

# Introducing a Statistical and Geometric approach to Visualising and Analysing Passing in Football using R

Aaron Carman 180350874

Year 3 - Semester 2



**Queen Mary**  
**University of London**

# Contents

<b>Introduction</b>	<b>4</b>
<b>Data Sourcing</b>	<b>4</b>
Availability of Football Data . . . . .	4
How to work with StatsBomb data . . . . .	4
Installing packages . . . . .	4
Pulling in StatsBomb data . . . . .	5
What data is included? . . . . .	5
<b>Pass Maps</b>	<b>8</b>
Creating a Pass Map in R . . . . .	8
Pass Map code for individuals . . . . .	9
Interpreting Pass Maps . . . . .	10
Liverpool Pass Map . . . . .	10
Liverpool and Tottenham Shot Maps . . . . .	11
Pass Maps for Individuals Liverpool Players . . . . .	12
Comparing Pass Maps with discrete data . . . . .	13
Table of Passing Frequency . . . . .	13
Barplots of discrete passing data . . . . .	14
<b>Pass Flow Graphs</b>	<b>16</b>
Creating a Pass Flow Graph in R . . . . .	16
Preparing the data . . . . .	16
Plotting the graph . . . . .	18
Interpreting Pass Flow Graphs . . . . .	19
Liverpool Pass Flow Graph . . . . .	19
Pass Flow Graphs for Individual Liverpool Players . . . . .	20
<b>Passing Networks</b>	<b>22</b>
What are Passing Networks? . . . . .	22
Creating Passing Networks . . . . .	22
Interpreting Passing Networks . . . . .	25
Liverpool Passing Network . . . . .	25
Barcelona Passing Networks vs Manchester United . . . . .	26
Manchester United Passing Network vs Barcelona . . . . .	27

<b>A Geometric Approach</b>	<b>28</b>
Introducing Voronoi diagrams . . . . .	28
What are Voronoi diagrams? . . . . .	28
How do Voronoi diagrams relate to football? . . . . .	28
Voronoi diagrams of generated examples using Euclidean distance . . . . .	29
Creating the Voronoi diagrams . . . . .	29
Comparing formations using Voronoi diagrams . . . . .	30
Snapshots of Voronoi diagrams from real game situations . . . . .	35
Tanguy Ndombele’s goal vs Sheffield . . . . .	35
<b>Discussions and Conclusions</b>	<b>37</b>
<b>A Note Regarding References</b>	<b>38</b>
<b>Bibliography</b>	<b>38</b>

# Introduction

In football, the analysis of data is more prevalent than ever. Mathematical approaches are used to assess football games, increasing the understanding of what happens on a football pitch through statistics and geometry rather than opinion. In the present work, I suggest methods of visualising passing data and describe how to assess pitch control visually using Voronoi diagrams. I present how to map passes and produce pass flow graphs in R; showing how to interpret these to extract useful information.

## Data Sourcing

### Availability of Football Data

It is known that sourcing high quality data can be difficult and expensive. However, the collection of football data is a rapidly growing industry in which many companies are trying to catch the boat before it leaves them behind. As expected, this has come due to higher demand, generated from a diverse range of people who are interested. Previously, betting companies were the only ones collecting sports data, to ensure they chose the correct odds that entice punters, yet ensure profit. In recent times, there are more people interested in the data. With the increasing possibilities, nearly everyone who is interested in football is also at least somewhat interested in the data that it produces. On TV, we see Gary Lineker on Match of the Day showing the average positions of a team from that day's game. The following evening, we see Jamie Carragher and Gary Neville on Monday Night Football discussing the statistics of England's right-back options and why they should or shouldn't be selected for the European Championships. Behind the scenes, the football teams are happy to spend their money to get the best data and analysts out there; desperate to gain an edge on their opponents. These potential advantages come in different fashions. Data is now used for tactics, scouting, strength and conditioning, and much more. Teams aim to use a combination of these methods to find the optimal solutions to each problem.

Many top-level football teams invest in the infrastructure to collect their own data, Memmert and Raabe (2018). Of course, there are also an abundance of companies that sell collected data at various prices for the varying levels of data. Examples of these companies include Opta and Squawka.

Currently, some of the most sought-after data is optical tracking data, which allows for a great deal of analysis into various components of football. For this project, I will mostly be using free data provided by StatsBomb, which does not include constant tracking data for each individual player. However, it does include the x and y coordinates of the ball at the time of each action. This is very important as it allows us to map passes including their start and end location.

### How to work with StatsBomb data

#### Installing packages

As an example of how to access the StatsBomb data that I will be using during this project, I will show, step-by-step, how to use StatsBomb's R package to import and filter the matches that you want to study. In this example, I will be looking at the recent all-English Champions League Final between my home team, Liverpool, and London's Tottenham Hotspurs.

Firstly, we install the packages that we want to use. We will be using: "StatsBombR", "tidyverse", "devtools", "ggplot2", "StatsBombR".

"tidyverse" is a collection of open-source R packages to be used for data science and manipulating data. It contains the most popular package for data visualisation, "ggplot2".

Most packages are hosted on CRAN (Comprehensive R Archive Network). However, "StatsBombR" is hosted on Github and we therefore need to use "devtools" to download it directly.

“StatsBombR” is the package made by StatsBomb which allows us to clean and pull the data that they provide for free. We can select data through the package by selecting Competitions, Matches, etc. It may be helpful to check what functions there are in the package by searching for its help page in the bottom right viewer in R studio.

## Pulling in StatsBomb data

Below is the code used to pull in the event-level data for the aforementioned 2018/19 Champion’s League Final between Liverpool and Tottenham.

```
install.packages("tidyverse")
install.packages("devtools")
devtools::install_github("statsbomb/StatsBombR")
```

Use the initial code above to install some of the important packages that we will be using to obtain the data.

```
library("StatsBombR")
library("tidyverse")
```

Use library() to load in the important packages that we need to use.

```
Comp <- FreeCompetitions()
Champ <- FreeCompetitions() %>%
  filter(competition_id==16 & season_name=="2018/2019")
```

Next, we filter the competitions and seasons we want to analyse. In this case, we are using the competition id 16 to represent the Champion’s League. If we were looking at the La Liga data, we would use 11. To find the corresponding competition id’s, look at the ‘Comp’ variable. Of course, the season name is 2018/2019 for our case. There is also an assigned season id for each season that can also be found in the ‘Comp’ variable.

```
Matches <- FreeMatches(Champ)
```

The following line of code pulls all the matches from the selected competition. Unfortunately, for the Champion’s League, the only data available for free is that of the finals. Hence, there is no additional filtering to do at this stage.

```
ChampData <- StatsBombFreeEvents(MatchesDF = Matches, Parallel = T)
ChampData = allclean(CHAMPData)
```

We then create the data frame “ChampData” of the free event-level data from our chosen game. The function allclean() cleans and extracts the data to make sure we have all the relevant data.

Now you should be able to look at a nice (big) table of all of the data in our data frame by running view(CHAMPData).

