)
    - Possessions that lead to goal, lead to shot, lead to box, run type, run features (60+ features)
    - No access to data about specific events associated with plays i.e. shots. There is no information about shot xG or timestamps of when shots occured.

    We have the ability to link possession, tracking, and run tables through match_id, event_id (run id), and possession_id columns that allow for easy merging and querying/filtering.

Skill Corner Open Source Australian A-League Tracking Data

The SkillCorner Open Source Australian A-League Tracking Dataset provides high-frequency positional data for all 22 players and the ball across multiple professional soccer matches in Australia's top division. Each frame of data captures player and ball locations at a rate of 10 frames per second (10 Hz), allowing for detailed reconstruction of match dynamics. Player positions are represented by their x and y coordinates in meters on a standardized pitch, typically 105 × 68 m. Additional computed metrics such as speed (m/s) and direction (radians) can be derived from positional data, offering insight into the tempo, spacing, and physical intensity of match phases.

This dataset is especially useful for analyzing off-the-ball movements, such as runs that create space or disrupt defensive structure. Since every frame captures all players and the frames that the runs occured in have already been tagged we are now ready to analyze the different aspects of runs. The fine temporal resolution supports the extraction of micro-movements (e.g., acceleration bursts or direction changes), which are crucial for understanding tactical intent beyond on-ball events.

However, there are important considerations when working with this dataset. First, it does not include event-level context (e.g., passes, shots, or possessions), so linking off-ball runs to outcomes like shot creation or defensive breakdowns requires external alignment with the table that gives us possession level information as this table contains whether a possession led to a shot. It is also limited in scope (covering a small number of matches) and may not represent the full tactical diversity of the A-League or professional soccer globally. Lastly, slight inconsistencies in tracking accuracy, camera calibration, or player ID assignment can introduce noise into fine-grained metrics like speed and acceleration—necessitating smoothing or filtering before analysis.

Overall, this dataset provides a rich foundation for exploring spatial-temporal dynamics and off-ball player behavior, enabling research into how players influence match outcomes without directly interacting with the ball.

Data was loaded and transformedd using the `collect_all_data` function in `get_data.py`.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt

match_info = pd.read_csv("data/02-processed/matches_info.csv")
possessions = pd.read_csv("data/02-processed/possessions.csv")
run_features = pd.read_csv("data/02-processed/run_features.csv")
tracking_data = pd.read_csv("data/02-processed/tracking_data.csv")
player_to_team = pd.read_csv("data/02-processed/player_team.csv").set_index("id")

merged = pd.merge(possessions,run_features,on=["match_id","possession_index"],how="

merged["possession_lead_to_shot"] = (merged["possession_lead_to_shot"] | merged["le
merged["possession_lead_to_goal"] = (merged["possession_lead_to_goal"] | merged["le

runs_behind = merged[merged.event_subtype == "behind"].dropna()
runs_behind["player_id"] = runs_behind["player_id"].astype(int)
```

Results

Exploratory Data Analysis

In our EDA we will be exploring the distributions for possessions and the target feature leading to a shot on goal. We will explore how many possessions lead to goals and whether if a possession contains runs affects that probability.

We will also be exploring for the possessions that do contain runs we will look into the distribution for specific run features such as run type, run curviness, xT of run, Pass probability of run, Num opponent's overtaken, angle of the run, and opponent's shape during the run.

Possession Based EDA

We will be exploring overall aspects of a possession that affect the target of leading to a shot

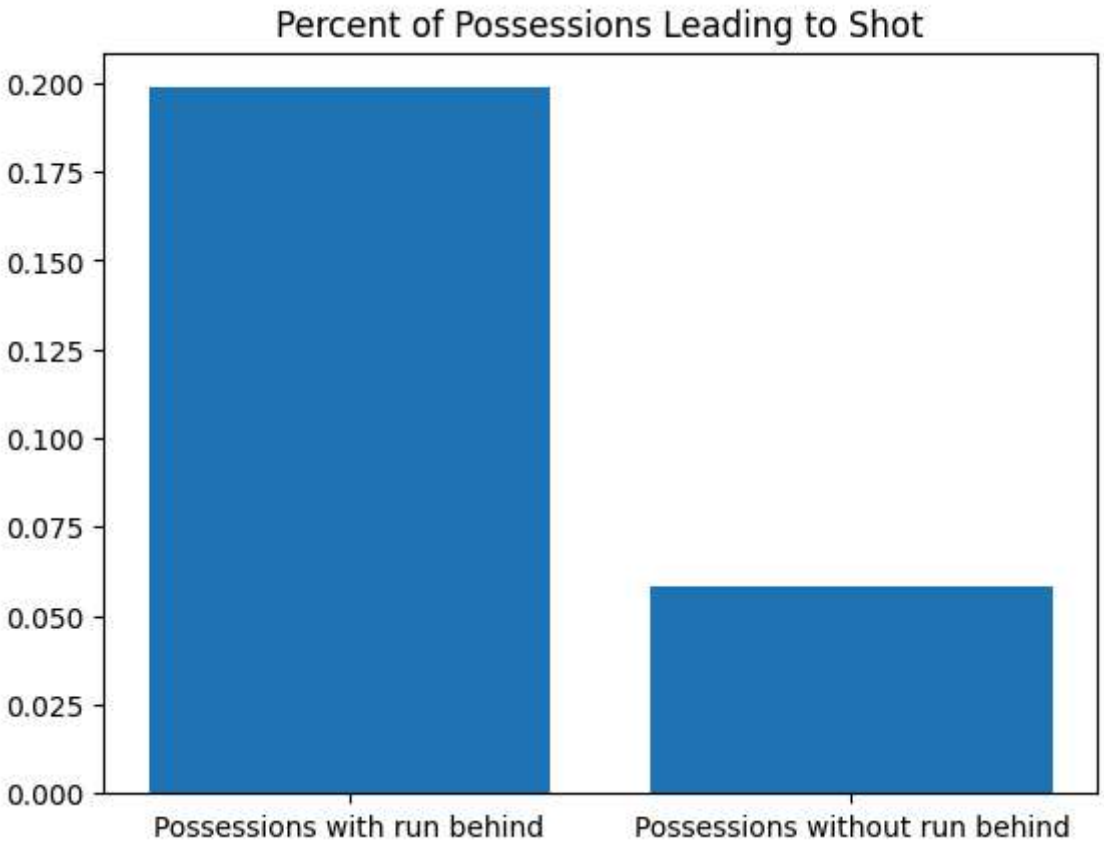
Percent of possessions that lead to a shot

In [2]:
```
merged[["match_id","possession_index","possession_lead_to_shot","event_subtype"]].d
```

Out[2]:
```
np.float64(0.10679611650485436)
```

In [3]:
```
run_behind_possessions = (
    merged.groupby(["match_id", "possession_index"])
          .filter(lambda g: (g["event_subtype"] == "behind").any())
)
no_run_behind_possessions = (
    merged.groupby(["match_id", "possession_index"])
          .filter(lambda g: not (g["event_subtype"] == "behind").any())
)
```

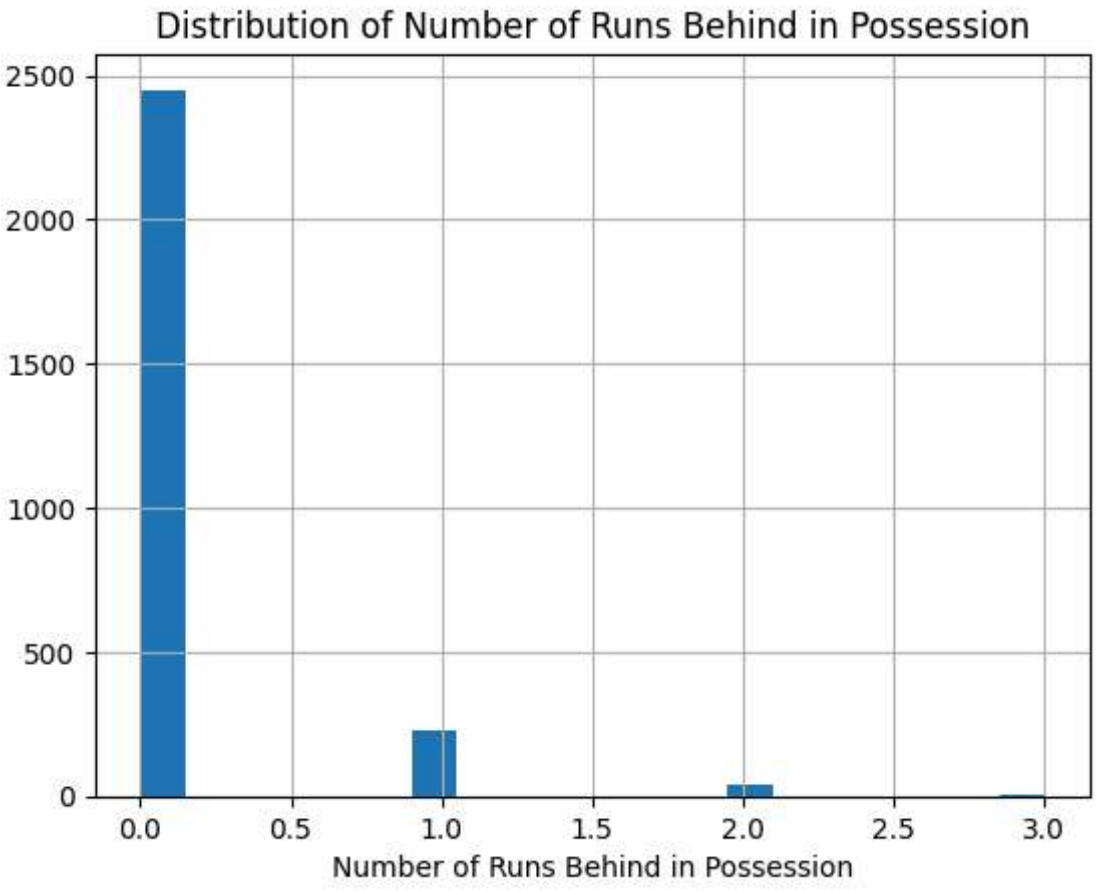
In [4]:
```
possessions_without_run_prob = no_run_behind_possessions[["match_id","possession_in

possessions_with_run_prob = run_behind_possessions[["match_id","possession_index","

