

```
In [40]: import pandas as pd
import matplotlib.pyplot as plt
from mplsoccer import Pitch, VerticalPitch
```

```
In [41]: #Read in the match data CSV as df
df = pd.read_csv('F:/My Drive/Soccer Analytics Exports/Pass Network/valladolidA.csv')
```

```
In [42]: df
```

```
Out[42]:
```

	id	eventId	minute	second	teamId	x	y	period	type	c
0	2248226915	2	0	0.0	Barcelona	0.0	0.0	1	Start	Si
1	2248226919	2	0	0.0	Valladolid	0.0	0.0	1	Start	Si
2	2248226929	3	0	1.0	Barcelona	50.0	50.0	1	Pass	Si
3	2248226941	4	0	2.0	Barcelona	43.8	46.9	1	Pass	Si
4	2248226951	5	0	4.0	Barcelona	36.6	56.6	1	Pass	Si
...
1673	2248257153	1030	94	0.0	Barcelona	0.0	0.0	2	End	Si
1674	2248257159	779	0	0.0	Valladolid	0.0	0.0	14	End	Si
1675	2248257147	1031	0	0.0	Barcelona	0.0	0.0	14	End	Si
1676	2248274127	1	0	0.0	Valladolid	0.0	0.0	16	FormationSet	Si
1677	2248257151	1	0	0.0	Barcelona	0.0	0.0	16	FormationSet	Si

1678 rows × 13 columns



```
In [43]: # Will need to adjust for the correct filetype
df = df[df['teamId'] == 'Barcelona']
```

```
In [ ]: #Will need to correct to params of the files

df['passer'] = df['playerId']
df['recipient'] = df['playerId'].shift(-1)

passes = df[df['type'] == 'Pass']
successful = passes[passes['outcome'] == 'Successful']
```

```
In [45]: successful
```

Out[45]:

	id	eventId	minute	second	teamId	x	y	period	type	outcome
2	2248226929	3	0	1.0	Barcelona	50.0	50.0	1	Pass	Successful
3	2248226941	4	0	2.0	Barcelona	43.8	46.9	1	Pass	Successful
4	2248226951	5	0	4.0	Barcelona	36.6	56.6	1	Pass	Successful
5	2248226973	6	0	5.0	Barcelona	28.3	74.1	1	Pass	Successful
6	2248226975	7	0	8.0	Barcelona	11.2	53.7	1	Pass	Successful
...
1651	2248256867	1018	93	14.0	Barcelona	76.7	31.8	2	Pass	Successful
1652	2248256869	1019	93	17.0	Barcelona	79.6	24.2	2	Pass	Successful
1653	2248256983	1020	93	18.0	Barcelona	77.1	56.4	2	Pass	Successful
1654	2248256989	1021	93	20.0	Barcelona	68.3	55.2	2	Pass	Successful
1667	2248257081	1027	93	50.0	Barcelona	75.6	4.6	2	Pass	Successful

635 rows × 15 columns



```
In [46]: subs = df[df['type'] == 'SubstitutionOff']
subs = subs['minute']
firstsub = subs.min()
firstsub
```

Out[46]: np.int64(70)

```
In [47]: #Finding the average Location of a Player when passing
```

```
In [48]: successful = successful[successful['minute']<firstsub]
successful
```

Out[48]:

	id	eventId	minute	second	teamId	x	y	period	type	outcome
2	2248226929	3	0	1.0	Barcelona	50.0	50.0	1	Pass	Successful
3	2248226941	4	0	2.0	Barcelona	43.8	46.9	1	Pass	Successful
4	2248226951	5	0	4.0	Barcelona	36.6	56.6	1	Pass	Successful
5	2248226973	6	0	5.0	Barcelona	28.3	74.1	1	Pass	Successful
6	2248226975	7	0	8.0	Barcelona	11.2	53.7	1	Pass	Successful
...
1230	2248252877	782	69	30.0	Barcelona	83.1	42.8	2	Pass	Successful
1231	2248252881	783	69	32.0	Barcelona	90.5	23.0	2	Pass	Successful
1234	2248252897	785	69	36.0	Barcelona	84.6	27.7	2	Pass	Successful
1235	2248252905	786	69	37.0	Barcelona	90.7	14.3	2	Pass	Successful
1236	2248252913	787	69	38.0	Barcelona	88.4	15.3	2	Pass	Successful

505 rows × 15 columns



```
In [ ]: #This may need be adjusted to names?
pas = pd.to_numeric(successful['passer'],downcast='integer')
rec = pd.to_numeric(successful['recipient'],downcast='integer')

successful['passer'] = pas
successful['recipient'] = rec
```

