Defenders Beat by Pass Statistic

By Will Sivolella

Project Description

I utilized StatsBomb open-source data and StatsBomb API to create a statistic that measured how many defenders are beat by a player's completed forward pass. I decided to use 2022 World Cup data because event data and event360 data were provided for every match and because it was a relatively recent competition. The new statistic along with other passing metrics provide an insight on the most effective and direct passers of the ball during the 2022 World Cup.

Data Exploration

First, download the StatsBomb open-data and stored it in a local directory.

```
In [1]:  import pandas as pd
         import numpy as np
         import math
         from statsbombpy import sb
         from mplsoccer import Pitch
         from IPython.display import display, HTML
```

Exploring competitions data set and sorting by recency

```
In [2]:  comps = sb.competitions()
```

```
In [3]:  comps.sort_values(by = "season_name", ascending = False).reset_index(drop = True).head
```

Out[3]:

	competition_id	season_id	country_name	competition_name	competition_gender	competition_yout
0	72	107	International	Women's World Cup	female	Fals
1	44	107	United States of America	Major League Soccer	male	Fals
2	7	235	France	Ligue 1	male	Fals
3	53	106	Europe	UEFA Women's Euro	female	Fals
4	43	106	International	FIFA World Cup	male	Fals
5	7	108	France	Ligue 1	male	Fals
6	1238	108	India	Indian Super league	male	Fals
7	11	90	Spain	La Liga	male	Fals
8	37	90	England	FA Women's Super League	female	Fals
9	55	43	Europe	UEFA Euro	male	Fals

◀ ▬▬▬▬▬▬▬▬▬ ▶

Exploring World Cup Matches dataset

In [4]:
```python
matches_df = sb.matches(competition_id = 43, season_id = 106).sort_values(by = "match_
```

In [5]:
```python
matches_df.head(10)
```

Out[5]:

	match_id	match_date	kick_off	competition	season	home_team	away_team	home_score	aw
0	3869685	2022-12-18	17:00:00.000	International - FIFA World Cup	2022	Argentina	France	3	
1	3869684	2022-12-17	17:00:00.000	International - FIFA World Cup	2022	Croatia	Morocco	2	
2	3869552	2022-12-14	21:00:00.000	International - FIFA World Cup	2022	France	Morocco	2	
3	3869519	2022-12-13	21:00:00.000	International - FIFA World Cup	2022	Argentina	Croatia	3	
4	3869486	2022-12-10	17:00:00.000	International - FIFA World Cup	2022	Morocco	Portugal	1	
5	3869354	2022-12-10	21:00:00.000	International - FIFA World Cup	2022	England	France	1	
6	3869321	2022-12-09	21:00:00.000	International - FIFA World Cup	2022	Netherlands	Argentina	2	
7	3869420	2022-12-09	17:00:00.000	International - FIFA World Cup	2022	Croatia	Brazil	1	
8	3869220	2022-12-06	17:00:00.000	International - FIFA World Cup	2022	Morocco	Spain	0	
9	3869254	2022-12-06	21:00:00.000	International - FIFA World Cup	2022	Portugal	Switzerland	6	

10 rows × 22 columns

◀ ▬▬▬▬▬▬▬▬▬ ▶

In [6]: `len(matches_df)`

Out[6]: 64

As shown above, all data for 64 2022 World Cup matches are provided.

Creating list of match_ids to get corresponding event and event360 data for each match.

In [7]: `match_id_list = list(matches_df.match_id)`

Retrieving event and event360 data corresponding to the first match in the matches data frame, which is the World Cup Final.

In [8]:
```python
events_df = sb.events(match_id = match_id_list[0])

#Change path to you local directory
events360_df = pd.read_json(f'C:/Users/wills/Documents/StatbombProject/open-data-maste
```