plt.bar(x = ["Possessions with run behind","Possessions without run behind"],height
plt.title("Percent of Possessions Leading to Shot")
plt.show()
```



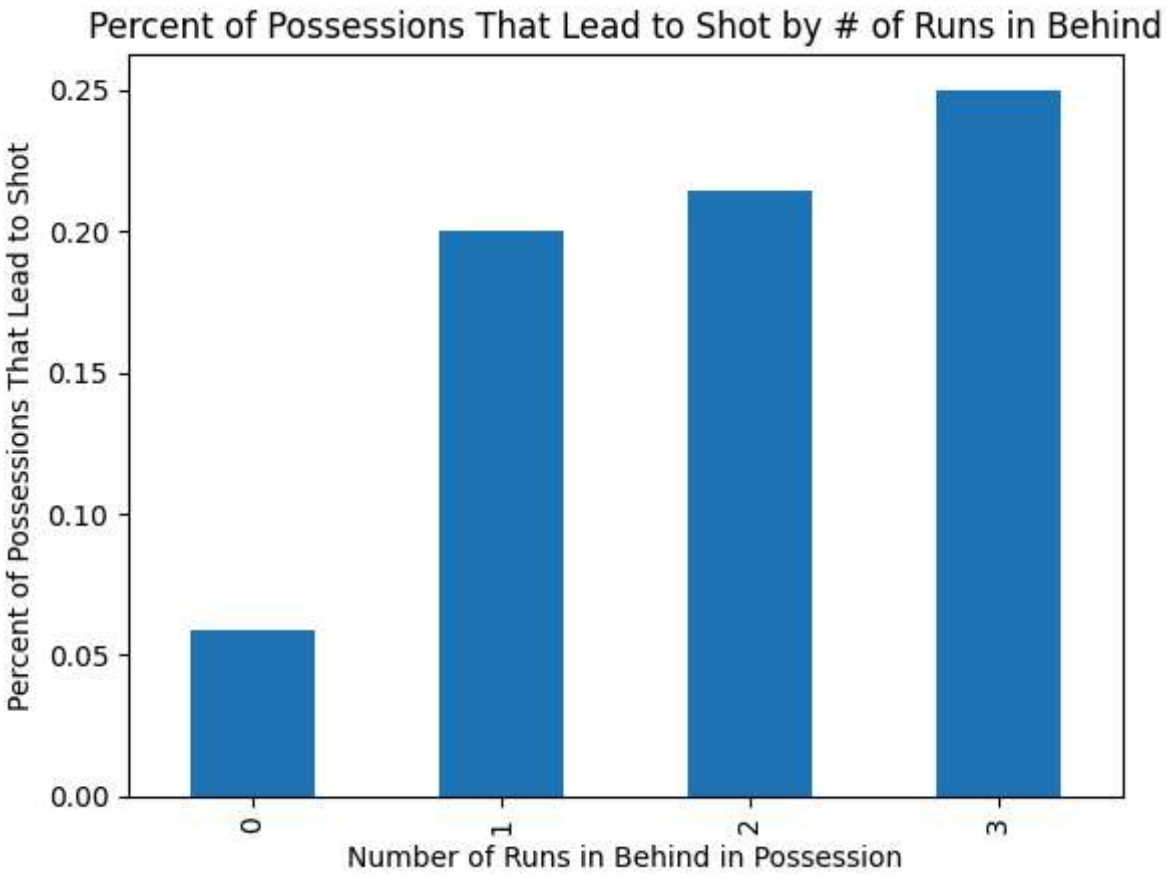
It appears that possessions that contain runs are more likely to result in shots

Number of runs in possession

In [5]:
```
merged = merged.assign(is_run_behind = merged.event_subtype == "behind")
merged.groupby(["match_id","possession_index"])[["is_run_behind"]].sum().hist(bins=
plt.title("Distribution of Number of Runs Behind in Possession")
plt.xlabel("Number of Runs Behind in Possession")
plt.show()
```



In [6]:
```
runs_per_poss = (
    merged.groupby(["match_id", "possession_index"])
          .agg({
              "is_run_behind": "sum",
              "possession_lead_to_shot": "max"
          })
          .rename(columns={"is_run_behind": "num_runs"})
)

runs_per_poss.groupby("num_runs")["possession_lead_to_shot"].mean().plot(kind="bar"
plt.xlabel("Number of Runs in Behind in Possession")
plt.ylabel("Percent of Possessions That Lead to Shot")
plt.title("Percent of Possessions That Lead to Shot by # of Runs in Behind")
plt.show()
```



It appears that within possessions that lead to a shot they tend to have more runs than fewer

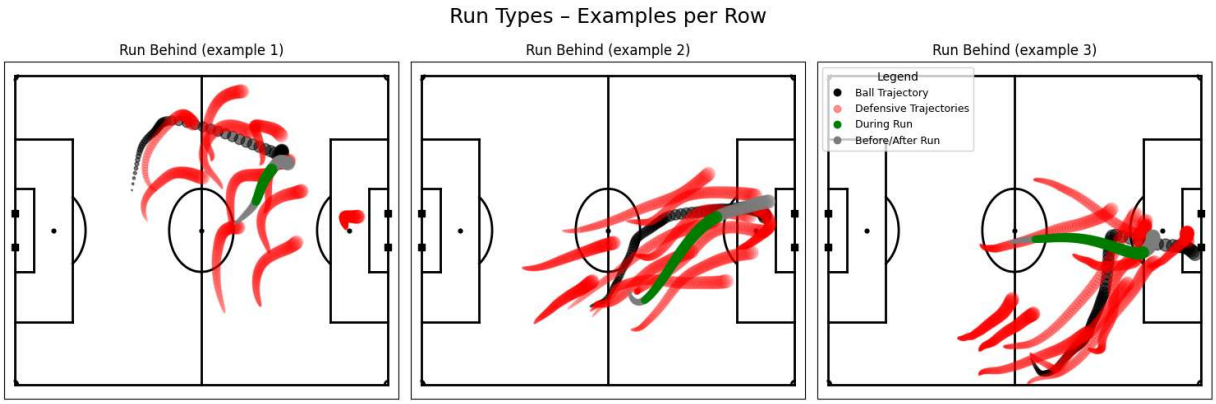
Run Based EDA

We will be looking at metrics that are specific to a run itself

Let's plot what a few runs in behind look like

In [7]:
```
from visualization_tools import plot_run
import numpy as np
runs_per_type = 3

fig, axes = plt.subplots(1, runs_per_type,
                         figsize=(5 * runs_per_type, 5))

axes = np.atleast_2d(axes)

selected = runs_behind.head(runs_per_type)

for j, (_, run) in enumerate(selected.iterrows()):
    ax = axes[0, j]

    plot_run(
        run,
```

```
        tracking_data=tracking_data,
        player_to_team=player_to_team,
        ax=ax,
        plot_ball=True,
        plot_defense=True,
        plot_offense=False,
        title=f"Run Behind (example {j+1})"
    )

# Add overall title
plt.suptitle("Run Types - Examples per Row", fontsize=18)
plt.tight_layout()
plt.show()
```



This cell below provides a visual animation of a run occuring

You can change `RUN_NUM` to view different runs in behind

In [23]:
```
%matplotlib tk

from visualization_tools import animate_run
RUN_NUM = 4

animate_run(runs_behind.iloc[RUN_NUM],tracking_data=tracking_data,player_to_team=pl
```

Out[23]:
```
<matplotlib.animation.FuncAnimation at 0x1f5abd1b9d0>
```

This will be good to keep in mind as we look at other metrics

Let's look at which run features on their own can significantly predict a shot occuring

1. Spatial Start/End Location

- **x_start** – Starting X-coordinate of the run
- **y_start** – Starting Y-coordinate of the run
- **x_end** – Ending X-coordinate of the run
- **y_end** – Ending Y-coordinate of the run

2. Defensive Line Interaction

- **last_defensive_line_height_start** – Height of the defensive line at start
- **last_defensive_line_height_gain** – Change in defensive line height during run
- **delta_to_last_defensive_line_start** – Initial distance to defensive line
- **delta_to_last_defensive_line_gain** – Distance gained relative to defensive line
- **broke_line** – Whether the run ended beyond the last defensive line (binary)

3. Opponent Interaction

- **n_opponents_overtaken** – Number of opponents passed during the run
- **inside_defensive_shape_start** – Whether start position was inside defensive structure
- **separation_start** – Initial separation from nearest defender
- **separation_gain** – Separation gained during the run

4. Run Geometry & Movement

- **trajectory_angle** – Direction of movement (degrees)
- **run_curve_ratio** – Bend/curvature of the run
- **distance_covered** – Total run length
- **distance_to_player_in_possession_start** – Distance from ball carrier at run start

We will be building a simple logistic regression for each one of these features and our target of wheter the possession lead to a shot. We will look at if the feature on it's own has a signifigant relationship to our target

In [9]:
```
import statsmodels.api as sm
runs_behind = runs_behind.assign(broke_line = runs_behind["delta_to_last_defensive_
runs_behind = runs_behind.assign(width_start = runs_behind["y_start"].abs())
runs_behind = runs_behind.assign(width_end = runs_behind["y_end"].abs())
runs_behind = runs_behind.assign(width_gain = runs_behind["width_end"] - runs_behin



features = ["n_opponents_overtaken","broke_line","trajectory_angle","x_start","y_st
def logistic_reg_pvalues(df, features, target="possession_lead_to_shot"):

    X = df[features].copy().astype(float)
    X = sm.add_constant(X)
    y = df[target]

    model = sm.Logit(y, X)
    result = model.fit(disp=False)

    summary_df = pd.DataFrame({
        "coef": result.params,
        "p_value": result.pvalues
    })
```

```
    return summary_df

for feature in features:
    p_values = logistic_reg_pvalues(runs_behind,feature)
    print(feature, ": ",p_values.loc[feature]["p_value"])
```