In [50]: successful

Out[50]:

	id	eventId	minute	second	teamId	x	y	period	type	outcome
2	2248226929	3	0	1.0	Barcelona	50.0	50.0	1	Pass	Successful
3	2248226941	4	0	2.0	Barcelona	43.8	46.9	1	Pass	Successful
4	2248226951	5	0	4.0	Barcelona	36.6	56.6	1	Pass	Successful
5	2248226973	6	0	5.0	Barcelona	28.3	74.1	1	Pass	Successful
6	2248226975	7	0	8.0	Barcelona	11.2	53.7	1	Pass	Successful
...
1230	2248252877	782	69	30.0	Barcelona	83.1	42.8	2	Pass	Successful
1231	2248252881	783	69	32.0	Barcelona	90.5	23.0	2	Pass	Successful
1234	2248252897	785	69	36.0	Barcelona	84.6	27.7	2	Pass	Successful
1235	2248252905	786	69	37.0	Barcelona	90.7	14.3	2	Pass	Successful
1236	2248252913	787	69	38.0	Barcelona	88.4	15.3	2	Pass	Successful

505 rows × 15 columns



In [51]:

```
average_locations = successful.groupby('passer').agg({'x': ['mean'], 'y' : ['mean'] ,
average_locations.columns = ('x', 'y', 'count')
```

In [52]:

```
average_locations
```

Out[52]:

	x	y	count
passer			
1	7.124000	50.324000	25
2	68.274359	11.915385	39
4	32.895833	43.518750	48
8	53.332927	41.608537	82
9	69.916667	51.383333	6
10	63.201613	42.351613	62
15	38.409091	74.378182	55
16	69.594286	69.471429	35
18	55.242553	83.793617	47
21	55.663636	52.263636	55
28	49.156863	16.282353	51

```
In [53]: pass_between = successful.groupby(['passer' , 'recipient']).id.count().reset_index(
pass_between.rename({'id' : 'pass_count'}, axis='columns', inplace=True)

pass_between = pass_between.merge(average_locations, left_on= 'passer',right_index=
pass_between = pass_between.merge(average_locations, left_on= 'recipient',right_ind
```

```
In [54]: pass_between
```

```
Out[54]:
```

	passer	recipient	pass_count	x	y	count	x_end	y_end	coun
0	1	2	3	7.124000	50.324000	25	68.274359	11.915385	
1	1	4	3	7.124000	50.324000	25	32.895833	43.518750	
2	1	8	4	7.124000	50.324000	25	53.332927	41.608537	
3	1	9	1	7.124000	50.324000	25	69.916667	51.383333	
4	1	15	4	7.124000	50.324000	25	38.409091	74.378182	
...
85	28	8	13	49.156863	16.282353	51	53.332927	41.608537	
86	28	9	3	49.156863	16.282353	51	69.916667	51.383333	
87	28	10	5	49.156863	16.282353	51	63.201613	42.351613	
88	28	18	1	49.156863	16.282353	51	55.242553	83.793617	
89	28	21	5	49.156863	16.282353	51	55.663636	52.263636	

90 rows × 9 columns



```
In [55]: pass_between = pass_between[pass_between['pass_count']>3]
```

```
In [56]: pass_between
```

Out[56]:

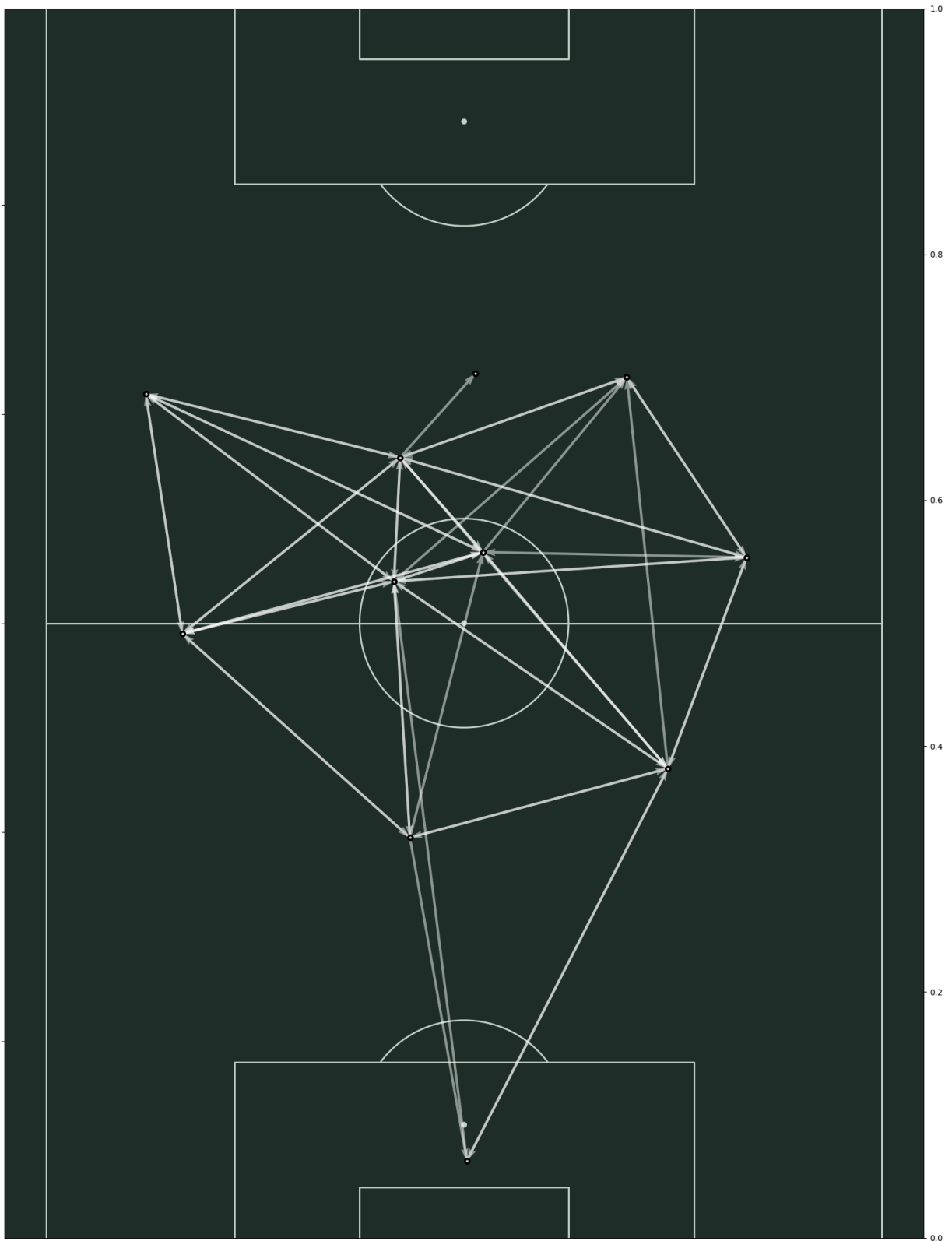
	passer	recipient	pass_count	x	y	count	x_end	y_end	coun
2	1	8	4	7.124000	50.324000	25	53.332927	41.608537	
4	1	15	4	7.124000	50.324000	25	38.409091	74.378182	
11	2	8	11	68.274359	11.915385	39	53.332927	41.608537	
13	2	10	11	68.274359	11.915385	39	63.201613	42.351613	
15	2	21	5	68.274359	11.915385	39	55.663636	52.263636	
16	2	28	7	68.274359	11.915385	39	49.156863	16.282353	
17	4	1	5	32.895833	43.518750	48	7.124000	50.324000	
19	4	8	6	32.895833	43.518750	48	53.332927	41.608537	
21	4	15	20	32.895833	43.518750	48	38.409091	74.378182	
23	4	21	6	32.895833	43.518750	48	55.663636	52.263636	
24	4	28	7	32.895833	43.518750	48	49.156863	16.282353	
26	8	2	8	53.332927	41.608537	82	68.274359	11.915385	
27	8	4	8	53.332927	41.608537	82	32.895833	43.518750	
29	8	10	23	53.332927	41.608537	82	63.201613	42.351613	
30	8	15	6	53.332927	41.608537	82	38.409091	74.378182	
31	8	16	8	53.332927	41.608537	82	69.594286	69.471429	
32	8	18	6	53.332927	41.608537	82	55.242553	83.793617	
33	8	21	10	53.332927	41.608537	82	55.663636	52.263636	
34	8	28	11	53.332927	41.608537	82	49.156863	16.282353	
39	10	2	6	63.201613	42.351613	62	68.274359	11.915385	
41	10	8	17	63.201613	42.351613	62	53.332927	41.608537	
42	10	9	4	63.201613	42.351613	62	69.916667	51.383333	
43	10	15	5	63.201613	42.351613	62	38.409091	74.378182	
44	10	16	5	63.201613	42.351613	62	69.594286	69.471429	
45	10	18	5	63.201613	42.351613	62	55.242553	83.793617	
46	10	21	12	63.201613	42.351613	62	55.663636	52.263636	
47	10	28	7	63.201613	42.351613	62	49.156863	16.282353	
48	15	1	6	38.409091	74.378182	55	7.124000	50.324000	
50	15	4	9	38.409091	74.378182	55	32.895833	43.518750	
51	15	8	6	38.409091	74.378182	55	53.332927	41.608537	