In [9]:
```python
events_df.columns
```

Out[9]:
```
Index(['50_50', 'bad_behaviour_card', 'ball_receipt_outcome',
       'ball_recovery_offensive', 'ball_recovery_recovery_failure',
       'block_deflection', 'block_offensive', 'carry_end_location',
       'clearance_aerial_won', 'clearance_body_part', 'clearance_head',
       'clearance_left_foot', 'clearance_other', 'clearance_right_foot',
       'counterpress', 'dribble_nutmeg', 'dribble_outcome', 'dribble_overrun',
       'duel_outcome', 'duel_type', 'duration', 'foul_committed_advantage',
       'foul_committed_card', 'foul_committed_offensive',
       'foul_committed_penalty', 'foul_committed_type', 'foul_won_advantage',
       'foul_won_defensive', 'foul_won_penalty', 'goalkeeper_body_part',
       'goalkeeper_end_location', 'goalkeeper_outcome', 'goalkeeper_position',
       'goalkeeper_technique', 'goalkeeper_type', 'id', 'index',
       'interception_outcome', 'location', 'match_id', 'minute', 'off_camera',
       'out', 'pass_aerial_won', 'pass_angle', 'pass_assisted_shot_id',
       'pass_body_part', 'pass_cross', 'pass_deflected', 'pass_end_location',
       'pass_goal_assist', 'pass_height', 'pass_inswinging', 'pass_length',
       'pass_outcome', 'pass_outswinging', 'pass_recipient',
       'pass_recipient_id', 'pass_shot_assist', 'pass_switch',
       'pass_technique', 'pass_through_ball', 'pass_type', 'period',
       'play_pattern', 'player', 'player_id', 'position', 'possession',
       'possession_team', 'possession_team_id', 'related_events', 'second',
       'shot_aerial_won', 'shot_body_part', 'shot_end_location',
       'shot_first_time', 'shot_freeze_frame', 'shot_key_pass_id',
       'shot_one_on_one', 'shot_outcome', 'shot_statsbomb_xg',
       'shot_technique', 'shot_type', 'substitution_outcome',
       'substitution_replacement', 'tactics', 'team', 'team_id', 'timestamp',
       'type', 'under_pressure'],
      dtype='object')
```

There is a lot of great event descriptions here, and I will isolate the columns relevant to passing later.

In [10]:
```python
events360_df.head()
```

	event_uuid	visible_area	freeze_frame
0	f651a6c4-55e3-4e0f-a178-59414ba83d6a	[8.98496759714251, 80.0, 41.4622037211361, 0.0...	[{'teammate': True, 'actor': False, 'keeper': ...
1	97b5dc82-547a-4f93-a632-a2a8daf5ac98	[8.74654439366353, 80.0, 40.9967310870193, 0.0...	[{'teammate': True, 'actor': False, 'keeper': ...
2	9707e590-1d55-4d25-a50f-475019356152	[8.74654439366353, 80.0, 40.9967310870193, 0.0...	[{'teammate': True, 'actor': False, 'keeper': ...
3	a5956ecc-d637-44d2-8705-e2079e12013e	[10.227276315524, 80.0, 41.5155332886657, 0.0,...	[{'teammate': True, 'actor': False, 'keeper': ...
4	5ba6dfab-49f0-4225-bbb6-8b1e27bc729a	[13.2232254880239, 80.0, 0.0, 65.4802459369524...	[{'teammate': True, 'actor': False, 'keeper': ...

The most relevant data from the event360 data frame is the freeze_frame column containing player location data for each event.

Data Cleaning

Now that I have an understanding of the data, I want to create data frames that will best allow me to formulate the defenders beat by pass statistic as well as showcase other relevant passing statistics. The first step is to merge event data with the event360 data in order to put all the information into one data frame.

```
In [11]:  events_merged_df = pd.merge(left = events_df, right = events360_df, left_on='id', righ
```

Repeat process so that the event data is grouped with the corresponding event360 data for each match and combine all matches into single data frame.

```
In [12]:  for i in range(1, len(match_id_list)):
              events_df = sb.events(match_id = match_id_list[i])

              #Change path to your local directory
              events360_df = pd.read_json(f'C:/Users/wills/Documents/StatbombProject/open-data-m
              events_merged_df_temp = pd.merge(left = events_df, right = events360_df, left_on='
              events_merged_df = pd.concat([events_merged_df, events_merged_df_temp], axis=0)

          events_merged_df = events_merged_df.reset_index(drop = True)
```

Filter data frame to only get pass and ball receipt instances. Ball receipt instances are needed to get player location data at the time a pass is received.

```
In [13]:  passes = events_merged_df.query('type == "Pass"').reset_index(drop = True)
          ball_receipts = events_merged_df.query('type == "Ball Receipt*"').reset_index(drop = T
```

Only include relevant columns for passes data frame (best to be generous with column inclusion).

```
In [14]: passes = passes[['timestamp', 'id', 'related_events', 'player_id', 'player', 'position'
                          'team', 'type', 'location', 'pass_end_location', 'pass_angle', 'pa
                          'ball_receipt_outcome', 'under_pressure', 'visible_area', 'freeze_
```

Only want completed forward passes to generate statistic.

```
In [15]: passes_complete = passes[passes['pass_outcome'].isna()]

         low_bound = -1*(math.pi/2)
         up_bound = math.pi/2

         passes_clean = passes_complete.query('@low_bound < pass_angle < @up_bound').reset_inde
```

```
In [16]: passes_clean.head()
```