## What data is included?

Below, we will familiarise ourselves with the types of data points that we have available by looking at the long list of all the variables.

```
names(ChampData)
```

```
## [1] "id" "index"
## [3] "period" "timestamp"
## [5] "minute" "second"
## [7] "possession" "duration"
## [9] "related_events" "location"
## [11] "under_pressure" "off_camera"
## [13] "counterpress" "out"
## [15] "type.id" "type.name"
## [17] "possession_team.id" "possession_team.name"
## [19] "play_pattern.id" "play_pattern.name"
## [21] "team.id" "team.name"
## [23] "tactics.formation" "tactics.lineup"
## [25] "player.id" "player.name"
## [27] "position.id" "position.name"
## [29] "pass.length" "pass.angle"
## [31] "pass.end_location" "pass.switch"
## [33] "pass.aerial_won" "pass.through_ball"
## [35] "pass.inswinging" "pass.straight"
## [37] "pass.cross" "pass.assisted_shot_id"
## [39] "pass.shot_assist" "pass.outswinging"
## [41] "pass.cut_back" "pass.deflected"
## [43] "pass.goal_assist" "pass.recipient.id"
## [45] "pass.recipient.name" "pass.height.id"
## [47] "pass.height.name" "pass.body_part.id"
## [49] "pass.body_part.name" "pass.type.id"
## [51] "pass.type.name" "pass.outcome.id"
## [53] "pass.outcome.name" "pass.technique.id"
## [55] "pass.technique.name" "carry.end_location"
## [57] "ball_receipt.outcome.id" "ball_receipt.outcome.name"
## [59] "duel.type.id" "duel.type.name"
## [61] "duel.outcome.id" "duel.outcome.name"
## [63] "foul_committed.penalty" "foul_committed.advantage"
## [65] "foul_committed.offensive" "foul_committed.type.id"
## [67] "foul_committed.type.name" "shot.statsbomb_xg"
## [69] "shot.end_location" "shot.freeze_frame"
## [71] "shot.key_pass_id" "shot.first_time"
## [73] "shot.aerial_won" "shot.technique.id"
## [75] "shot.technique.name" "shot.body_part.id"
## [77] "shot.body_part.name" "shot.type.id"
## [79] "shot.type.name" "shot.outcome.id"
## [81] "shot.outcome.name" "goalkeeper.end_location"
## [83] "goalkeeper.punched_out" "goalkeeper.position.id"
## [85] "goalkeeper.position.name" "goalkeeper.type.id"
## [87] "goalkeeper.type.name" "goalkeeper.outcome.id"
## [89] "goalkeeper.outcome.name" "goalkeeper.technique.id"
## [91] "goalkeeper.technique.name" "goalkeeper.body_part.id"
## [93] "goalkeeper.body_part.name" "clearance.head"
## [95] "clearance.aerial_won" "clearance.right_foot"
## [97] "clearance.left_foot" "clearance.other"
## [99] "clearance.body_part.id" "clearance.body_part.name"
## [101] "foul_won.advantage" "foul_won.defensive"
```

```

## [103] "interception.outcome.id"      "interception.outcome.name"
## [105] "ball_recovery.recovery_failure" "dribble.overrun"
## [107] "dribble.nutmeg"               "dribble.outcome.id"
## [109] "dribble.outcome.name"         "50_50.outcome.id"
## [111] "50_50.outcome.name"           "substitution.outcome.id"
## [113] "substitution.outcome.name"     "substitution.replacement.id"
## [115] "substitution.replacement.name" "block.deflection"
## [117] "block.offensive"              "match_id"
## [119] "competition_id"               "season_id"
## [121] "location.x"                   "location.y"
## [123] "carry.end_location.x"         "carry.end_location.y"
## [125] "pass.end_location.x"          "pass.end_location.y"
## [127] "shot.end_location.x"          "shot.end_location.y"
## [129] "shot.end_location.z"          "shot_impact_height"
## [131] "player.name.GK"               "player.id.GK"
## [133] "location.x.GK"                "location.y.GK"
## [135] "DistToGoal"                   "DistToKeeper"
## [137] "AngleToGoal"                  "AngleToKeeper"
## [139] "AngleDeviation"               "avevelocity"
## [141] "DistSGK"                      "density"
## [143] "density.incone"               "distance.ToD1"
## [145] "distance.ToD2"                "AttackersBehindBall"
## [147] "DefendersBehindBall"          "DefendersInCone"
## [149] "InCone.GK"                    "DefArea"
## [151] "distance.ToD1.360"            "distance.ToD2.360"
## [153] "milliseconds"                 "ElapsedTime"
## [155] "StartOfPossession"            "TimeInPoss"
## [157] "TimeToPossEnd"

```

For this project, I will mostly analyse the variables related to passes. If I want to see who is receiving the ball, I will look at the variable 'pass.recipient.name'; and to determine if a pass is successful, I will look at 'pass.outcome.id', etc. Included in the available data is the x, y location of the ball and also details of each pass, including the end location of each pass. This information is vital for the pass maps and pass flow graphs, along with other analysis.

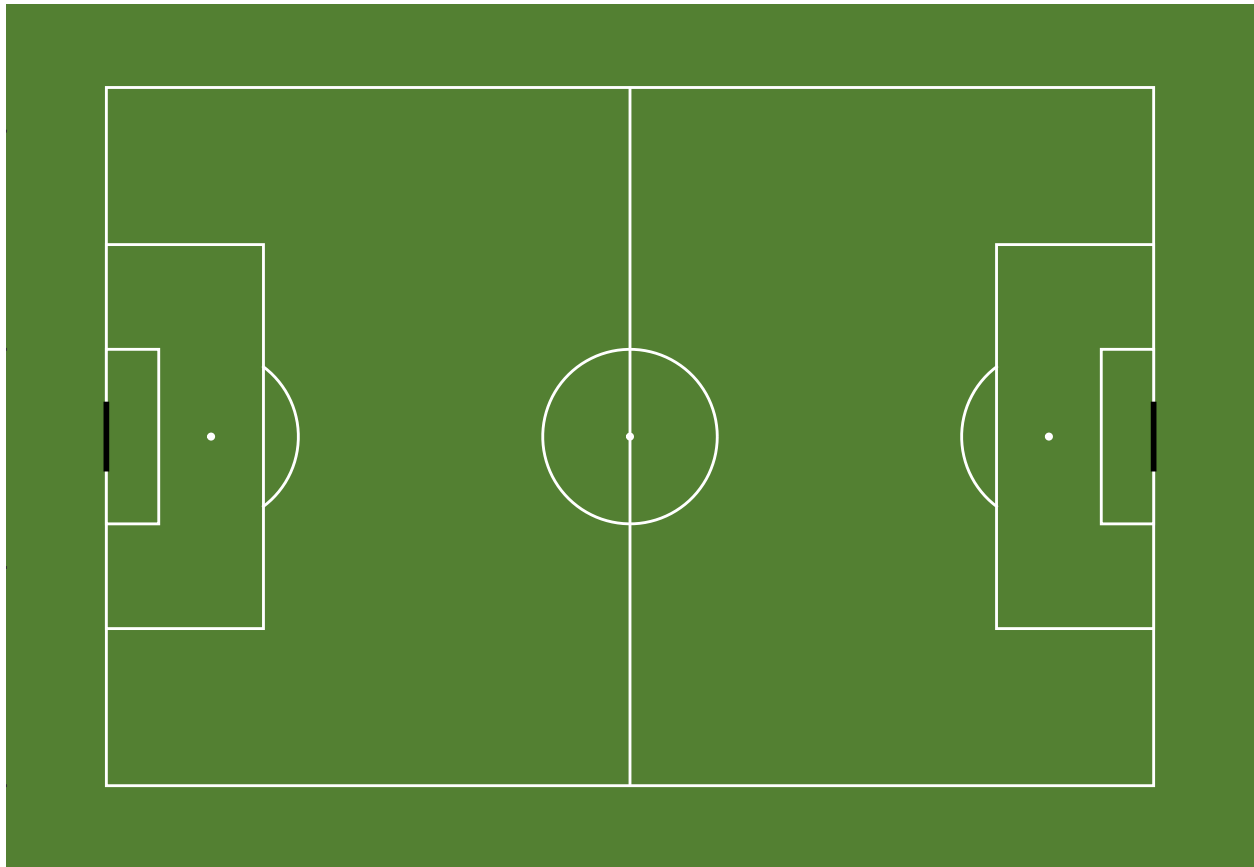
# Pass Maps

## Creating a Pass Map in R

```
passes <- ChampData%>%  
  filter(type.name=="Pass", team.name=="Liverpool")
```

We use the previously prepared ‘ChampData’ data set which includes a whole manner of different data points. In fact, there are 3165 observations of 157 different variables. That’s a lot of data! All that we want to include in our graph are the passes made by Liverpool. To do this, we specify that we want the type of action: `type.name=="Pass"` and the team: `team.name=="Liverpool"`. If you want to instead create a shot map, replace “Pass” or “passes” here and anywhere else in the code with “Shot” and “shots”. It’s simple, but it works. I will show it alongside the pass map just for fun.

```
library(SBpitch)  
create_Pitch(grass_colour = "#538032",  
line_colour = "#ffffff",  
background_colour = "#538032",  
goal_colour = "#000000")
```