```
n_opponents_overtaken :  0.6631002775131609
broke_line :  0.63313256160814
trajectory_angle :  0.7708740924488703
x_start :  0.21039286273753532
y_start :  0.09487974815279984
x_end :  3.631794403302565e-08
y_end :  0.09508705043511861
width_start :  0.7056926311764824
width_end :  0.004734835685513114
width_gain :  0.0047125444634853846
run_curve_ratio :  0.9416767469267113
distance_covered :  9.253879023009517e-05
separation_gain :  0.8863921798821931
last_defensive_line_height_start :  0.014895687596903216
last_defensive_line_height_gain :  0.00020245373327404246
separation_start :  0.33167755542788846
delta_to_last_defensive_line_start :  0.011014181239192351
delta_to_last_defensive_line_gain :  0.010518449645970198
inside_defensive_shape_start :  0.2634934420126088
distance_to_player_in_possession_start :  8.830112967552322e-05
```

From this analysis we can now take a more in depth look at the relationships between the significant features and the run leading to a shot

We will take an in depth look at these run features:

- x, y coordinates at the start/end of the runs
- distance covered during the run
- width and direction of the run
- defensive line height and distance to runner
- distance from runner to player with ball

Let's see if any of our features are correlated to each other

In [10]:
```
corr_matrix = runs_behind[features].corr()

mask = np.triu(np.ones_like(corr_matrix, dtype=bool), k=1)

high_corr = corr_matrix.where(mask).stack().reset_index()
high_corr.columns = ['Feature_1', 'Feature_2', 'Correlation']

high_corr_filtered = high_corr[high_corr['Correlation'].abs() > 0.4]
high_corr_filtered
```

Out[10]:

	Feature_1	Feature_2	Correlation
15	n_opponents_overtaken	delta_to_last_defensive_line_start	0.576321
16	n_opponents_overtaken	delta_to_last_defensive_line_gain	-0.643269
55	x_start	x_end	0.563600
61	x_start	distance_covered	-0.549156
63	x_start	last_defensive_line_height_start	-0.936900
64	x_start	last_defensive_line_height_gain	0.475919
66	x_start	delta_to_last_defensive_line_start	-0.517988
67	x_start	delta_to_last_defensive_line_gain	0.495395
71	y_start	y_end	0.862654
92	x_end	last_defensive_line_height_start	-0.643688
112	width_start	width_end	0.679362
113	width_start	width_gain	-0.511975
122	width_start	inside_defensive_shape_start	-0.445768
155	distance_covered	last_defensive_line_height_start	0.402874
156	distance_covered	last_defensive_line_height_gain	-0.891406
158	distance_covered	delta_to_last_defensive_line_start	0.559054
159	distance_covered	delta_to_last_defensive_line_gain	-0.577609
164	separation_gain	separation_start	-0.739161
169	last_defensive_line_height_start	last_defensive_line_height_gain	-0.452723
184	delta_to_last_defensive_line_start	delta_to_last_defensive_line_gain	-0.955832

Here we see that there are some pretty strong correlations between the features we selected to look at. On one hand we would expect quite a few of these correlations to be present as runs are typically within the same area of the field and the relationship between the runner and the defensive line will be quite similiar.

In [11]:
```
significant_features = ["x_end","distance_covered","width_end","width_gain","last_d
logistic_reg_pvalues(runs_behind,significant_features)
```

Out[11]:

	coef	p_value
const	-14.437096	0.557997
x_end	0.295574	0.528335
distance_covered	-0.052022	0.438012
width_end	-0.036961	0.146418
width_gain	-0.053054	0.048311
last_defensive_line_height_start	0.220158	0.637228
last_defensive_line_height_gain	0.147318	0.754021
delta_to_last_defensive_line_start	0.314599	0.514508
delta_to_last_defensive_line_gain	0.176737	0.715031
distance_to_player_in_possession_start	-0.047529	0.010221

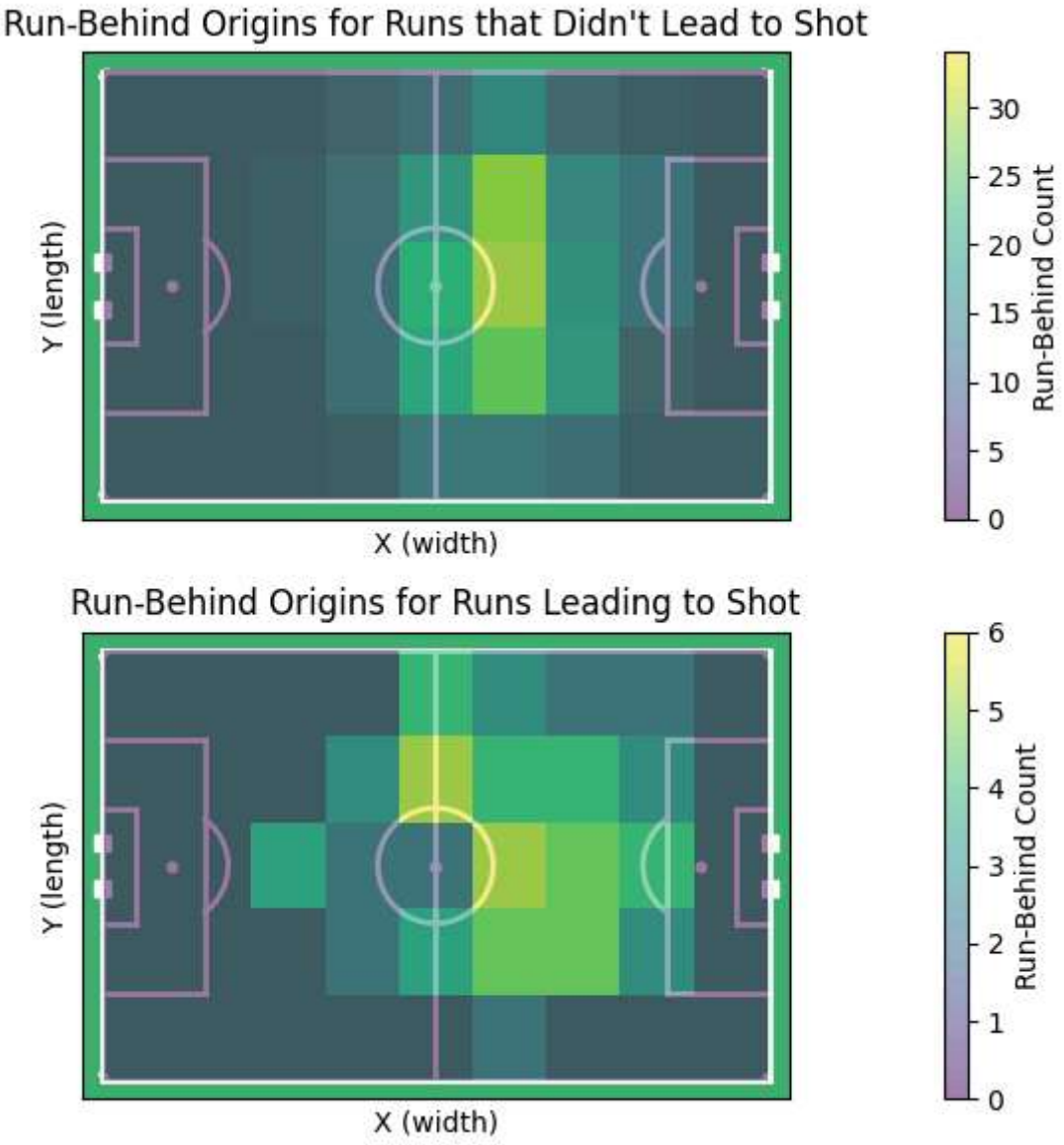
Here we have built a logistic regression model using the list of features that were significant on their own. We can see that due to the correlations many of these features are now insignificant.

We see that logistic regression determined that the majority of the variance in the possession leading to a shot can be explained by the change in width throughout the run and the distance from the runner to the player with the ball at the start of the run. We see that if a run starts wider and comes to the middle to expected probability of leading to a goal increases and if the runner starts closer to the player in possession the expected probability of leading to a goal increases as well.

Let's start off by exploring how the geometry of run origins affects getting a shot

In [21]:
```
%matplotlib inline
from visualization_tools import plot_soccer_pitch

x = runs_behind["x_start"].values
y = runs_behind["y_start"].values

x_shot = runs_behind[runs_behind.possession_lead_to_shot]["x_start"].values
y_shot = runs_behind[runs_behind.possession_lead_to_shot]["y_start"].values
x_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["x_start"].values
y_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["y_start"].values

x_min, x_max = -52.5, 52.5
y_min, y_max = -34, 34

num_x_bins = 9   # horizontal bins
num_y_bins = 5   # vertical bins
```

```
H_count_shot, x_edges_shot, y_edges_shot = np.histogram2d(
    x_shot, y_shot,
    bins=[num_x_bins, num_y_bins],
    range=[[x_min, x_max], [y_min, y_max]]
)
H_count_no_shot, x_edges_no_shot, y_edges_no_shot = np.histogram2d(
    x_no_shot, y_no_shot,
    bins=[num_x_bins, num_y_bins],
    range=[[x_min, x_max], [y_min, y_max]]
)
fig, axes = plt.subplots(2,1, figsize=(16, 6))

ax = axes[0]
plot_soccer_pitch(ax=ax)

im1 = ax.imshow(
    H_count_no_shot.T,
    origin='lower',
    extent=[x_min, x_max, y_min, y_max],
    aspect='equal',
    alpha=0.5
)

cbar1 = fig.colorbar(im1, ax=ax)
cbar1.set_label("Run-Behind Count")

ax.set_title("Run-Behind Origins for Runs that Didn't Lead to Shot")
ax.set_xlabel("X (width)")
ax.set_ylabel("Y (length)")

ax = axes[1]
plot_soccer_pitch(ax=ax)

im2 = ax.imshow(
    H_count_shot.T,
    origin='lower',
    extent=[x_min, x_max, y_min, y_max],
    aspect='equal',
    alpha=0.5
)

cbar2 = fig.colorbar(im2, ax=ax)
cbar2.set_label("Run-Behind Count")

ax.set_title("Run-Behind Origins for Runs Leading to Shot")
ax.set_xlabel("X (width)")
ax.set_ylabel("Y (length)")