Out[16]:

	timestamp	id	related_events	player_id	player	position	pass_recipient	team
0	00:00:16.423	5ba6dfab-49f0-4225-bbb6-8b1e27bc729a	[2be9a9f2-84ab-4935-8fb2-bd497202fa2f]	29201.0	Nahuel Molina Lucero	Right Back	Rodrigo Javier De Paul	Argentina
1	00:00:43.285	4fd86774-8f37-4ea2-b562-b1a43c5adedf	[b2e3ba63-98d2-4e2e-b456-3b9af203f6b2]	20572.0	Cristian Gabriel Romero	Right Center Back	Nicolás Hernán Otamendi	Argentina
2	00:00:45.601	4a4fd288-9139-4987-9c2a-3ab0e1793310	[f6c30bac-9533-4dd1-bff7-95900d699747]	3090.0	Nicolás Hernán Otamendi	Left Center Back	Nicolás Alejandro Tagliafico	Argentina
3	00:00:47.599	bdf6b02c-1c04-4777-9617-bf73ececbf6c	[585421b8-3acc-4627-8b3c-74496fd20d1d]	5507.0	Nicolás Alejandro Tagliafico	Left Back	Alexis Mac Allister	Argentina
4	00:01:31.390	1af94a8d-cb50-4d93-96e1-19fa1ae154fb	[caa393a7-8230-44aa-8b37-ad3929626929]	6704.0	Theo Bernard François Hernández	Left Back	Adrien Rabiot	France

Show only relevant columns in ball receipts dataframe (again generous with inclusion of column names).

```
In [17]: ball_receipts = ball_receipts[['id', 'related_events', 'player_id', 'player', 'positic
                          'team', 'type', 'ball_receipt_outcome', 'under_pressure
                          'freeze_frame']]
```

```
In [18]: ball_receipts.head()
```

Out[18]:

	id	related_events	player_id	player	position	team	type	ball_receipt_outo
0	97b5dc82-547a-4f93-a632-a2a8daf5ac98	[f651a6c4-55e3-4e0f-a178-59414ba83d6a]	10481.0	Aurélien Djani Tchouaméni	Right Defensive Midfield	France	Ball Receipt*	
1	810629a9-ac49-494a-b859-89ef3ff36c57	[a5956ecc-d637-44d2-8705-e2079e12013e]	3604.0	Olivier Giroud	Center Forward	France	Ball Receipt*	Incom
2	2be9a9f2-84ab-4935-8fb2-bd497202fa2f	[54bd94ec-86b7-4428-b71b-286c1e080b9f, 5ba6dfa...	7797.0	Rodrigo Javier De Paul	Right Center Midfield	Argentina	Ball Receipt*	
3	22dde0b6-5149-411e-a910-68f6aa5b7523	[27829865-a43c-4c50-b545-b09e8a965e16]	20572.0	Cristian Gabriel Romero	Right Center Back	Argentina	Ball Receipt*	
4	29f8db49-f71f-4177-9825-3e93697fa4ba	[59420b0c-94cc-4fb8-86fb-e1105936f6eb]	3090.0	Nicolás Hernán Otamendi	Left Center Back	Argentina	Ball Receipt*	

◄ ━━━━━━━━━━━━━━━━━━━ ►

Merge data frames so that so passes and corresponding ball receipts are on same row.

In [19]:
```python
passes_clean = passes_clean.explode('related_events').reset_index(drop = True)
combined_df = pd.merge(passes_clean, ball_receipts, left_on='related_events', right_or
```

In [20]:
```python
lost_in_merge = len(passes_clean) - len(combined_df)
percent_retained_in_merge = 100-round(((lost_in_merge)/len(passes_clean))*100,2)
percent_retained_in_merge
```

Out[20]: 88.2

88.2% of the completed forward passes for every match throughout the 2022 World Cup has a corresponding ball receipt event.

Filter by if freeze_frame location data is available for ball receipt instances.

In [21]:
```python
combined_df_clean = combined_df[combined_df['freeze_frame_y'].notna()].reset_index(drc
```

In [22]:
```python
no_freeze_frame = len(combined_df) - len(combined_df_clean)
percent_retained = 100-round(((len(passes_clean) - len(combined_df_clean))/len(passes_
percent_retained
```

Out[22]: 79.25

79.25% of the completed forward passes for every match throughout the 2022 World Cup has a corresponding ball receipt and location data at the time the pass is received.