To create a pitch much more easily, I have used the package “SBpitch” and I would recommend anyone else to do the same, (“FCrSTATS/Sbpitch” 2021). Essentially, it allows someone else to do the hard work of coordinating the lines correctly, and you can move more quickly onto plotting and analysing the passes. I have chosen to use the input choices for ‘the green pitch’, but there are a few more. Alternatively, you



could choose to have a red pitch for Liverpool if you really wanted to. It's not quite Atletico's Wanda Metropolitana Stadium that the game was actually played in; but it will do for our pass maps.

```
geom_point(data=passes, aes(x=location.x, y=location.y), alpha=0.9)
```

The `geom_point` function from 'ggplot2' plots points of our passes on top of our pitch. You must input the data, which I previously named as 'passes'. Then, using `aes()`, add the x and y location of the ball. The alpha input changes size, feel free to play around with this to see what works for you. I found 0.9 to be suitable.

```
geom_segment(data = passes, aes(x = location.x, y = location.y, xend = pass.end_location.x,
yend = pass.end_location.y), alpha=0.9, arrow = arrow(length = unit(0.08, "inches")))
```

We are then adding code to draw an arrow from the start of the pass to the end. Similarly to the last line of code, we add the data, which is 'passes'. We again use the `aes` function, this time adding information for the end position for the pass. For this, we use `pass.end_location.x` and `pass.end_location.y` from our StatsBomb data. Again, I have `alpha = 0.9`. Finally, we include the last part so that we can have an arrow, indicating direction, instead of just a line.

```
scale_y_reverse()
```

For StatsBomb data, the y-axis data is flipped. Meaning zero is at the top and sixty-eight is at the bottom. Therefore, we must flip the y-axis using the function above. If we didn't apply this function, we could come up with opposite interpretations from the true picture.

```
labs(title=" Liverpool", subtitle = " vs Tottenham")
```

To finish, I have included a title that is suitable for the plot. I have individualised this for each player pass map.

### Pass Map code for individuals

There is thankfully not much code that needs to be changed for each individual pass map. I will show the changes that I made to create a pass map for each of the 14 individuals who made an appearance for Liverpool against Tottenham in the Champions League Final.

```
Alisson = "Alisson Ramsés Becker"
Trent = "Trent Alexander-Arnold"
Matip = "Joël Andre Job Matip"
Virgil = "Virgil van Dijk"
Robertson = "Andrew Robertson"
Henderson = "Jordan Brian Henderson"
Fabinho = "Fábio Henrique Tavares"
Wijnaldum = "Georginio Wijnaldum"
Salah = "Mohamed Salah"
Firmino = "Roberto Firmino Barbosa de Oliveira"
Mane = "Sadio Mané"
Origi = "Divock Okoth Origi"
Milner = "James Philip Milner"
Gomez = "Joe Gomez"
```

In StatsBomb dataframes, the player names are long, awkward to remember, and include accents that aren't easy to reproduce. To avoid having to input their full names every time, I wrote the above code so that I could input their names in a way that I am more familiar with.

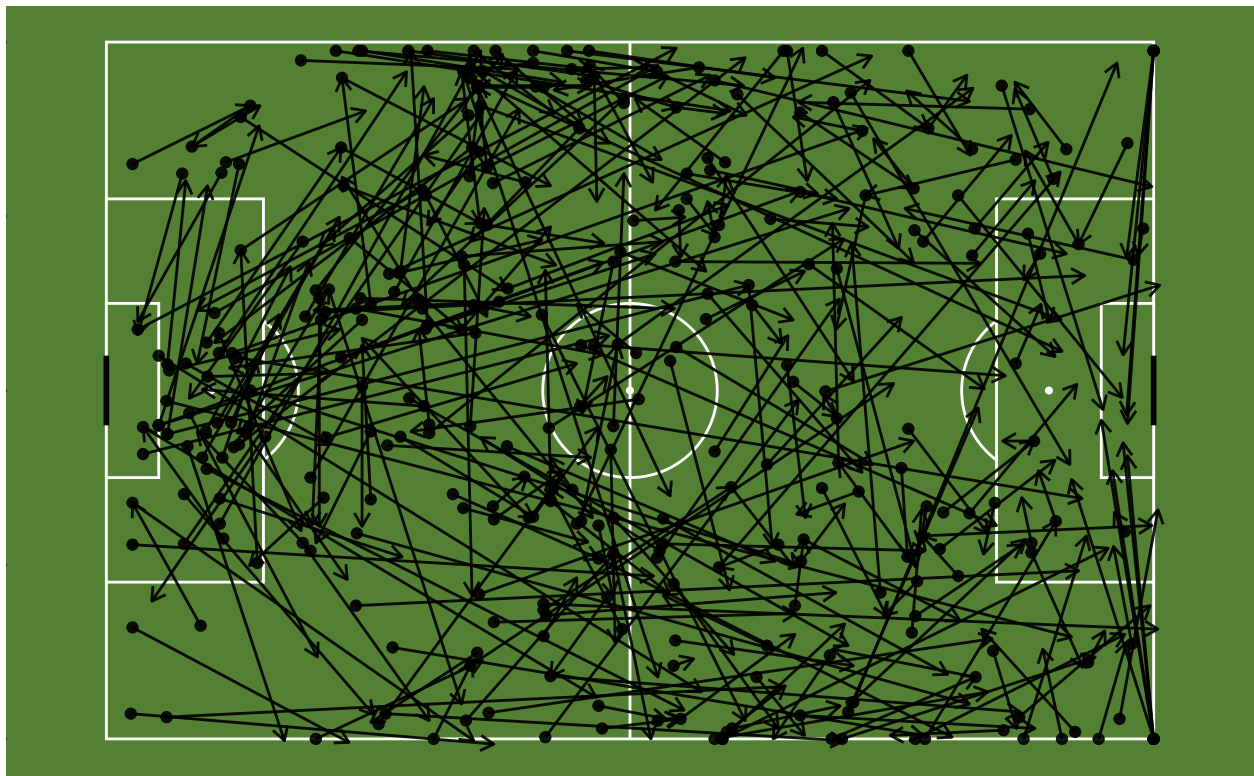
```
p3<-ChampData%>%  
  filter(type.name=="Pass", player.name==Fabinho)
```

As an example above, I have shown how I filtered the data for Fábio Henrique Tavares's passes against Tottenham. Instead of team.name, I have specified the player.name, which due to the code above is now "Fabinho". I have assigned the data to 'p3', referring to the number 3 that Fábio wears on his jersey. This is an easy, short way to assign them. Where we previously had 'passes', we will now have 'p3' for Fábio's passes, 'p66' for Trent Alexander-Arnold's passes, etc.