plt.tight_layout()
plt.show()
```



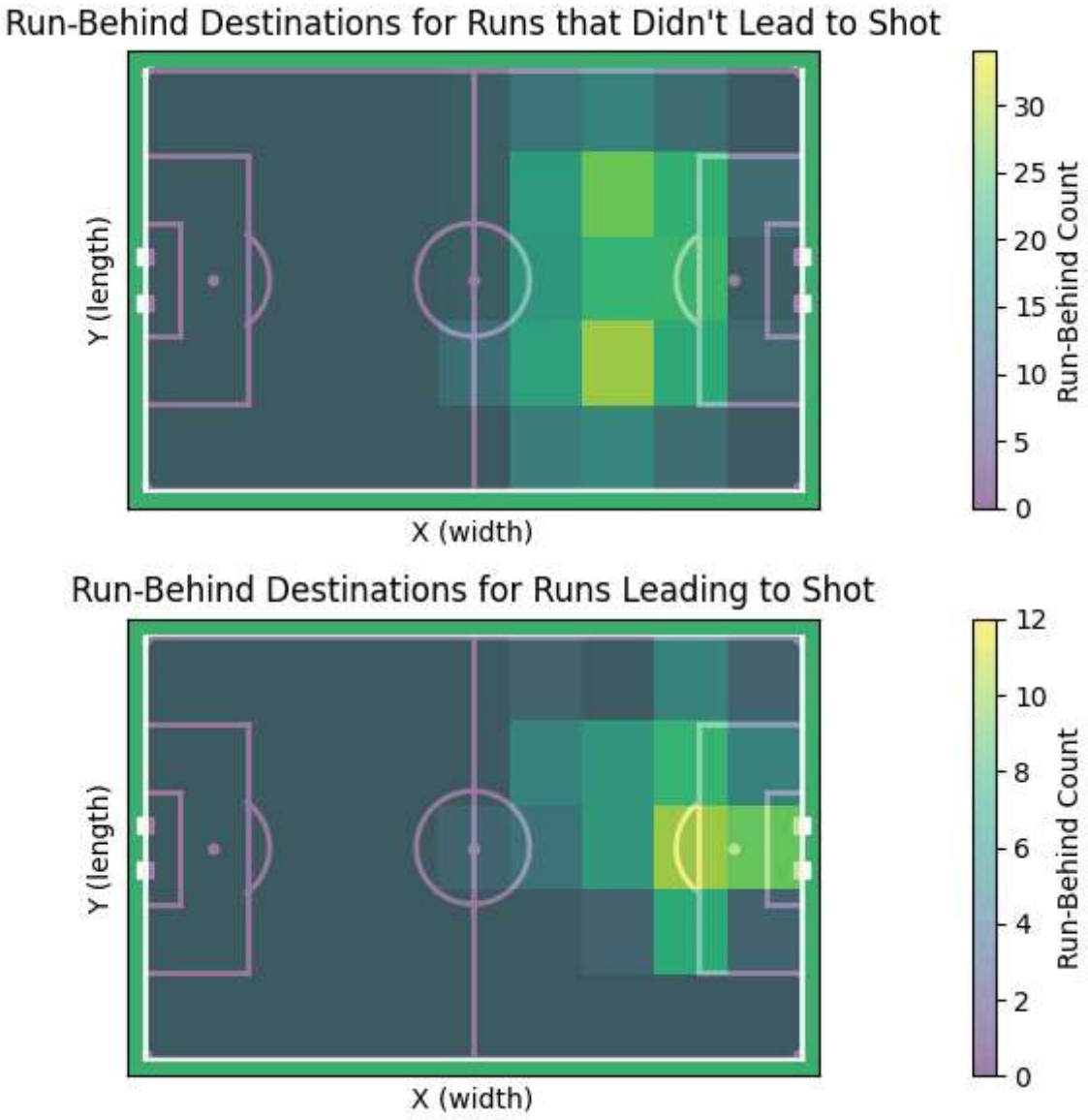
This matches our intuition that runs that start closer to the opponent's goal will lead to more shots

How does the ending location of runs vary in possessions that lead to shots

In [13]:
```
%matplotlib inline
from visualization_tools import plot_soccer_pitch

x = runs_behind["x_end"].values
y = runs_behind["y_end"].values

x_shot = runs_behind[runs_behind.possession_lead_to_shot]["x_end"].values
y_shot = runs_behind[runs_behind.possession_lead_to_shot]["y_end"].values
x_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["x_end"].values
y_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["y_end"].values

x_min, x_max = -52.5, 52.5
```

```
y_min, y_max = -34, 34

num_x_bins = 9   # horizontal bins
num_y_bins = 5   # vertical bins


H_count_shot, x_edges_shot, y_edges_shot = np.histogram2d(
    x_shot, y_shot,
    bins=[num_x_bins, num_y_bins],
    range=[[x_min, x_max], [y_min, y_max]]
)
H_count_no_shot, x_edges_no_shot, y_edges_no_shot = np.histogram2d(
    x_no_shot, y_no_shot,
    bins=[num_x_bins, num_y_bins],
    range=[[x_min, x_max], [y_min, y_max]]
)
fig, axes = plt.subplots(2, 1, figsize=(16, 6))

ax = axes[0]
plot_soccer_pitch(ax=ax)

im1 = ax.imshow(
    H_count_no_shot.T,
    origin='lower',
    extent=[x_min, x_max, y_min, y_max],
    aspect='equal',
    alpha=0.5
)

cbar1 = fig.colorbar(im1, ax=ax)
cbar1.set_label("Run-Behind Count")

ax.set_title("Run-Behind Destinations for Runs that Didn't Lead to Shot")
ax.set_xlabel("X (width)")
ax.set_ylabel("Y (length)")

ax = axes[1]
plot_soccer_pitch(ax=ax)

im2 = ax.imshow(
    H_count_shot.T,
    origin='lower',
    extent=[x_min, x_max, y_min, y_max],
    aspect='equal',
    alpha=0.5
)

cbar2 = fig.colorbar(im2, ax=ax)
cbar2.set_label("Run-Behind Count")

ax.set_title("Run-Behind Destinations for Runs Leading to Shot")
ax.set_xlabel("X (width)")
ax.set_ylabel("Y (length)")
```

```
plt.tight_layout()
plt.show()
```

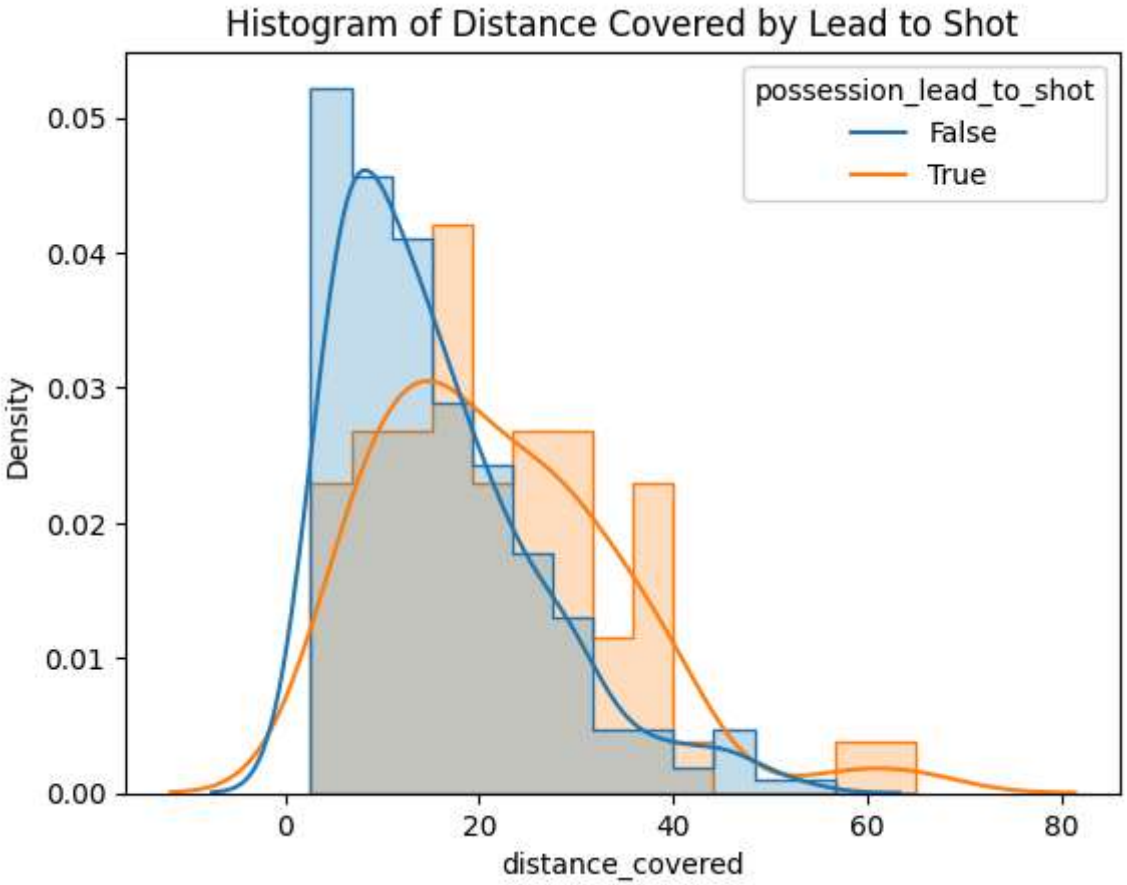


This also matches our intution that runs that are made into the center/top of the box will lead to more shots

Do runs in behind that cover more distance result in more shots?

In [14]:
```
import seaborn as sns

sns.histplot(data=runs_behind, x='distance_covered', hue='possession_lead_to_shot',
sns.kdeplot(data=runs_behind, x='distance_covered', hue='possession_lead_to_shot',
plt.title('Histogram of Distance Covered by Lead to Shot')
plt.show()

from scipy.stats import mannwhitneyu

dist_shot = runs_behind.loc[runs_behind['possession_lead_to_shot'], 'distance_cover
```