In [23]: `combined_df_clean.head()`

Out[23]:

	timestamp	id_x	related_events_x	player_id_x	player_x	position_x	pass_recipient	t
0	00:00:16.423	5ba6dfab-49f0-4225-bbb6-8b1e27bc729a	2be9a9f2-84ab-4935-8fb2-bd497202fa2f	29201.0	Nahuel Molina Lucero	Right Back	Rodrigo Javier De Paul	Arg
1	00:00:43.285	4fd86774-8f37-4ea2-b562-b1a43c5adedf	b2e3ba63-98d2-4e2e-b456-3b9af203f6b2	20572.0	Cristian Gabriel Romero	Right Center Back	Nicolás Hernán Otamendi	Arg
2	00:00:45.601	4a4fd288-9139-4987-9c2a-3ab0e1793310	f6c30bac-9533-4dd1-bff7-95900d699747	3090.0	Nicolás Hernán Otamendi	Left Center Back	Nicolás Alejandro Tagliafico	Arg
3	00:00:47.599	bdf6b02c-1c04-4777-9617-bf73ececbf6c	585421b8-3acc-4627-8b3c-74496fd20d1d	5507.0	Nicolás Alejandro Tagliafico	Left Back	Alexis Mac Allister	Arg
4	00:01:31.390	1af94a8d-cb50-4d93-96e1-19fa1ae154fb	caa393a7-8230-44aa-8b37-ad3929626929	6704.0	Theo Bernard François Hernández	Left Back	Adrien Rabiot	

5 rows × 28 columns

Generating Statistics

Calculating Defenders Beat by Pass Statistic

In [24]:
```python
pass_defenders_beat = []

for i in range(len(combined_df_clean)):
    x = i
    y = x+1

    inst = combined_df_clean[x:y]

    x_start = inst.location[x][0]
    x_end = inst.pass_end_location[x][0]

    count = 0

    for i in inst.iloc[0]['freeze_frame_y']:
        if i['teammate'] == False:
            if x_start < i['location'][0] < x_end:
                count += 1
    pass_defenders_beat.append(count)
```

Add stat to data frame

```
In [25]: combined_df_clean['pass_defenders_beat'] = pass_defenders_beat
```

```
In [26]: combined_df_clean.head()
```

Out[26]:

	timestamp	id_x	related_events_x	player_id_x	player_x	position_x	pass_recipient	t
0	00:00:16.423	5ba6dfab-49f0-4225-bbb6-8b1e27bc729a	2be9a9f2-84ab-4935-8fb2-bd497202fa2f	29201.0	Nahuel Molina Lucero	Right Back	Rodrigo Javier De Paul	Arg
1	00:00:43.285	4fd86774-8f37-4ea2-b562-b1a43c5adedf	b2e3ba63-98d2-4e2e-b456-3b9af203f6b2	20572.0	Cristian Gabriel Romero	Right Center Back	Nicolás Hernán Otamendi	Arg
2	00:00:45.601	4a4fd288-9139-4987-9c2a-3ab0e1793310	f6c30bac-9533-4dd1-bff7-95900d699747	3090.0	Nicolás Hernán Otamendi	Left Center Back	Nicolás Alejandro Tagliafico	Arg
3	00:00:47.599	bdf6b02c-1c04-4777-9617-bf73ececbf6c	585421b8-3acc-4627-8b3c-74496fd20d1d	5507.0	Nicolás Alejandro Tagliafico	Left Back	Alexis Mac Allister	Arg
4	00:01:31.390	1af94a8d-cb50-4d93-96e1-19fa1ae154fb	caa393a7-8230-44aa-8b37-ad3929626929	6704.0	Theo Bernard François Hernández	Left Back	Adrien Rabiot	