## Interpreting Pass Maps

### Liverpool Pass Map

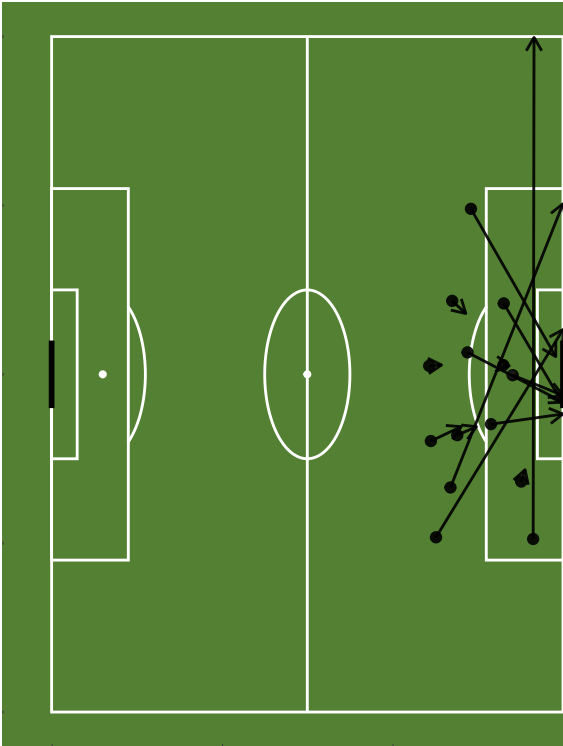
#### Liverpool vs Tottenham



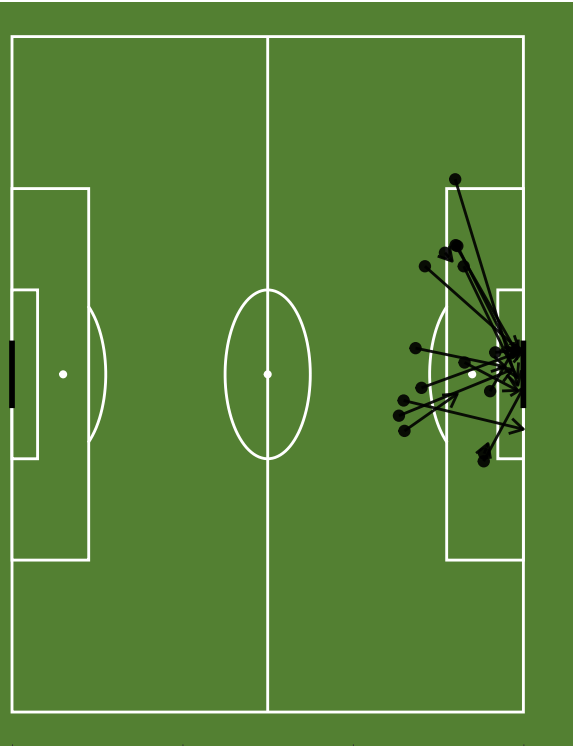
The above passing map of all of Liverpool's passes is very busy and it is hard to ascertain any clear patterns. We can see that Liverpool are attacking from left to right and that they seem to play a variety of different passes: short, medium and long. There is also not a clear bias to either side of the pitch, with both full backs being a regular option.

## Liverpool and Tottenham Shot Maps

Liverpool  
vs Tottenham



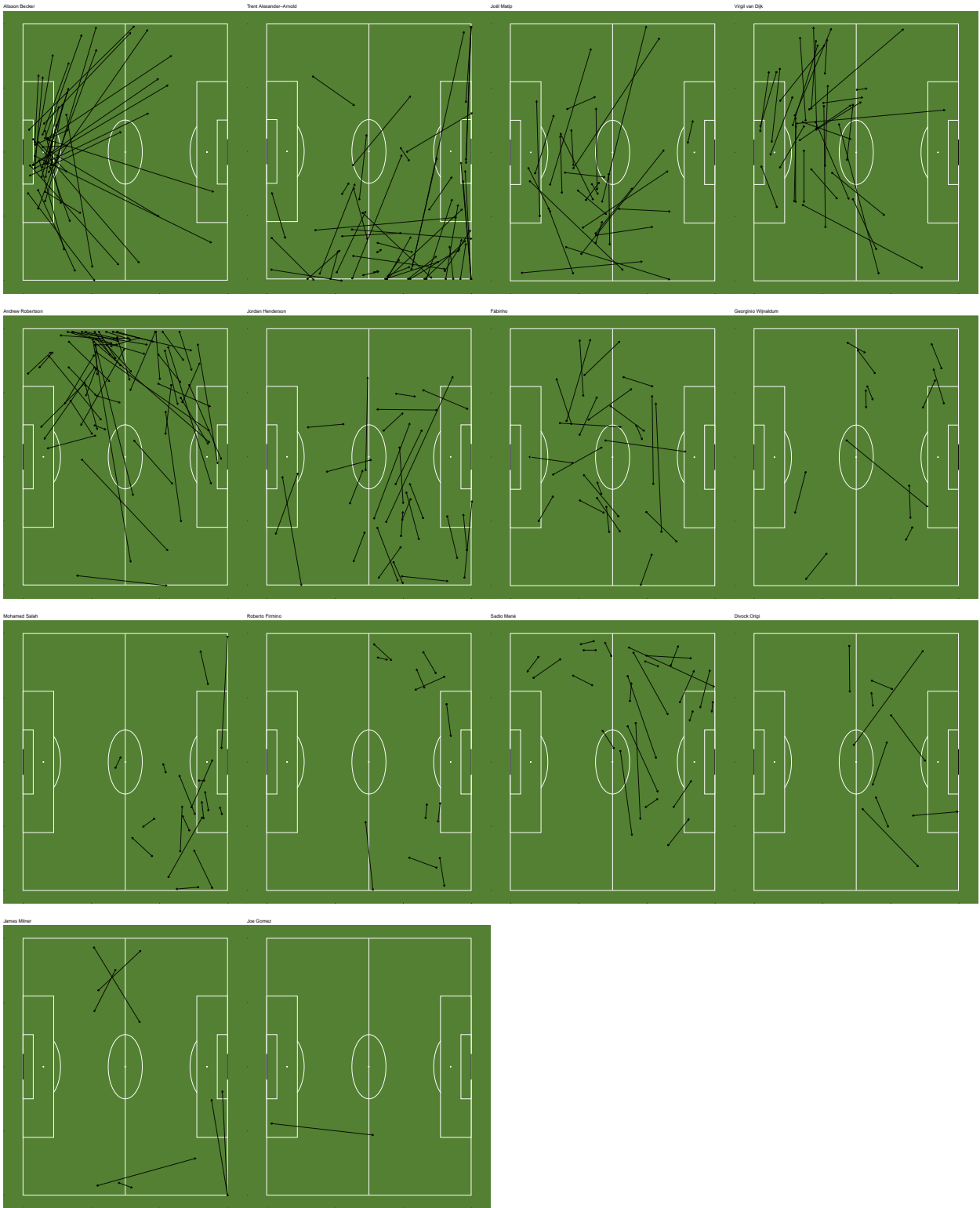
Tottenham  
vs Liverpool



Just for fun, I have included shot maps for both Liverpool and Tottenham. We can see that both teams had a similar number of chances. In fact, we can see from the short arrows that Tottenham managed to block most of Liverpool's shots from outside the box, before they could trouble Hugo Lloris. In football, the score line doesn't always represent the team which had the better chances. The only thing that a score line definitively tells us is how many times each team put the ball in the other team's net.

Happily for me, you can notice the two goals scored by Liverpool. First, the early goal from Mohamed Salah's penalty. Then you can also notice the shot from Divock Origi across the goalkeeper that nestled into the bottom right corner.

Pass Maps for Individuals Liverpool Players



Looking at the pass maps for each player, we can often see patterns that align with our understanding of the player. For example, Roberto Firmino makes short passes, acting as a link between players; not trying to find players with long passes. As expected, Liverpool's most active players are the defenders. This is because these are the players with the least pressure on them, and can often pass between each other. This reasoning particularly applies to the centre back partnership of Virgil van Dijk and Joel Matip who you can see frequently pass the ball sideways to each other. This could be exaggerated in this specific game due to Salah's early goal from a penalty. Therefore, Liverpool had less desire to make adventurous forward passes since there was a lower risk-reward ratio based on their advantage in the score line. The other defenders, Andrew Robertson and Trent Alexander-Arnold are also involved regularly but with slightly different styles. Andrew Robertson passes more balls inwards from the touchline compared to Trent Alexander-Arnold who seems to more often look down the line to Mohamed Salah and Jordan Henderson.