	passer	recipient	pass_count	x	y	count	x_end	y_end	coun
53	15	16	6	38.409091	74.378182	55	69.594286	69.471429	
54	15	18	15	38.409091	74.378182	55	55.242553	83.793617	
55	15	21	10	38.409091	74.378182	55	55.663636	52.263636	
60	16	10	10	69.594286	69.471429	35	63.201613	42.351613	
62	16	18	11	69.594286	69.471429	35	55.242553	83.793617	
68	18	8	6	55.242553	83.793617	47	53.332927	41.608537	
70	18	10	7	55.242553	83.793617	47	63.201613	42.351613	
71	18	15	7	55.242553	83.793617	47	38.409091	74.378182	
72	18	16	10	55.242553	83.793617	47	69.594286	69.471429	
73	18	21	9	55.242553	83.793617	47	55.663636	52.263636	
74	21	2	4	55.663636	52.263636	55	68.274359	11.915385	
76	21	8	13	55.663636	52.263636	55	53.332927	41.608537	
77	21	10	14	55.663636	52.263636	55	63.201613	42.351613	
78	21	15	7	55.663636	52.263636	55	38.409091	74.378182	
79	21	16	5	55.663636	52.263636	55	69.594286	69.471429	
81	21	28	7	55.663636	52.263636	55	49.156863	16.282353	
83	28	2	10	49.156863	16.282353	51	68.274359	11.915385	
84	28	4	13	49.156863	16.282353	51	32.895833	43.518750	
85	28	8	13	49.156863	16.282353	51	53.332927	41.608537	
87	28	10	5	49.156863	16.282353	51	63.201613	42.351613	
89	28	21	5	49.156863	16.282353	51	55.663636	52.263636	

```
In [57]: pitch = VerticalPitch(pitch_type='statsbomb',pitch_color='#22312b' , line_color='#c
#Figsize goes in the draw function
fig, ax = pitch.draw(figsize= (16,21))
#All of the original X Y were in a 100X100 grid so we need to transfer them to to a
arrows = pitch.arrows(1.2*pass_between.x,.8*pass_between.y,1.2*pass_between.x_end,.
nodes = pitch.scatter(1.2*average_locations.x,.8*average_locations.y,
                      s=30,color='#d3d3d3',edgecolors = 'black', linewidth =2.5, al
ax.yaxis.tick_right()
ax2 = ax.twinx()
ax2.set_ylabel
```

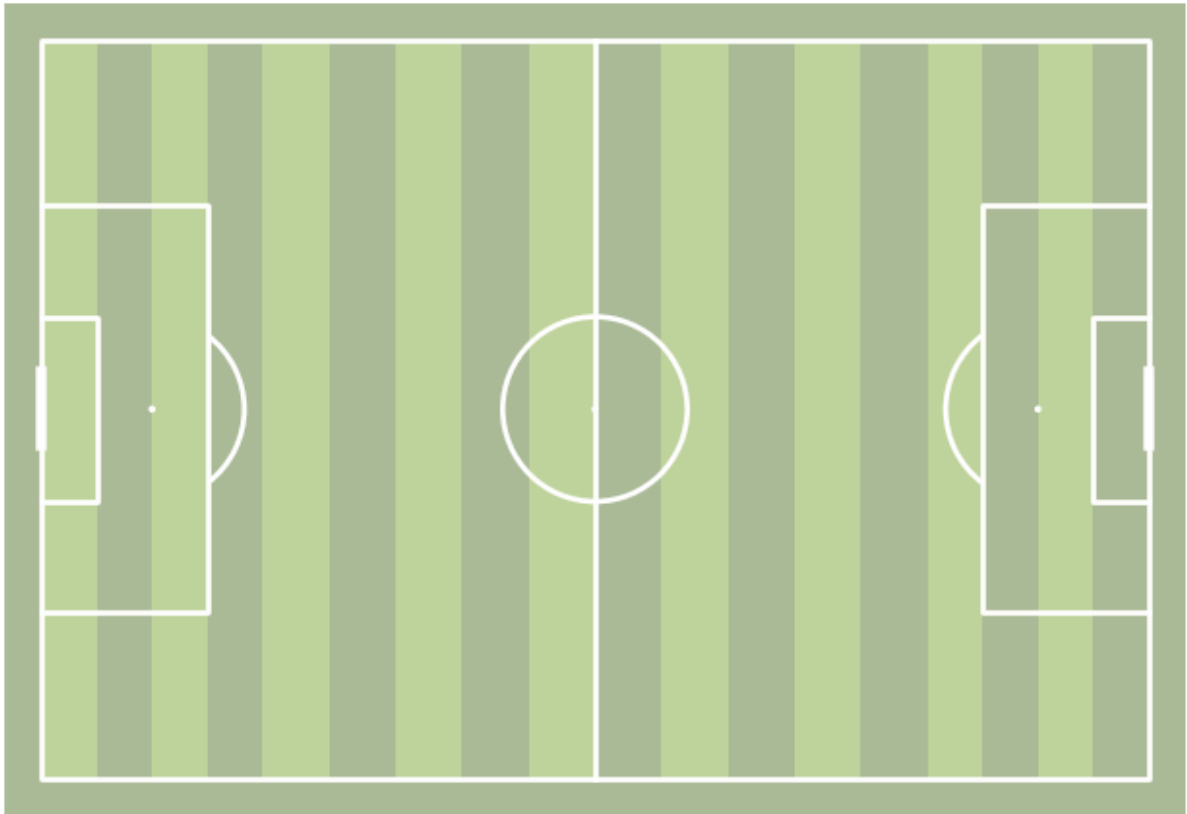
Out[57]: <bound method _AxesBase.set_ylabel of <Axes: >>