5 rows × 29 columns

◀ ▬▬▬▬▬ ▶

Explaination of Defenders Beat by Pass Statistic

The defenders beat by pass statistic measures how many defenders are between the location of where the ball was played and the location of where the ball was successfully received by a teammate at the time when the ball was received. The aspect of location that is considered for the calculation of this statistic is the x-coordinate, the goal-to-goal direction of the field. I only chose the x-direction because passing in the y-direction or side-to-side does not inherently advance the ball and take defenders temporarily out of the play. I also chose to measure the position of players at the time the pass was received instead of when the pass was played to account for movement of defenders while the pass is traveling.

```
In [27]: p = Pitch(pitch_type = 'statsbomb')
         fig, ax = p.draw(figsize = (12, 8))

         x = 24
         y = x+1
```
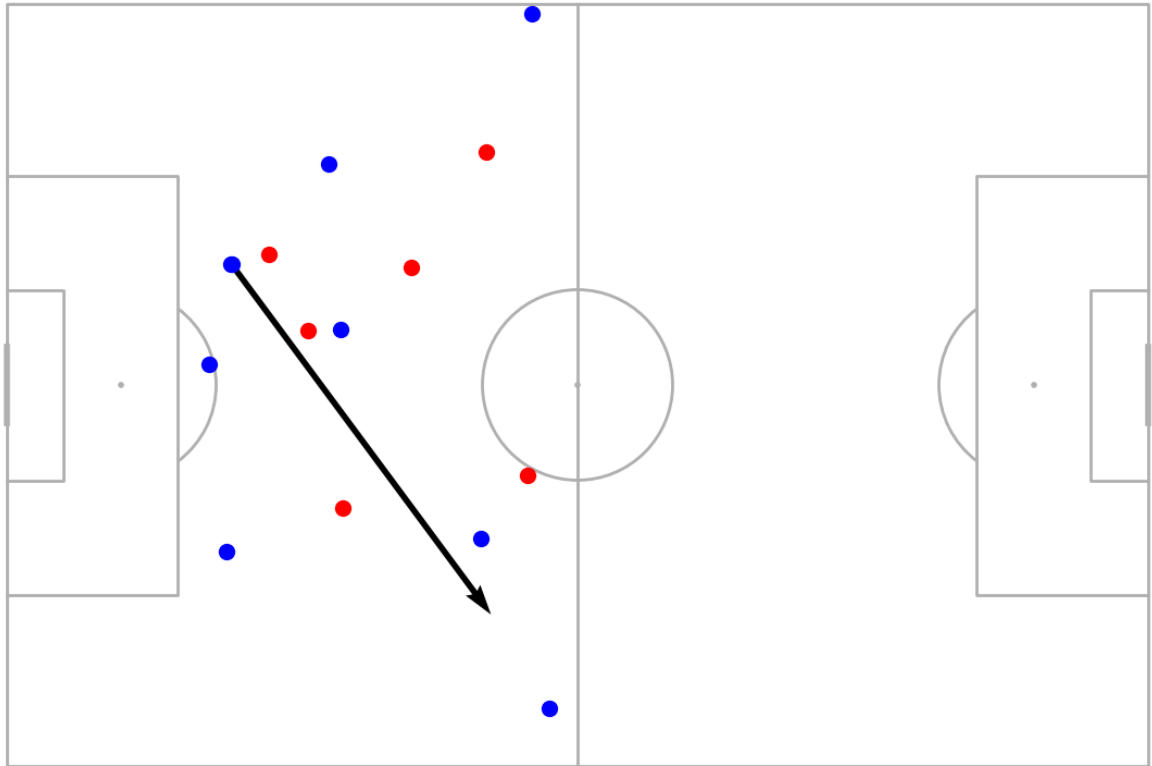
```python
inst = combined_df_clean[x:y]

p.scatter(x = inst.location[x][0], y = inst.location[x][1], ax=ax, c = 'blue', s = 100
p.arrows(xstart = inst.location[x][0], ystart = inst.location[x][1], xend = inst.pass_

for i in inst.iloc[0]['freeze_frame_x']:
    if i['teammate']:
        color = 'blue'
    else:
        color = 'red'
    p.scatter(x=i['location'][0], y=i['location'][1], ax=ax, c = color, s = 100)
```


The plot above is a single pass instance from the 2022 World Cup Final. It shows the player passing the pass, the pass vector, his teammates (blue) and his opponents (red) at the time when the pass is played.

In [28]:
```python
#Defenders beaten code.

x = 24
y = x+1

inst = combined_df_clean[x:y]

x_start = inst.location[x][0]
x_end = inst.pass_end_location[x][0]

count = 0

for i in inst.iloc[0]['freeze_frame_x']:
    if i['teammate'] == False:
        if x_start < i['location'][0] < x_end:
```

```
            count += 1
print(count)
```