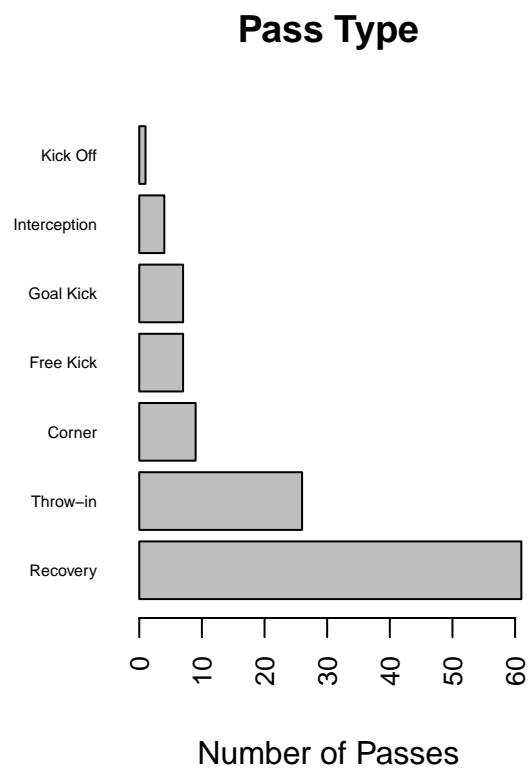
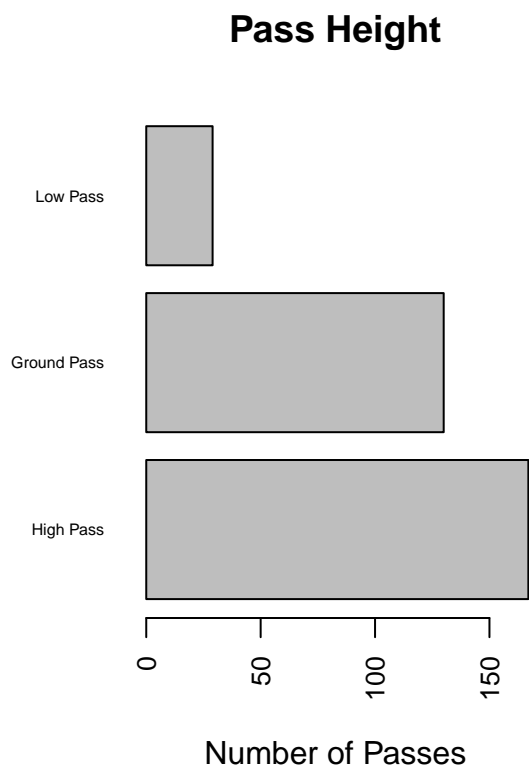
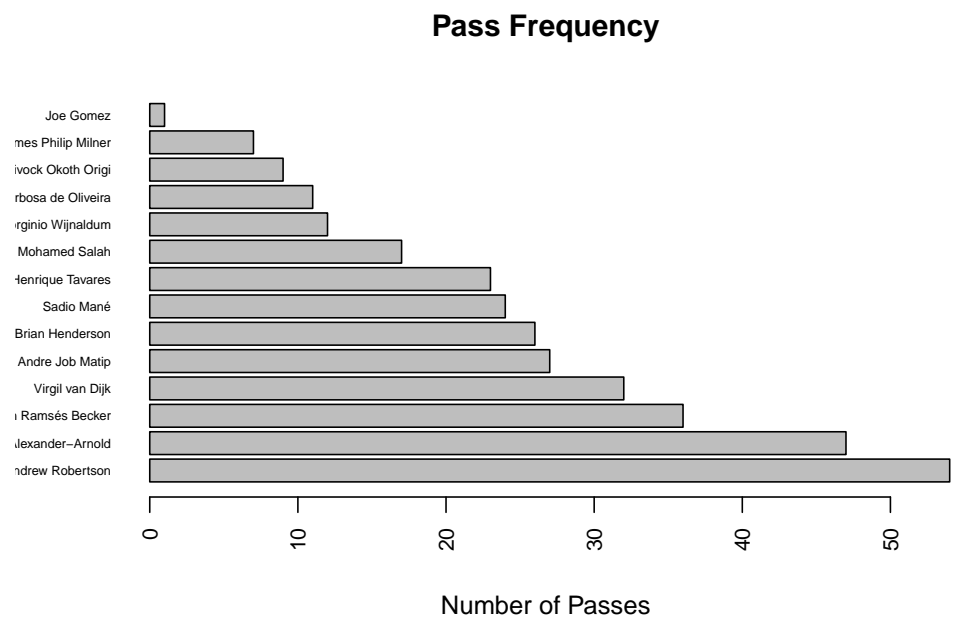
You can also recognise the players who did not get much of the ball. Most notably from the list of starting players, Roberto Firmino made a small number of passes, as seen on his pass map. In addition, most of these passes were short passes made outside of the area that he usually works in. Particularly in the 2018/19 season, Firmino was a player who links up play very well in the central area between midfield and attack. However, in the final against Tottenham, this was clearly not the case. After missing the previous three games due to an injury, Firmino seemed out of sorts and couldn't find his way into the flow of the game. Viewers didn't see any of his link up play that they were anticipating. Instead, they saw him drifting wide into less pressurised areas, where he still made only a few short passes.

## Comparing Pass Maps with discrete data

### Table of Passing Frequency

##		
##	Alisson Ramsés Becker	Andrew Robertson
##	36	54
##	Divock Okoth Origi	Fábio Henrique Tavares
##	9	23
##	Georginio Wijnaldum	James Philip Milner
##	12	7
##	Joe Gomez	Joël Andre Job Matip
##	1	27
##	Jordan Brian Henderson	Mohamed Salah
##	26	17
##	Roberto Firmino Barbosa de Oliveira	Sadio Mané
##	11	24
##	Trent Alexander-Arnold	Virgil van Dijk
##	47	32

Barplots of discrete passing data



In the bar plot showing player pass totals, we see that what we saw on the pass maps is backed up by the discrete data. Liverpool's left-back Andrew Robertson has made the most passes, followed by Trent Alexander-Arnold. We can see that Alisson Becker is the next most frequent passer for Liverpool, making 36 passes. This is surprising, but again could be a reflection on the way the game panned out after Liverpool's early goal.

Our pass maps do not tell the entire story and there are improvements to be made. In the bar plots, I have looked at some other data that is not represented by the pass maps. For example, the current pass maps do not tell us what type of pass is made, or the circumstances the pass was made under. Perhaps an improvement that could be made to the pass maps would be to alter the colour and/or style of the lines to indicate this information. Just from looking at the pass maps, I would not have known that approximately 50% of Liverpool's passes were high passes. Of course, this can be understood when you consider the 9 corners, 7 freekicks, 7 goal kicks and many defensive clearances.

# Pass Flow Graphs

Pass flow graphs are a good way of making messy pass maps clearer; giving us further insight into what is actually happening. A pass flow graph splits the pitch into grids representing different areas on the pitch where passes have been made. From the graph, you can see the frequency, average direction and average length of each pass made by an individual or a team.

## Creating a Pass Flow Graph in R

To create these pass flow graphs, I have implemented much of the code and reasoning seen used by Mishra (2021).

### Preparing the data

```
passes
```

For these pass flow graphs, I will be using the 'passes' data that I have previously filtered from StatsBomb. Please refer back to 'Pulling in StatsBomb data' and 'Creating a Pass Map in R' to see the code for this.

```
library(tidyverse)
library(ggplot2)
library(dplyr)
library(ggsoccer)
library(viridis)
library(StatsBombR)
```

To produce the following pass flow graphs, you must ensure that the above packages are installed in R and ready to be utilised.

```
bin <- 20
x_bin <- 120/bin
y_bin <- 80/bin
```

Choose a suitable value for 'bin' which represents the size of the grids that we will be splitting our pass flow graph into. Then, 'x\_bin' creates our columns by dividing the length of the StatsBomb pitch by our chosen 'bin' value; and 'y\_bin' creates our rows by dividing the height of the pitch by the same 'bin' value.

```
passfx <- seq(0,120, by=bin)
passfy <- seq(0,80, by=bin)
```

Creating an array, we code a sequence going from 0 to 120 for 'passfx' and from 0 to 80 for 'passfy'. Each sequence increases in increments of our chosen 'bin' value at each interval. 'passfx' creates the array (0, 20, 40, 60, 80, 100, 120) and 'passfy' creates the array (0,20,40,60,80).

```
PassFlow <- data.frame("x"=0, "y"=0, "endx"=0, "endy"=0, countP=0)
```