```
dist_no_shot = runs_behind.loc[~runs_behind['possession_lead_to_shot'], 'distance_c

u_stat, p_val_u = mannwhitneyu(dist_shot, dist_no_shot, alternative='greater')
print(f"Mann-Whitney U test: U-statistic = {u_stat}, p-value = {p_val_u:.3}")
```



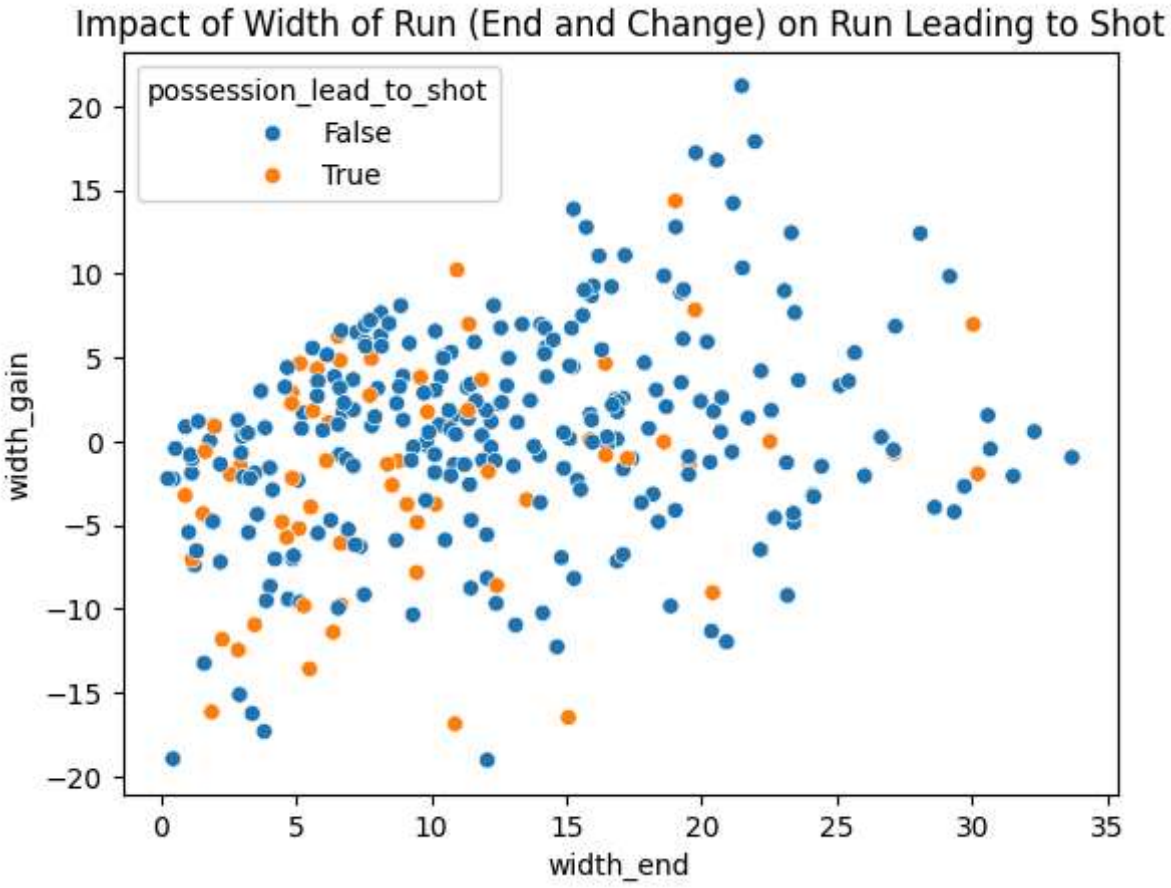

```
Mann-Whitney U test: U-statistic = 10783.5, p-value = 2.89e-05
```

Yes runs in behind that cover more distance tend to result in shots more than runs that cover short distance. However, this could be due to confounds such as if the team in possession lost the ball right after a player starts making a run and they have to cut their run short. This feature could just be a result of things happend to go well in the possession so the player had to make their run longer.

What impact does width have on leading to a shot?

Here we are looking at how runs of various widths (far from the center of the field) and how the width changed throughout the run to visualize the difference in distributions of leading to shots and not leading to shots

In [15]:
```
sns.scatterplot(data = runs_behind,x="width_end",y="width_gain",hue="possession_lea
plt.title("Impact of Width of Run (End and Change) on Run Leading to Shot")
plt.show()
```

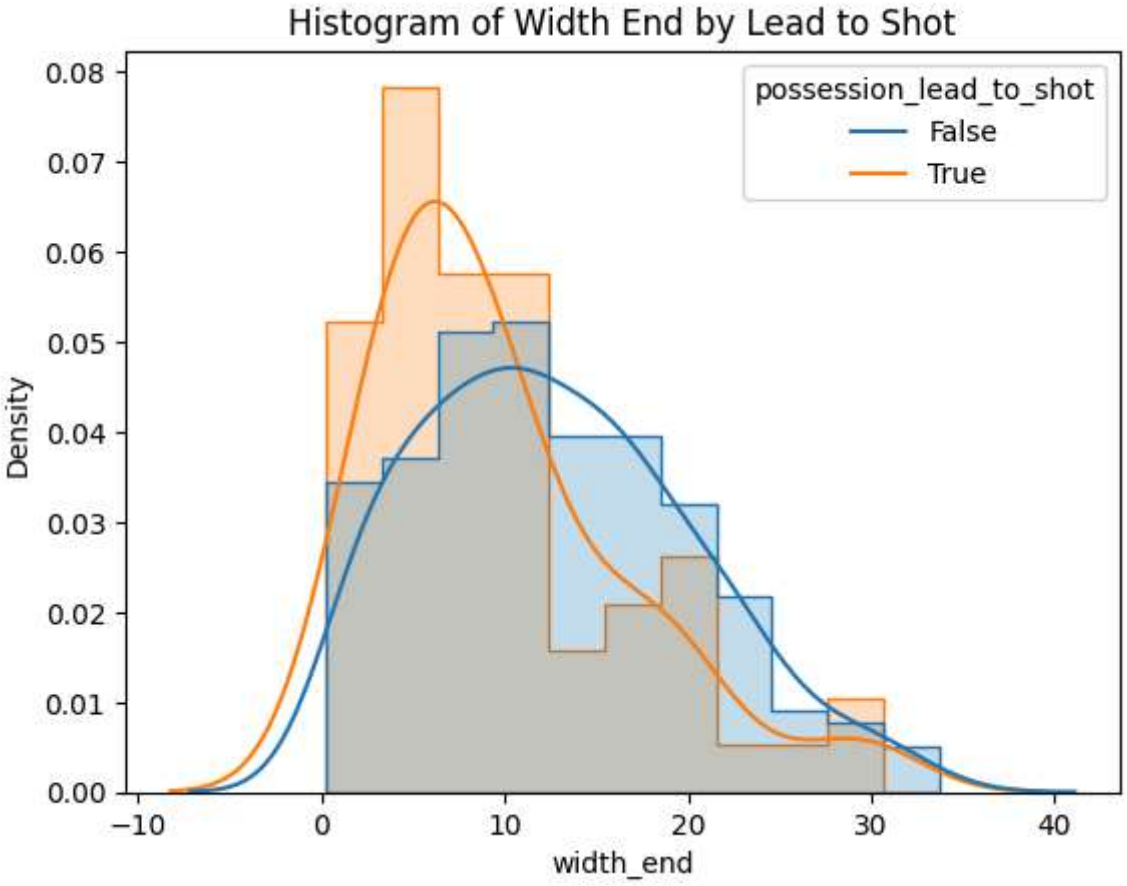


Here we can see that runs that started wider and finished in the center of the field have a lot higher of a chance of being involved in a shot than runs that started wide and stayed wide or etc.

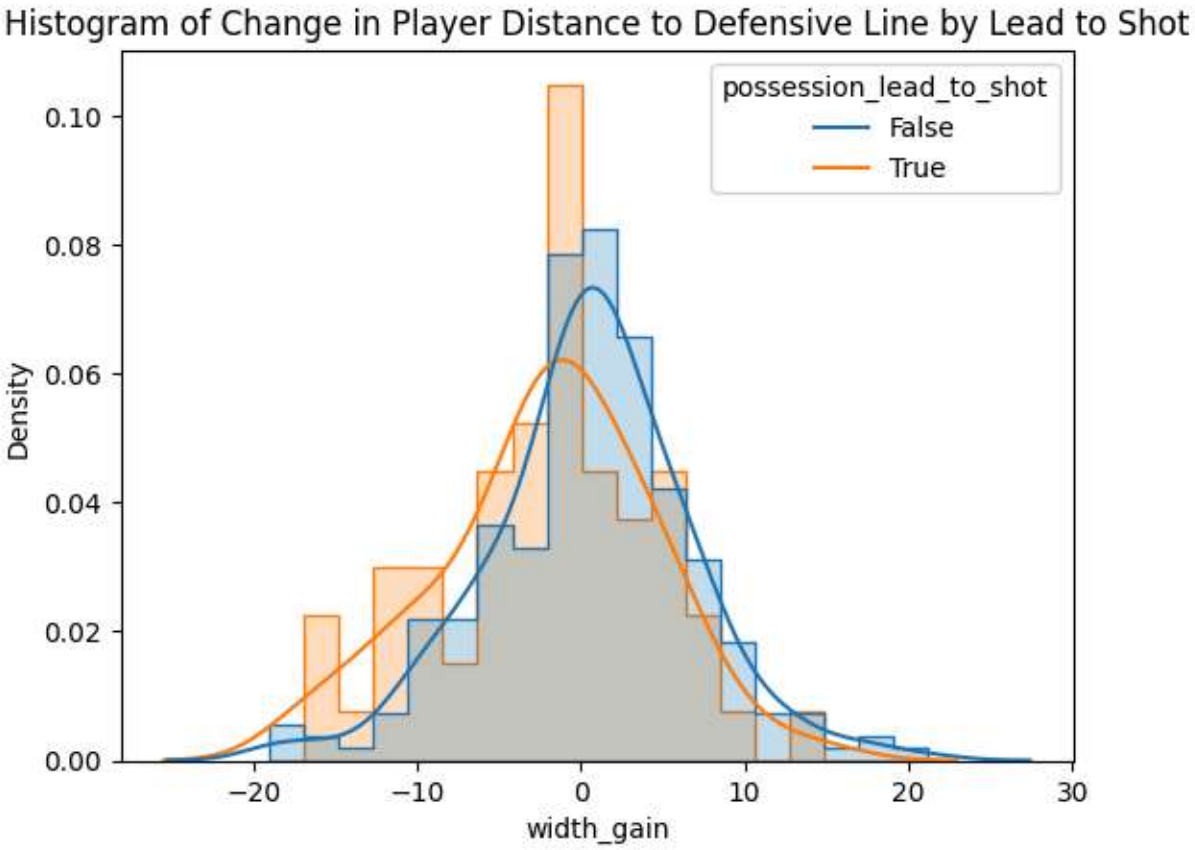
This matches what coaches are commonly telling the striker/wingers is to start wide and make their runs across the Center Backs into the middle of the field.

In [16]:
```
sns.histplot(data=runs_behind, x='width_end', hue='possession_lead_to_shot',
             element='step', stat='density', common_norm=False)
sns.kdeplot(data=runs_behind, x='width_end', hue='possession_lead_to_shot',
            common_norm=False)
plt.title('Histogram of Width End by Lead to Shot')
plt.show()

from scipy.stats import ttest_ind

dist_shot = runs_behind.loc[runs_behind['possession_lead_to_shot'], 'width_end']
dist_no_shot = runs_behind.loc[~runs_behind['possession_lead_to_shot'], 'width_end'

t_stat, p_val = ttest_ind(dist_shot, dist_no_shot, equal_var=False)
print(f"T-test for Width at End of Run: t = {t_stat:.3f}, p = {p_val:.3g}")

sns.histplot(data=runs_behind, x='width_gain', hue='possession_lead_to_shot',
             element='step', stat='density', common_norm=False)
sns.kdeplot(data=runs_behind, x='width_gain', hue='possession_lead_to_shot',
            common_norm=False)
plt.title('Histogram of Change in Player Distance to Defensive Line by Lead to Shot
```