5

If I were to use the location data at the time the pass is played to calculate the statistic, 5 defenders would be beated by that pass in the x-direction (as shown by the code output and plot)
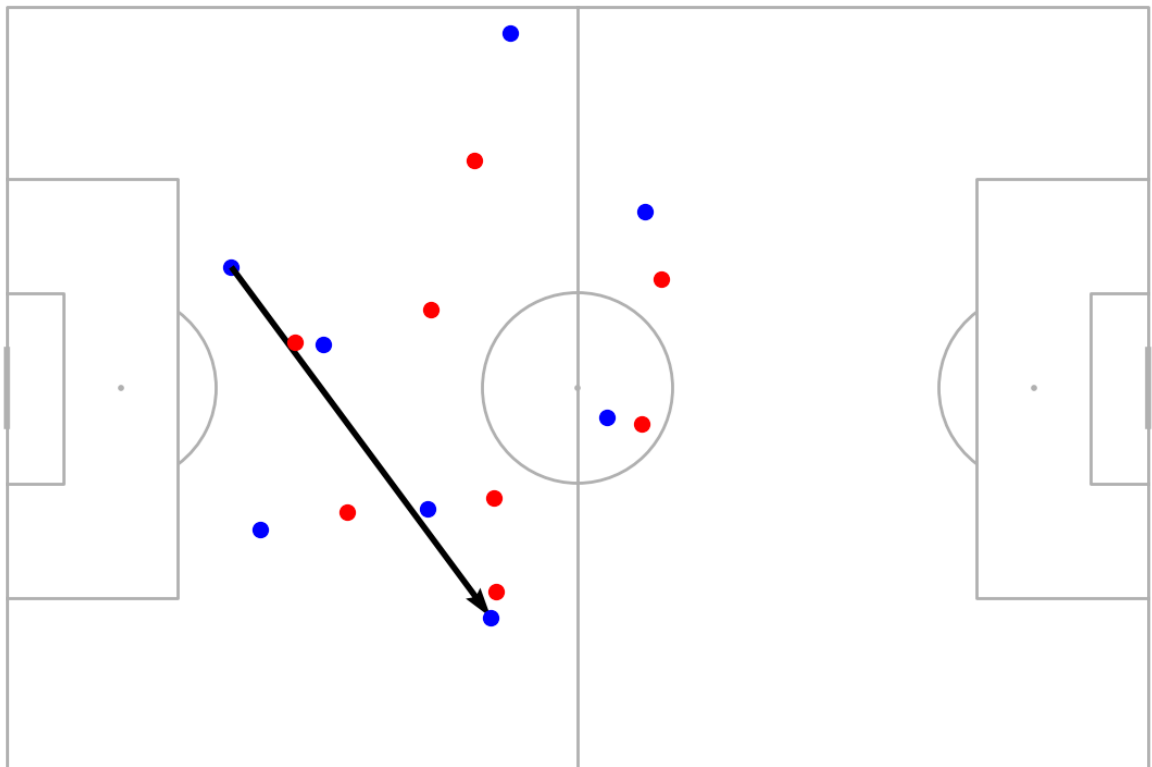
In [29]:
```
p = Pitch(pitch_type = 'statsbomb')
fig, ax = p.draw(figsize = (12, 8))

x = 24
y = x+1

inst = combined_df_clean[x:y]

p.scatter(x = inst.location[x][0], y = inst.location[x][1], ax=ax, c = 'blue', s = 100
p.arrows(xstart = inst.location[x][0], ystart = inst.location[x][1], xend = inst.pass_

for i in inst.iloc[0]['freeze_frame_y']:
    if i['teammate']:
        color = 'blue'
    else:
        color = 'red'
    p.scatter(x=i['location'][0], y=i['location'][1], ax=ax, c = color, s = 100)
```


The plot above uses the location data at the time the ball is received to account for the recovery of the defenders while the pass is traveling.

```
In [30]:  inst['pass_defenders_beat'][24]
```

Out[30]: 4

When accounting for the recovery of defenders while the pass is traveling, the actual amount of defenders beat in the x-direction is 4.

Get Individual Player Stats from Event Data

```
In [31]:  combined_df_clean
```

```
In [30]:  inst['pass_defenders_beat'][24]
```

Out[30]: 4

Out[31]:

	timestamp	id_x	related_events_x	player_id_x	player_x	position_x	pass_recipient
0	00:00:16.423	5ba6dfab-49f0-4225-bbb6-8b1e27bc729a	2be9a9f2-84ab-4935-8fb2-bd497202fa2f	29201.0	Nahuel Molina Lucero	Right Back	Rodrigo Javier De Paul
1	00:00:43.285	4fd86774-8f37-4ea2-b562-b1a43c5adedf	b2e3ba63-98d2-4e2e-b456-3b9af203f6b2	20572.0	Cristian Gabriel Romero	Right Center Back	Nicolás Hernán Otamendi
2	00:00:45.601	4a4fd288-9139-4987-9c2a-3ab0e1793310	f6c30bac-9533-4dd1-bff7-95900d699747	3090.0	Nicolás Hernán Otamendi	Left Center Back	Nicolás Alejandro Tagliafico
3	00:00:47.599	bdf6b02c-1c04-4777-9617-bf73ececbf6c	585421b8-3acc-4627-8b3c-74496fd20d1d	5507.0	Nicolás Alejandro Tagliafico	Left Back	Alexis Mac Allister
4	00:01:31.390	1af94a8d-cb50-4d93-96e1-19fa1ae154fb	caa393a7-8230-44aa-8b37-ad3929626929	6704.0	Theo Bernard François Hernández	Left Back	Adrien Rabiot
...	...	...	...	...	...	...	...
28516	00:48:42.009	428c82f6-39be-4c05-82de-0e7241afb48a	df972265-2cd3-4bbe-92c6-6a16551e6780	40211.0	Jeremy Leonel Sarmiento Morante	Left Wing	Gonzalo Jordy Plata Jiménez
28517	00:48:43.740	8b7325d3-f505-4100-b078-98219812b5e1	1fb696a7-7dd8-473a-893d-c12a3f35b03d	31152.0	Gonzalo Jordy Plata Jiménez	Right Wing	Pervis Josué Estupiñán Tenorio
28518	00:48:46.244	c4c656db-dfbc-4ac6-a654-951fa8aef19d	dba0cd00-e592-4b47-9332-7a88a2bcbb5c	24085.0	Pervis Josué Estupiñán Tenorio	Left Back	Gonzalo Jordy Plata Jiménez
28519	00:49:40.105	c44934a0-a37d-4ce1-ad8e-ce899a784174	df9a32c1-e770-42b3-bea8-c76c42e76d7d	124496.0	Saad Abdullah Al Sheeb	Goalkeeper	Abdulaziz Hatem Mohammed Abdullah
28520	00:49:45.572	6043a0d2-b2db-46d9-b7e2-c4aff2d4a613	8314f079-d9f0-4cd3-b480-ddd4615ca5cc	124493.0	Boualem Khoukhi	Center Back	Abdelkarim Hassan Al Haj Fadlalla

28521 rows × 29 columns

In [32]:
```
result = combined_df_clean.groupby(['player_x', 'team_x']).agg(Total_Defenders_Beat=('
         Forward_Completed_Passes=('player_x', 'count')).reset_index()
```

```
result['Avg_Defenders_Beat_per_Pass'] = round(result['Total_Defenders_Beat']/result['F
sorted_result = result.sort_values(by='Total_Defenders_Beat', ascending=False).reset_i
```

Results

In [33]:
```
sorted_result.rename(columns={'player_x':'Player', 'team_x':'Team'}, inplace=True)
```

In [34]:
```
display(HTML('<h1 style="text-align: center;">Top 25 Players with Highest Total Defenc
display(sorted_result.head(25))
```

Top 25 Players with Highest Total Defenders Beat by Passes

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_per
0	Luka Modrić	Croatia	706	267	
1	Rodrigo Hernández Cascante	Spain	583	380	
2	Pedro González López	Spain	549	198	
3	Lionel Andrés Messi Cuccittini	Argentina	526	152	
4	Marcelo Brozović	Croatia	471	244	
5	Joško Gvardiol	Croatia	463	311	
6	Nicolás Hernán Otamendi	Argentina	454	326	
7	Rodrigo Javier De Paul	Argentina	443	173	
8	Enzo Fernandez	Argentina	435	227	
9	John Stones	England	432	281	
10	Achraf Hakimi Mouh	Morocco	431	168	
11	Aurélien Djani Tchouaméni	France	424	218	
12	Mateo Kovačić	Croatia	418	175	
13	Aymeric Laporte	Spain	391	287	
14	Luke Shaw	England	390	147	
15	Josip Juranović	Croatia	386	146	
16	Harry Maguire	England	372	219	
17	Frenkie de Jong	Netherlands	368	155	
18	Nathan Aké	Netherlands	361	200	

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_per
19	Cristian Gabriel Romero	Argentina	361	205	
20	Jordi Alba Ramos	Spain	353	127	
21	Éder Gabriel Militão	Brazil	346	132	
22	Antoine Griezmann	France	332	102	
23	Jules Koundé	France	329	121	
24	Theo Bernard François	France	328	108	

```
In [35]:  sorted_result_avg = sorted_result.sort_values(by='Avg_Defenders_Beat_per_Pass', ascend
          sorted_result_avg = sorted_result_avg.query('Forward_Completed_Passes >= 100').reset_i
```

```
In [36]:  header = '<h1 style="text-align: center;">Top 25 Players with Highest Avg. Defenders B
          subheader = '<h3 style="text-align: center;">*minimum 100 completed forward passes</h3
          html_content = header + subheader
          display(HTML(html_content))
          display(sorted_result_avg.head(25))
```