Using the data.frame function, we make a data frame to store the coordinates for all of the passes made by Liverpool players. We choose the column names to be x, y, endx, endy and countP; representing the start and end locations of the passes and the number (count) of passes made. We set all the variables to be zero.



```
PFLFC <- passes
for(i in 1:x_bin){
  filterx<- PFLFC %>% filter(location.x>=passfx[i])%>%
  filter(location.x<passfx[i+1])
}
```

In this case, we are using all of the players in the ‘passes’ data frame, therefore we do not need to filter the data to remove any of the players passes. Our first ‘for’ loop goes through each of the 6 created columns. In these columns, we filter our passes to find those where the starting point of the pass belongs to that column. The code begins by filtering all of the passes that belong between x=0 and x=20, then between x=20 to x=40, etc.

```
for(j in 1:y_bin){
  filtery <- filterx %>% filter(location.y>=passfy[j])%>%
  filter(location.y<passfy[j+1])
}
```

Reproducing the previous ‘for’ loop, we again filter the passes by row. At each point in the loop, we are finding the passes that correspond to each of the 24 sections of the pass flow graph; starting with the bottom left grid.

```
if(nrow(filtery)>=1){
  meanx = mean(filtery$location.x)
  meany = mean(filtery$location.y)
  meanendx = mean(filtery$pass.end_location.x)
  meanendy = mean(filtery$pass.end_location.y)
}
```

The first line of code here is an ‘if’ condition asking if we have correctly filtered all of the passes. Then, we calculate the mean beginning and end x and y locations of each pass in every individual box.

```
count= nrow(filtery)
```

We also count the number of passes that were made from each grid.

```
x<- c(meanx, meany, meanendx, meanendy, count)
```

Next, we make another variable ‘x’ of a vector with those values that we have just calculated for each grid.

```
PassFlow <- rbind(PassFlow, x) }}}}
PassFlow <- PassFlow[2:nrow(PassFlow),]
```

Adding x to our pass flow data and removing our zero entries that we began with; we now have all of the information that we want to visualise in our pass flow graph. Below is the data that we have extracted and manipulated in preparation for our graph.

```
PassFlow
```

##	x	y	endx	endy	countP
## 2	11.93750	12.475000	17.13750	20.70000	8
## 3	10.68462	34.576923	46.78462	27.03846	13
## 4	10.59000	47.726667	40.75333	40.86333	30
## 5	5.87500	72.200000	38.15000	71.97500	4

## 6	30.19091	6.763636	50.41818	13.89091	11
## 7	29.46818	29.881818	44.80000	28.18636	22
## 8	30.62667	48.813333	46.48667	46.90667	15
## 9	31.15000	72.400000	85.32500	67.67500	4
## 10	47.52667	6.230000	57.57000	16.96000	30
## 11	48.00000	28.773333	61.52000	33.60000	15
## 12	51.22857	51.528571	60.26190	46.56190	21
## 13	49.24375	70.562500	74.80625	62.30000	16
## 14	70.70000	9.713333	80.03333	17.86667	15
## 15	69.57857	29.400000	77.92143	36.25714	14
## 16	69.87273	53.809091	77.50909	50.53636	11
## 17	70.48333	71.716667	81.30833	67.55833	12
## 18	89.96000	9.910000	95.13000	19.80000	10
## 19	91.51667	23.650000	103.35000	29.05000	6
## 20	90.06154	51.007692	97.73077	47.64615	13
## 21	90.27000	69.190000	94.68000	50.83000	10
## 22	108.01667	10.966667	100.01667	16.65000	6
## 23	110.78333	25.466667	105.88333	24.26667	6
## 24	107.60000	54.333333	107.05000	55.31667	6
## 25	109.44286	74.800000	109.18571	67.10000	7

We can see the average start and end coordinates for passes in each grid as well as the count of passes. Even after reducing all of our pass data down to averages for different sections of the pitch, we can see that there is still a huge collection of data; making it difficult to ascertain a clear picture of what is happening before the graph has been made. This is why data visualization is so important.

## Plotting the graph

```
PassFlow%>%
  ggplot()+
  annotate_pitch(dimensions= pitch_statsbomb, colour="white", fill="#224C56") +
  theme_pitch()+
```

This section of code shows that we are going to use the ‘PassFlow’ data that we have prepared. We then plot a pitch with suitable dimensions for StatsBomb data which we are going to superimpose our grids and arrows onto.

```
geom_bin2d(data=PFLFC, aes(x=location.x, y=location.y),alpha=0.6, binwidth= c(bin,bin), position = "identity")
```

Using the ‘geom\_bin2d’ function, we are creating the grids on the pitch with the data from our pass flow data. Here, the input for ‘alpha’ alters the opacity of the grids, this can be whatever value works for you.

```
scale_fill_viridis()+
```

This function from the “viridis” package adds the colours for each of the sections.

```
geom_segment(aes(x=x,y=y,xend=endx,yend=endy,alpha=countP),
  color="white", lineend = "round", size=2, arrow = arrow(length=unit(0.08, "inches")))+
```

Using the ‘geom\_segment’ function, we are producing the lines representing the average length and direction. By adding alpha=countP, we show the arrows as being brighter when there are more passes played from that grid.

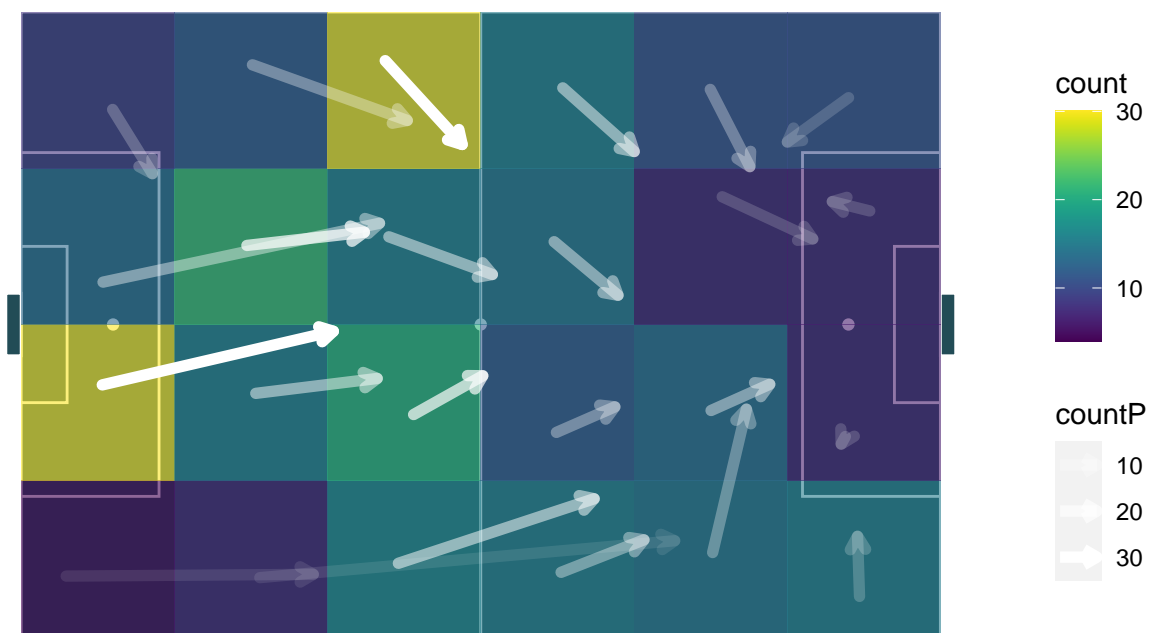
```
scale_y_reverse()+
labs(title=" Liverpool", subtitle = " vs Tottenham")
```

As usual, I have reversed the data for the y-axis and added a title for the graph.