```
plt.show()

dist_shot = runs_behind.loc[runs_behind['possession_lead_to_shot'], 'width_gain']
dist_no_shot = runs_behind.loc[~runs_behind['possession_lead_to_shot'], 'width_gain

t_stat, p_val = ttest_ind(dist_shot, dist_no_shot, equal_var=False)
print(f"T-test for Change in Width During Run: t = {t_stat:.3f}, p = {p_val:.3g}")
```




```
T-test for Width at End of Run: t = -3.031, p = 0.0031
```




```
T-test for Change in Width During Run: t = -2.814, p = 0.00599
```

Here we can see that the distribution of possessions that lead to shots is significantly different than possessions that don't lead to shots for both the ending width of runs and the change in width of runs.

How does the angle of the run at different parts of the field affect run probability

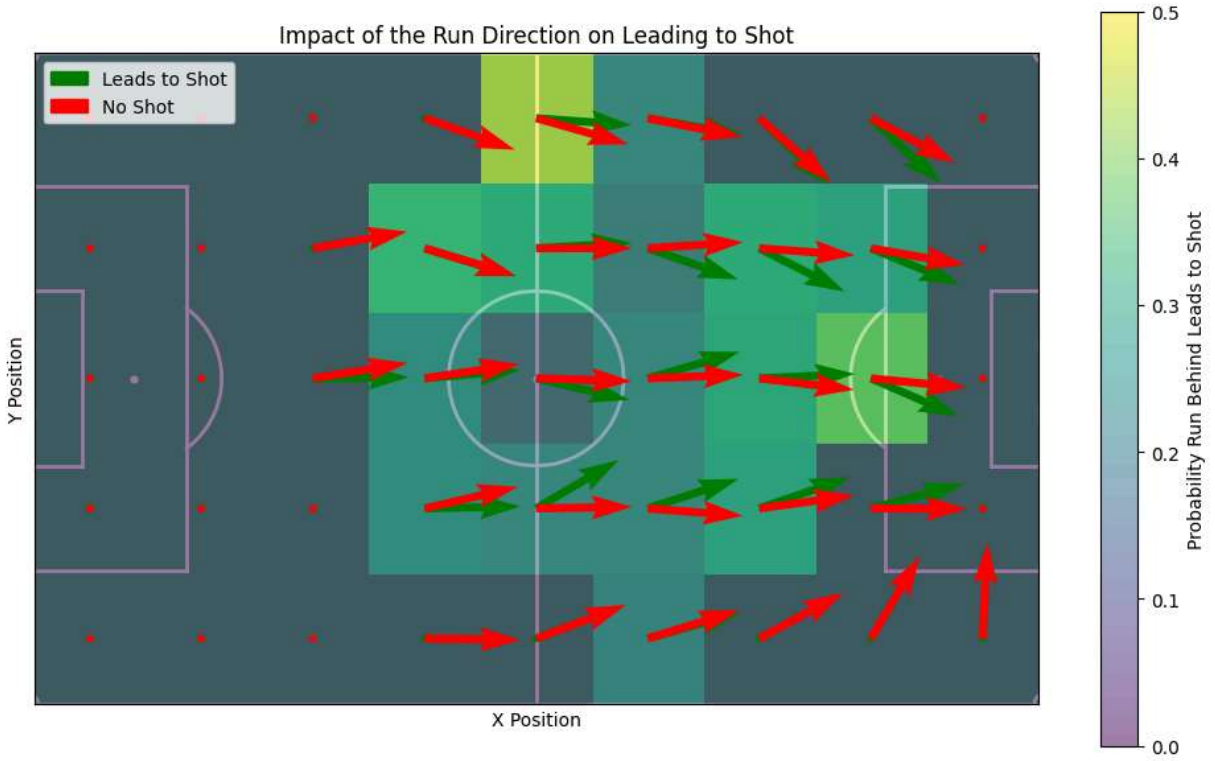
The common notion in soccer is that diagonal runs are better than straight runs, so can we see in our data if the angle of the run is able to show that diagonal runs tend to lead to more shots

In [22]:
```
x_shot = runs_behind[runs_behind.possession_lead_to_shot]["x_start"].values
y_shot = runs_behind[runs_behind.possession_lead_to_shot]["y_start"].values
x_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["x_start"].values
y_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["y_start"].values

MIN_RUNS = 5

angles_shot = runs_behind[runs_behind.possession_lead_to_shot]["trajectory_angle"].
angles_no_shot = runs_behind[~runs_behind.possession_lead_to_shot]["trajectory_angl
weights = runs_behind["possession_lead_to_shot"].values  # 0/1

H_count, x_edges, y_edges = np.histogram2d(
    x, y,
    bins=[num_x_bins, num_y_bins],
```

```
    range=[[x_min, x_max], [y_min, y_max]]
)
H_count_shot, x_edges_shot, y_edges_shot = np.histogram2d(
    x_shot, y_shot,
    bins=[num_x_bins, num_y_bins],
    range=[[x_min, x_max], [y_min, y_max]]
)
H_count_no_shot, x_edges_no_shot, y_edges_no_shot = np.histogram2d(
    x_no_shot, y_no_shot,
    bins=[num_x_bins, num_y_bins],
    range=[[x_min, x_max], [y_min, y_max]]
)


heat_map = np.zeros_like(H_count, dtype=float)
for i in range(num_x_bins):
    for j in range(num_y_bins):
        if H_count[i,j]<MIN_RUNS:
            continue

        in_bin = (
            (x >= x_edges[i]) & (x < x_edges[i+1]) &
            (y >= y_edges[j]) & (y < y_edges[j+1])
        )
        if np.any(in_bin):
            heat_map[i, j] = weights[in_bin].mean()
        else:
            heat_map[i, j] = 0


arrow_dx_shot = np.zeros_like(H_count_shot)
arrow_dy_shot = np.zeros_like(H_count_shot)

for i in range(num_x_bins):
    for j in range(num_y_bins):
        in_bin = (
            (x_shot >= x_edges_shot[i]) & (x_shot < x_edges_shot[i + 1]) &
            (y_shot >= y_edges_shot[j]) & (y_shot < y_edges_shot[j + 1])
        )

        if np.any(in_bin):
            avg_angle_deg = angles_shot[in_bin].mean()
            avg_angle_rad = np.deg2rad(avg_angle_deg)

            arrow_dx_shot[i, j] = 10 * np.cos(avg_angle_rad)
            arrow_dy_shot[i, j] = 10 * np.sin(avg_angle_rad)


arrow_dx_no_shot = np.zeros_like(H_count_shot)
arrow_dy_no_shot = np.zeros_like(H_count_shot)

for i in range(num_x_bins):
    for j in range(num_y_bins):
        in_bin = (
            (x_no_shot >= x_edges_no_shot[i]) & (x_no_shot < x_edges_no_shot[i + 1]
            (y_no_shot >= y_edges_no_shot[j]) & (y_no_shot < y_edges_no_shot[j + 1]
```

```
        )

        if np.any(in_bin):
            avg_angle_deg = angles_no_shot[in_bin].mean()
            avg_angle_rad = np.deg2rad(avg_angle_deg)

            arrow_dx_no_shot[i, j] = 10 * np.cos(avg_angle_rad)
            arrow_dy_no_shot[i, j] = 10 * np.sin(avg_angle_rad)



fig, ax = plt.subplots(figsize=(10, 6))
plot_soccer_pitch(ax=ax)

# Heatmap
im = ax.imshow(
    heat_map.T,
    origin="lower",
    extent=[x_min, x_max, y_min, y_max],
    cmap="viridis",
    alpha=0.5
)

plt.colorbar(im, ax=ax, label="Probability Run Behind Leads to Shot")

x_centers = 0.5 * (x_edges[:-1] + x_edges[1:])
y_centers = 0.5 * (y_edges[:-1] + y_edges[1:])

Xc, Yc = np.meshgrid(x_centers, y_centers, indexing="ij")

ax.quiver(
    Xc, Yc,
    arrow_dx_shot, arrow_dy_shot,
    color="green",
    scale=1,
    scale_units="xy",
    width=0.008
)
ax.quiver(
    Xc, Yc,
    arrow_dx_no_shot, arrow_dy_no_shot,
    color="red",
    scale=1,
    scale_units="xy",
    width=0.008
)

import matplotlib.patches as mpatches

legend_elements = [
    mpatches.FancyArrow(0, 0, 0.5, 0, color="green", label="Leads to Shot"),
    mpatches.FancyArrow(0, 0, 0.5, 0, color="red", label="No Shot"),
]

ax.legend(handles=legend_elements, loc="upper left")
```

```
ax.set_title("Impact of the Run Direction on Leading to Shot")
ax.set_xlabel("X Position")
ax.set_ylabel("Y Position")
ax.set_xlim(x_min, x_max)
ax.set_ylim(y_min, y_max)
ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