Top 25 Players with Highest Avg. Defenders Beat per Completed Forward Pass

*minimum 100 completed forward passes

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_per_
0	Lionel Andrés Messi Cuccittini	Argentina	526	152	
1	Antoine Griezmann	France	332	102	
2	Theo Bernard François Hernández	France	328	108	
3	Daley Blind	Netherlands	326	114	
4	Borna Sosa	Croatia	302	107	
5	Jordi Alba Ramos	Spain	353	127	
6	Pedro González López	Spain	549	198	
7	Jules Koundé	France	329	121	
8	Carlos Henrique Casimiro	Brazil	288	108	
9	Luke Shaw	England	390	147	
10	Luka Modrić	Croatia	706	267	
11	Josip Juranović	Croatia	386	146	
12	Éder Gabriel Militão	Brazil	346	132	
13	Achraf Hakimi Mouh	Morocco	431	168	
14	Rodrigo Javier De Paul	Argentina	443	173	
15	Sergio Busquets i Burgos	Spain	319	125	
16	Joshua Kimmich	Germany	268	105	
17	In-Beom Hwang	South Korea	258	104	

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_per_
18	Mateo Kovačić	Croatia	418	175	
19	Pierre-Emile Højbjerg	Denmark	237	100	
20	Frenkie de Jong	Netherlands	368	155	
21	Azzedine Ounahi	Morocco	233	100	
22	Nahuel Molina Lucero	Argentina	286	123	
23	Manuel Obafemi Akanji	Switzerland	311	137	
24	Adrien	France	275	131	

In [37]:
```python
sorted_result_avg2 = sorted_result.sort_values(by='Avg_Defenders_Beat_per_Pass', ascer
sorted_result_avg2 = sorted_result_avg2.query('Forward_Completed_Passes >= 50').reset_
```

In [39]:
```python
header = '<h1 style="text-align: center;">Top 25 Players with Highest Avg. Defenders B
subheader = '<h3 style="text-align: center;">*minimum 50 completed forward passes</h3>
html_content = header + subheader
display(HTML(html_content))
display(sorted_result_avg2.head(25))
```

Top 25 Players with Highest Avg. Defenders Beat per Completed Forward Pass

*minimum 50 completed forward passes

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_pe
0	Lionel Andrés Messi Cuccittini	Argentina	526	152	
1	Ngoran Suiru Fai Collins	Cameroon	194	57	
2	Weston McKennie	United States	212	65	
3	Antoine Griezmann	France	332	102	
4	Rasmus Nissen Kristensen	Denmark	196	61	
5	Yahia Attiyat allah	Morocco	163	51	
6	Abdulelah Al Amri	Saudi Arabia	183	58	
7	Steven Berghuis	Netherlands	163	52	
8	Kevin De Bruyne	Belgium	162	52	
9	Neymar da Silva Santos Junior	Brazil	213	69	
10	Silvan Widmer	Switzerland	194	63	
11	Antonee Robinson	United States	250	82	
12	Theo Bernard François Hernández	France	328	108	
13	Mohammed Kanoo	Saudi Arabia	176	58	
14	Bruno Miguel Borges Fernandes	Portugal	245	81	
15	Luis Gerardo Chávez Magallón	Mexico	162	55	
16	Kylian Mbappé Lottin	France	196	67	

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_pe
17	David Raum	Germany	160	55	
18	Ricardo Iván Rodríguez Araya	Switzerland	221	76	
19	Youssouf Sabaly	Senegal	223	77	
20	Celso Borges Mora	Costa Rica	175	61	
21	Daley Blind	Netherlands	326	114	
22	Alistair Johnston	Canada	200	70	
23	Hakim Ziyech	Morocco	268	94	
24	Federico Santiago Valverde	Uruguay	208	73	

```
In [40]:   display(HTML('<h1 style="text-align: center;">Top 15 Players with Highest Total Defend
           display(sorted_result.head(15))
```

Top 15 Players with Highest Total Defenders Beat by Passes

	Player	Team	Total_Defenders_Beat	Forward_Completed_Passes	Avg_Defenders_Beat_per_F
0	Luka Modrić	Croatia	706	267	2
1	Rodrigo Hernández Cascante	Spain	583	380	1
2	Pedro González López	Spain	549	198	2
3	Lionel Andrés Messi Cuccittini	Argentina	526	152	3
4	Marcelo Brozović	Croatia	471	244	1
5	Joško Gvardiol	Croatia	463	311	1
6	Nicolás Hernán Otamendi	Argentina	454	326	1
7	Rodrigo Javier De Paul	Argentina	443	173	2
8	Enzo Fernandez	Argentina	435	227	1
9	John Stones	England	432	281	1
10	Achraf Hakimi Mouh	Morocco	431	168	2
11	Aurélien Djani Tchouaméni	France	424	218	1
12	Mateo Kovačić	Croatia	418	175	2
13	Aymeric Laporte	Spain	391	287	1