## Interpreting Pass Flow Graphs

### Liverpool Pass Flow Graph

#### Liverpool vs Tottenham

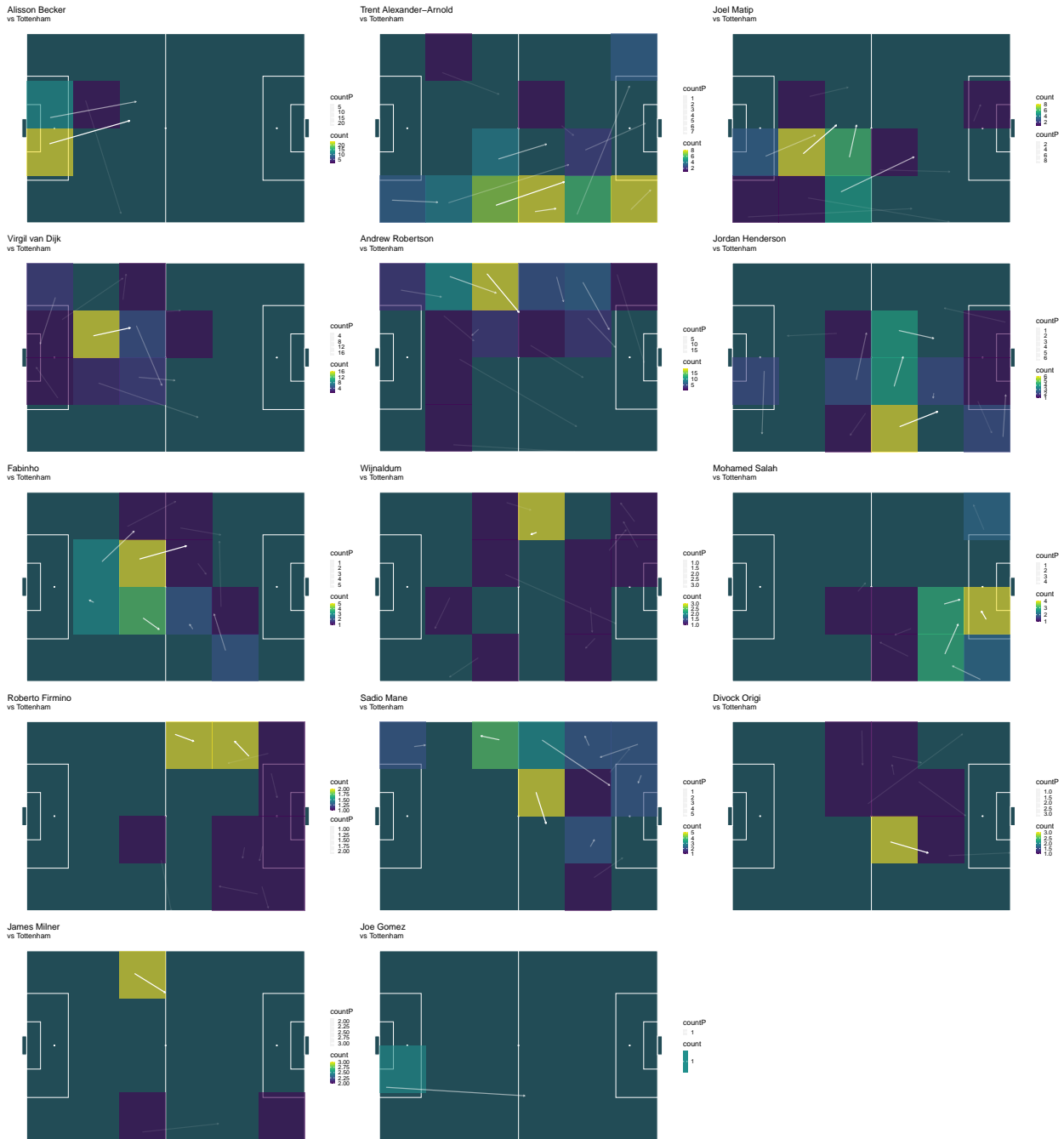


Looking at the graph, we can straight away see the areas in which Liverpool made the most passes. We can see this by looking at the colour of each grid, and also with the opacity of the white arrows. The legend on the right-hand side of the graph indicates the corresponding numerical value. For example, 30 passes were made from the grid (3,4). With the security of an early goal, most of Liverpool's passes come from defenders. Passes from defensive areas, in and around Liverpool's own box are more common than we would normally expect.

We can also easily identify that there were a lot of passes made from the section that is likely occupied by Andrew Robertson. As a fan of Liverpool, I would expect to see Andy Robertson and Trent Alexander-Arnold making many of Liverpool's passes, as I know that they like to play through their full backs for many attacks. Considering Liverpool's full backs further, you can see from the size of the arrows that the passes made on the right side of the pitch are longer, which shows that passes made by Trent Alexander-Arnold are longer on average, which is also something that you would expect to hear as a Liverpool fan. We can look further at player comparisons in the following pass flow graphs of individual players.

In the game against Tottenham, there were not many passes made by Liverpool in and around the Tottenham penalty area. We can look at the upcoming individual pass maps to see why this may be the case.

Pass Flow Graphs for Individual Liverpool Players



Looking at the pass flow graphs, it is a bit easier to see patterns. Particularly, you can get an idea of to whom a player is passing to. Maybe this was determined before the game by the manager, for example, Trent Alexander-Arnold has been told to look down the line to Mohamed Salah. On the contrary, perhaps this is just where the space opened up during the game against Tottenham.

The patterns that we see often reproduce ideas that we have already discussed from looking at the pass maps, or that we will see in passing networks. Comparing Sadio Mané and Mohamed Salah's pass flow graphs, the first thing we notice is the drastically different frequencies of passes in the different grids of the graph. Mohamed Salah is actually the furthest forward Liverpool player in terms of where his passes are made, with the majority of his passes being made in Tottenham's box. Sadio Mané, on the other hand, drops much deeper to make passes. From regularly watching Liverpool play, I can easily attribute this to each player's style of play. Sadio Mané prefers to offer himself for the ball and try to work an opportunity for himself or the team. Oppositely, Mohamed Salah's style is as a clinical finisher who occupies areas in and around the box. In fact, when comparing to the full pass flow graph for Liverpool, Mohamed Salah produces nearly all of Liverpool's passes inside the box during the game against Tottenham. Only a small number of other players contributed a few passes in this area.

Other individual pass flow graphs which stand out include both of Liverpool's box-to-box midfielders: Jordan Henderson and Georginio Wijnaldum. Looking at their pass flow graphs, knowing that they are meant to be central midfielders, you may notice that the grids which include the most passes are those very wide, close to the touchline. For Jordan Henderson, that is the right side, and for Georginio Wijnaldum, the left. Before looking at the individual pass flow graphs, I would have assumed that the passes on the right side of the pitch were made mostly by Trent Alexander-Arnold and Mohamed Salah. However, Jordan Henderson also gets involved in that area. Lastly, as previously discussed, Roberto Firmino also makes his small amount of passes out wide; occupying a different position to usual.

# Passing Networks

## What are Passing Networks?

Passing networks are another way in which we can visualise passing data. Passing networks show the average position of passes for each player with a corresponding node. These nodes have links between each other showing the frequency of passes between players. In the passing networks I have created below, a link appears when there are 3 or more passes between a pair. The thickness of these lines is altered to be thicker when there are more passes played between those players. We say that the size of each node is proportional to their eigenvector centrality, Buldú et al. (2019). This measures the importance of the players influence in the network structure. From the passing networks, we can see the locations which players occupy during passing movements; we can see which players pass to each other; and we can see how often they make these passes. Note that links between players are undirectional, meaning that both passes from player A to player B, and passes from player B to player A are represented.