Ignoring fixed y limits to fulfill fixed data aspect with adjustable data limits.
Ignoring fixed y limits to fulfill fixed data aspect with adjustable data limits.

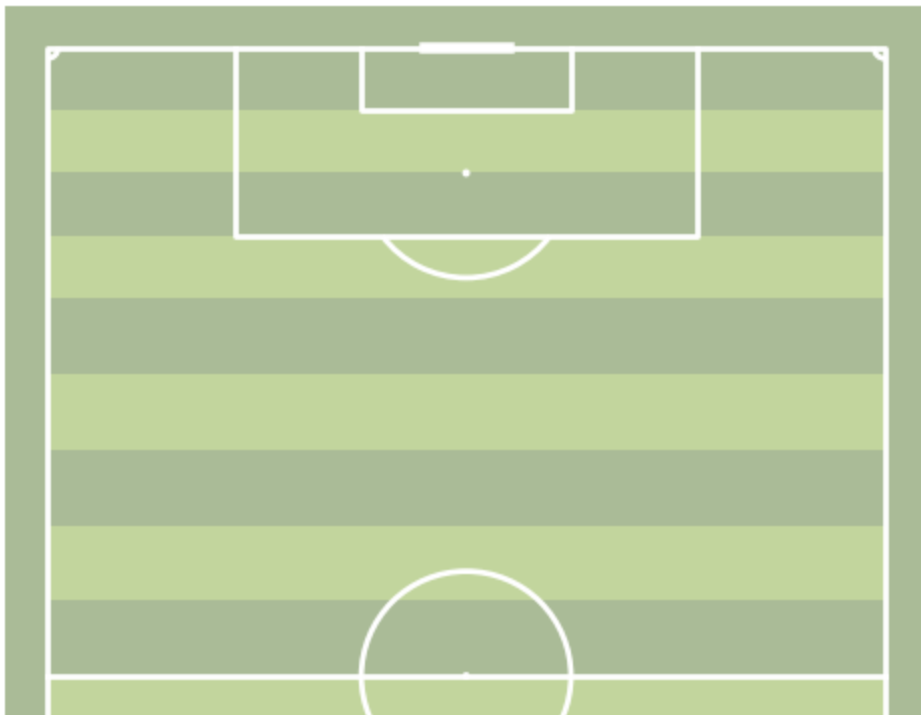


In [58]: *#The following are examples of available pitch types and layouts.*

```
In [59]: pitch = Pitch(pitch_color='#aabb97', line_color='white',  
                      stripe_color='#c2d59d', stripe=True) # optional stripes  
fig, ax = pitch.draw()
```

```
In [60]: pitch = VerticalPitch(corner_arcs=True, half=True, pitch_color='#aabb97', line_color=
        stripe_color='#c2d59d', stripe=True)
        #Made /2 because size in file was too big
        fig, ax = pitch.draw(figsize=(10/2, 7.727/2))
```



```
In [61]: pitch = Pitch(positional=True, shade_middle=True, positional_color='#eadddd', shade  
fig, ax = pitch.draw()
```