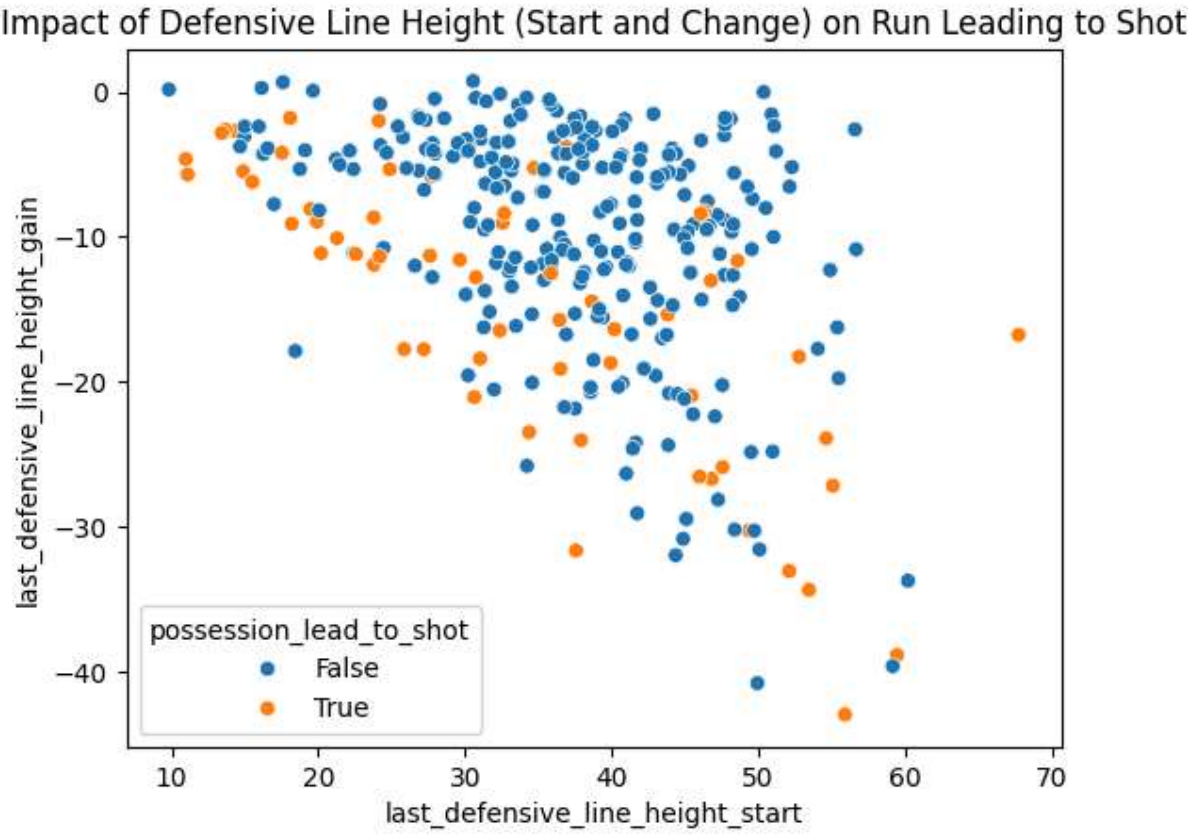


Here we are able to see that the runs that lead to shots tend to be more diagonal towards the center of the field (This matches our earlier analysis done on the change in width of runs) while runs that don't lead to shots tend to be straight forward runs.

The Impact of the Defensive Line

The defense has a huge role in how a forward makes their run so in the next couple sections we will take a look at how the relationship between the player making the run in behind and the defensive line can be utilized to identify the differences in the distributions of the runs leading to shots or not.

In [18]:
```
sns.scatterplot(data = runs_behind,x="last_defensive_line_height_start",y="last_def
plt.title("Impact of Defensive Line Height (Start and Change) on Run Leading to Sho
plt.show()
```



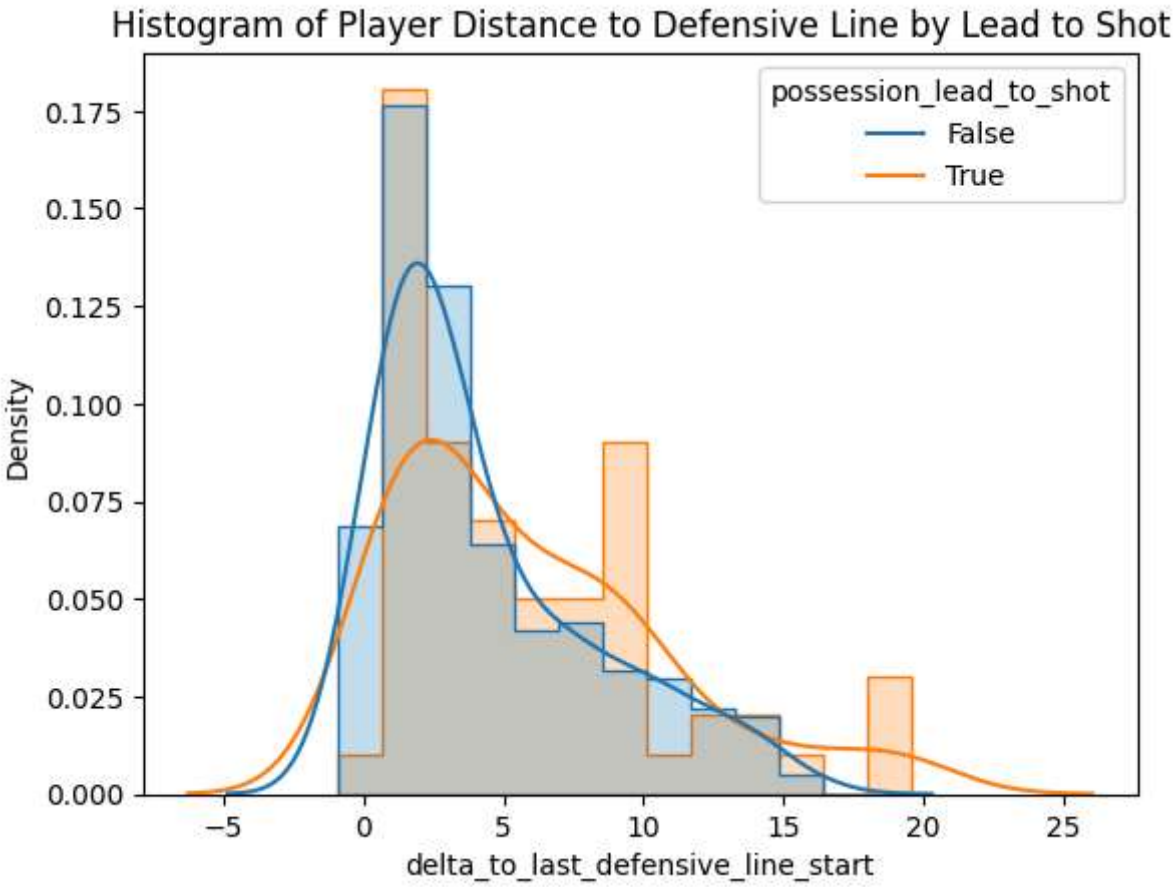
Here we are able to a very important distinction between the runs that lead to shots and the runs that don't. The runs that have a large disruption on the opponent's defense (The change in height of the defensive line) in relation to the start height of the defensive line tend to lead to more shots. Simply, if the defensive line is not very high but the run is still able to push the line back it leads to more shots and if the defensive line starts very high and the run forces the defensive line to drop off a lot this also leads to more shots.

What about the runner's distance to the defensive line?

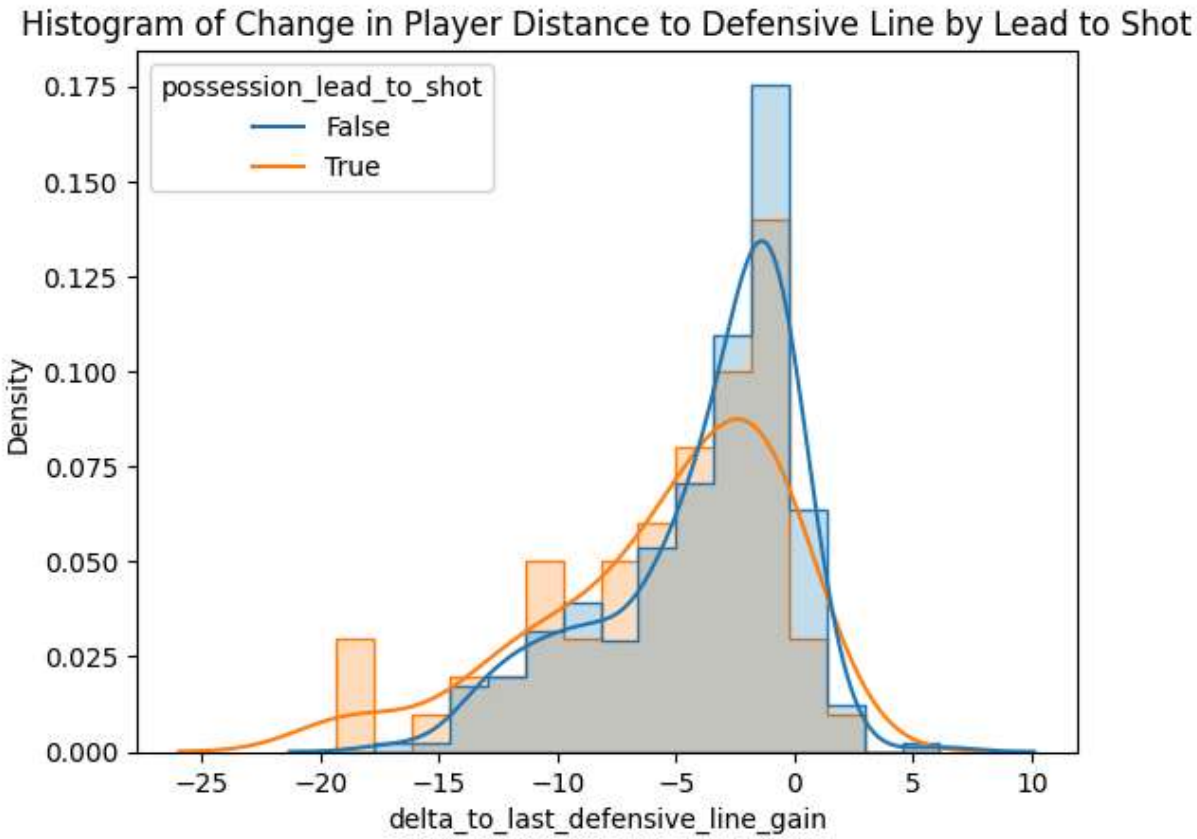
We just analyzed how the the run impacts the defensive line itself but what about the runner's relationship to the defensive line. What impact does being closer to the defensive line have when making a run?

In [19]:
```
sns.histplot(data=runs_behind, x='delta_to_last_defensive_line_start', hue='possess
sns.kdeplot(data=runs_behind, x='delta_to_last_defensive_line_start', hue='possessi
plt.title('Histogram of Player Distance to Defensive Line by Lead to Shot')
plt.show()

dist_shot = runs_behind.loc[runs_behind['possession_lead_to_shot'], 'delta_to_last_
dist_no_shot = runs_behind.loc[~runs_behind['possession_lead_to_shot'], 'delta_to_l

u_stat, p_val_u = mannwhitneyu(dist_shot, dist_no_shot, alternative='greater')
print(f"Mann-Whitney U test for Player Distance to Defensive Line: U-statistic = {u

sns.histplot(data=runs_behind, x='delta_to_last_defensive_line_gain', hue='possessi
sns.kdeplot(data=runs_behind, x='delta_to_last_defensive_line_gain', hue='possessio
```

```
plt.title('Histogram of Change in Player Distance to Defensive Line by Lead to Shot
plt.show()

dist_shot = runs_behind.loc[runs_behind['possession_lead_to_shot'], 'delta_to_last_
dist_no_shot = runs_behind.loc[~runs_behind['possession_lead_to_shot'], 'delta_to_l

u_stat, p_val_u = mannwhitneyu(dist_shot, dist_no_shot, alternative='less')
print(f"Mann-Whitney U test for Change in Player Distance to Defensive Line: U-stat
```




```
Mann-Whitney U test for Player Distance to Defensive Line: U-statistic = 9534.5, p-v
alue = 0.0166
```

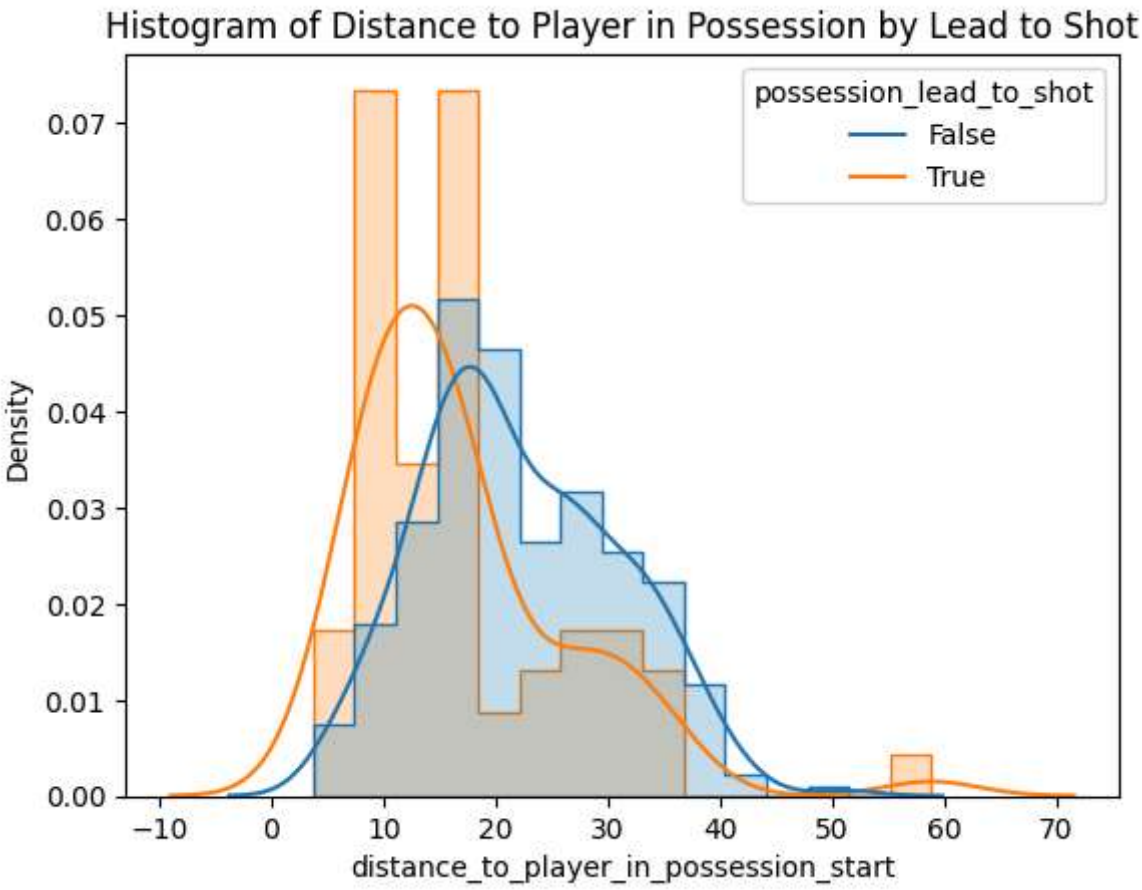



```
Mann-Whitney U test for Change in Player Distance to Defensive Line: U-statistic = 6
656.5, p-value = 0.013
```

Here we see another two distinctions in the runner's relationship to the defensive line. First, when a player starts further away from the defensive line (not as far forward) these runs tend to result more in shots. Second, when the runner is able to decrease the distance between them and the defensive line it tends to lead to more shots. This goes to show that actively attacking a, currently not threatened, defensive line with a run in behind will lead to more shots.

What impact does the distance of the run to the player with the ball have?

In [20]:
```
sns.histplot(data=runs_behind, x='distance_to_player_in_possession_start', hue='pos
sns.kdeplot(data=runs_behind, x='distance_to_player_in_possession_start', hue='poss
plt.title('Histogram of Distance to Player in Possession by Lead to Shot')
plt.show()

dist_shot = runs_behind.loc[runs_behind['possession_lead_to_shot'], 'distance_to_pl
dist_no_shot = runs_behind.loc[~runs_behind['possession_lead_to_shot'], 'distance_t

u_stat, p_val_u = mannwhitneyu(dist_shot, dist_no_shot, alternative='less')
print(f"Mann-Whitney U test for Distance to Player in Possession: U-statistic = {u_
```




```
Mann-Whitney U test for Distance to Player in Possession: U-statistic = 5003.0, p-va
lue = 1.12e-06
```

Here we see that there is a strong relationship between the player in possession at the start of a run and the player making the run. Runs that lead to shots tend to be closer to the player in possession than runs that don't lead to shots. These runs that are closer are more of a potential threat because it is easier for the player in possession to get the player making the run the ball and even if the runner doesn't receive the ball the runner will be able to disrupt and occupy the defense more than if the run was further from the ball.

Discusison and Conclusion

Off-the-ball runs in behind play a critical role in soccer, they possess both a clear potential attacking threat as well as a disrupting factor to the defense. First, we were able to conclude that possessions that contained runs in behind had a significantly higher (14%) probability of leading to a shot. This analysis was made possible by the data collection team at SkillCorner who provided the spatial temporal tracking dataset. SkillCorner was able to use algorithms to first detect and find off ball runs and categorize them. Using the the SKillCorner datasets we were able to combine the dataset of spatial temporal data (player and ball locations) with an event level dataset (list of off-ball-runs) to compute features for each run that are essential to our analysis. These features include but are not limited to: x, y, angle, speed, curve, broke the line, opponents passed, etc. We were then able to find all the possessions that occured within the games and determine wheter they resulted in a shot or not and

create a possession index to map possessions to runs in possessions. This allowed us to build our dataset for analysis.

The second biggest part of our analysis was determining the features of the runs in behind that are significant predictors of whether the possession will lead to a shot. Using 1 variable logistic regression analysis we were able to simplify our list of features into a smaller list of significant features. These included: X end of run, distance covered, width at end of run and change in width during run, the angle of the run, defensive line height and runner's relation to defensive line, and finally distance from player with the ball to the runner. One very important thing to note is that soccer and off balls runs can't be purely broken down linearly and that there are so many iteractions and correlations between are features that these simple models may not be able to capture. This was demonstrated when we performed correlation analysis and determined that quite a few of these significant features were highly correlated to each other. However, nonetheless this was something that was expected as some of the features such as change in defensive line height and distance covered by run are expected to be pretty highly correlated due to how plays unfold. With all this being said we can still take a look at how these simple linear models are able to capture all of these complex relationships.

Not only were we able to determine the significant features in runs behind we also were able to make some general assumptions on how these feataures affected a possession being involed in a shot and therefore the run being a success. By splitting our dataset of runs into runs that lead to shot and runs that didn't lead to shot we were able to plot the two distributions seperately to identify the patterns. From this we were able to gather 6 critical points about runs in behind:

1. Runs that start further up the field and that end at the top of the box lead to more shots. This feels pretty intuitive as the main goal is to get the ball closer to goal and running into that space to receive the ball can help accomlish this task.
2. Runs that cover more distance tend to lead to more shots. This goes along with the idea of how dangerous counterattacks are in soccer. If a run is longer there is more of a chance that this run is a part of a counter attack that is more dangerous.
3. Runs that have a large decrease in width (i.e. they go towards the center) and end in the center of the field have a higher chance of leading to a shot than runs that start wide and stay wide.
4. Runs that are diagonal towards the center of the field lead to more shots than veritcal straight runs. This also goes hand in hand with the previous point and backs up the notion of why striker should make diagonal runs.
5. Runs that are able to force a lot of disruption (Change in defensive line) with relation to where the defensive line starts are able to produce more shots. If the line starts low when run is made but still forces the line to move a bit then it leads to more shots. Also, if the defensive line starts a lot higher and the run forces the defense to drop off a lot this also results in a lot more shots.

6. Last and most significant result is that runs that are made closer to the player in possession lead to more shots than runs that are made farther from the player with the ball. This signifies that the run should be an easy passing option for the player on the ball as this will pose more of a threat for the defense.

Some limitaitons of this approach is that many of these variables interweave with each other in ways linear models will not be able to pick up. However, these basic notions of what successful runs in behind look like can help to back up how strikers/wingers should approach off ball runs. This can help lay the groundwork for youth academies and training as players learn these aspects of the sport.