## Creating Passing Networks

Below is a description of the short code used to produce a passing network in R using the ‘soccermetrics’ package from Joe Gallagher (2021).

```
install.packages("remotes")
remotes::install_github("JoGall/soccermetrics")
library(soccermetrics)
```

Use the code above to install the package. ‘Soccermetrics’ is a very useful package which is designed to be user-friendly, with a variety of useful functions to create pass networks, heatmaps and more.

```
passes %>%
  filter(team.name == "Liverpool") %>%
```

I used the ‘passes’ data from the previous pass maps example of the 2019 Champions League Final. Use the pipe operator and filter function to choose to work with just the passes made by Liverpool players.

```
soccerPassmap(fill = "red", arrow = "r",
               title = "Liverpool vs Tottenham", theme = "grass")+
```

Then, simply use the ‘soccerPassmap’ function from the soccermetrics package. There are various input options that are shown on the corresponding github page, Gallagher (2021). I have chosen to have the nodes that represent the Liverpool players filled red; there to be an arrow indicating the direction of play; used the ‘grass’ theme, and added a title.

```
scale_y_reverse()
```

As before, the y-axis must be reversed when using StatsBomb data to show the true positions of the players.

```
passes <- passes %>%
  mutate(player.name = case_when(
    player.name == "Jordan Brian Henderson" ~ "Henderson",
    player.name == "Joël Andre Job Matip" ~ "Matip",
    player.name == "Fábio Henrique Tavares" ~ "Fábinho",
```

```

player.name == "Virgil van Dijk" ~ "van Dijk",
player.name == "Georginio Wijnaldum" ~ "Wijnaldum",
player.name == "Sadio Mané" ~ "Mané",
player.name == "Andrew Robertson" ~ "Robertson",
player.name == "Trent Alexander-Arnold" ~ "Alexander-Arnold",
player.name == "Mohamed Salah" ~ "Salah",
player.name == "Alisson Ramsés Becker" ~ "Alisson",
player.name == "Roberto Firmino Barbosa de Oliveira" ~ "Firmino",
player.name == "James Philip Milner" ~ "Milner",
player.name == "Divock Okoth Origi" ~ "Origi",
player.name == "Joe Gomez" ~ "Gomez",
TRUE ~ player.name
)) %>%

mutate(pass.recipient.name = case_when(
  pass.recipient.name == "Jordan Brian Henderson" ~ "Henderson",
  pass.recipient.name == "Joël Andre Job Matip" ~ "Matip",
  pass.recipient.name == "Fábio Henrique Tavares" ~ "Fábinho",
  pass.recipient.name == "Virgil van Dijk" ~ "van Dijk",
  pass.recipient.name == "Georginio Wijnaldum" ~ "Wijnaldum",
  pass.recipient.name == "Sadio Mané" ~ "Mané",
  pass.recipient.name == "Andrew Robertson" ~ "Robertson",
  pass.recipient.name == "Trent Alexander-Arnold" ~ "Alexander-Arnold",
  pass.recipient.name == "Mohamed Salah" ~ "Salah",
  pass.recipient.name == "Alisson Ramsés Becker" ~ "Alisson",
  pass.recipient.name == "Roberto Firmino Barbosa de Oliveira" ~ "Firmino",
  pass.recipient.name == "James Philip Milner" ~ "Milner",
  pass.recipient.name == "Divock Okoth Origi" ~ "Origi",
  pass.recipient.name == "Joe Gomez" ~ "Gomez",
  TRUE ~ pass.recipient.name
))

```

When producing the pass networks, I came across an issue where lots of the players' names were unrecognisable to me, even though I knew who I was expecting to be in each position. This is because a lot of players do not go by their real names, particularly the Barcelona players. Most football fans would not recognise the names Burgos, Creus and Luján as being the masterful midfield of Busquets, Xavi and Iniesta; or even Cuccittini as being the 'GOAT' (Greatest of All Time), Lionel Messi. Therefore, I have mutated all of their names using the shown code above, Nakagawara (2019). After adjusting and running the code for each team, when you see each player's name on the pass networks, it is much more recognisable to most readers. If you need to check the full names of players, you can use the code below (adjusted for your dataset).

```
unique(passes$player.name)
```

```
## [1] "Henderson"      "Matip"          "Fábinho"        "van Dijk"
## [5] "Wijnaldum"      "Mané"           "Robertson"      "Alexander-Arnold"
## [9] "Salah"          "Alisson"        "Firmino"        "Milner"
## [13] "Origi"          "Gomez"
```

Of course, the output above now shows the nicely mutated names for the Liverpool players. Below is an example of the Barcelona players where they do not appear so familiarly.

```
unique(champpassesbarca09$player.name)
```

```
## [1] "Andrés Iniesta Luján"      "Lionel Andrés Messi Cuccittini"  
## [3] "Sergio Busquets i Burgos"  "Gnégneri Yaya Touré"  
## [5] "Gerard Piqué Bernabéu"     "Víctor Valdés Arribas"  
## [7] "Sylvio Mendes Campos Junior" "Samuel Eto\"o Fils"  
## [9] "Thierry Henry"            "Xavier Hernández Creus"  
## [11] "Carles Puyol i Saforcada"  "Seydou Kéita"
```

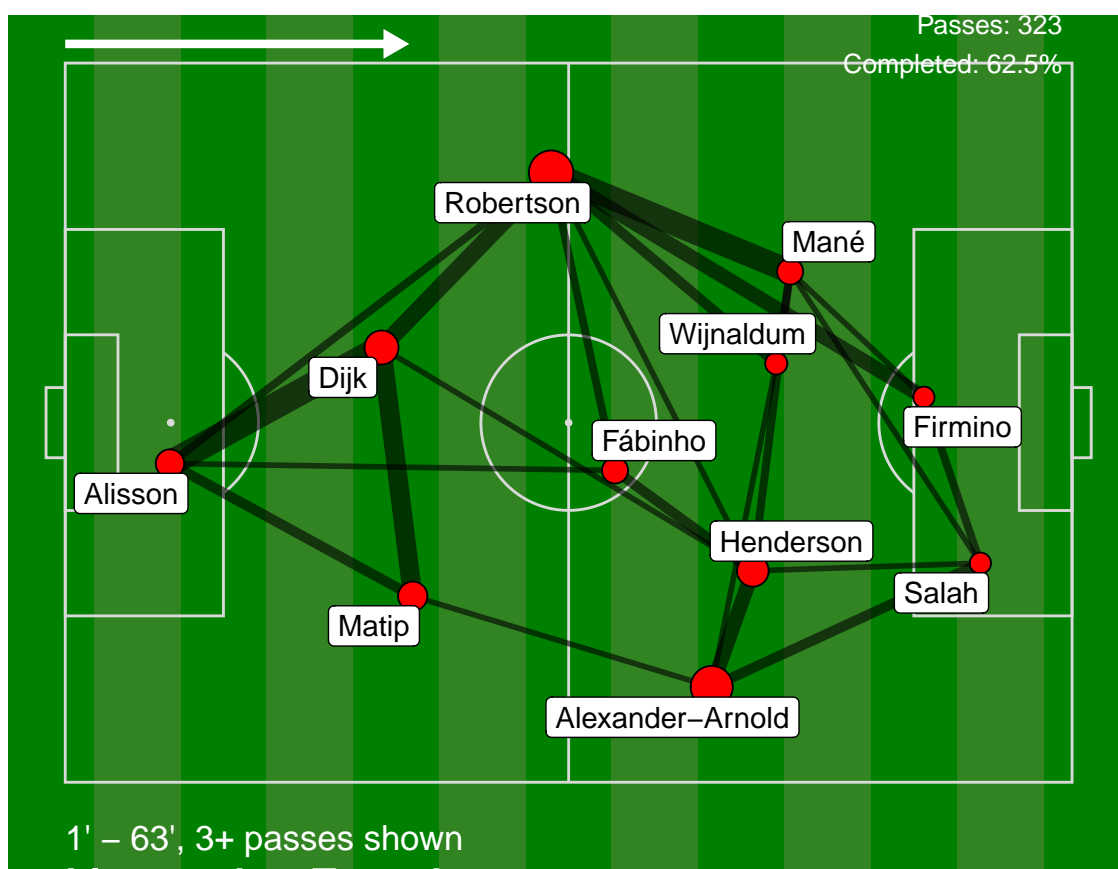


## Interpreting Passing Networks

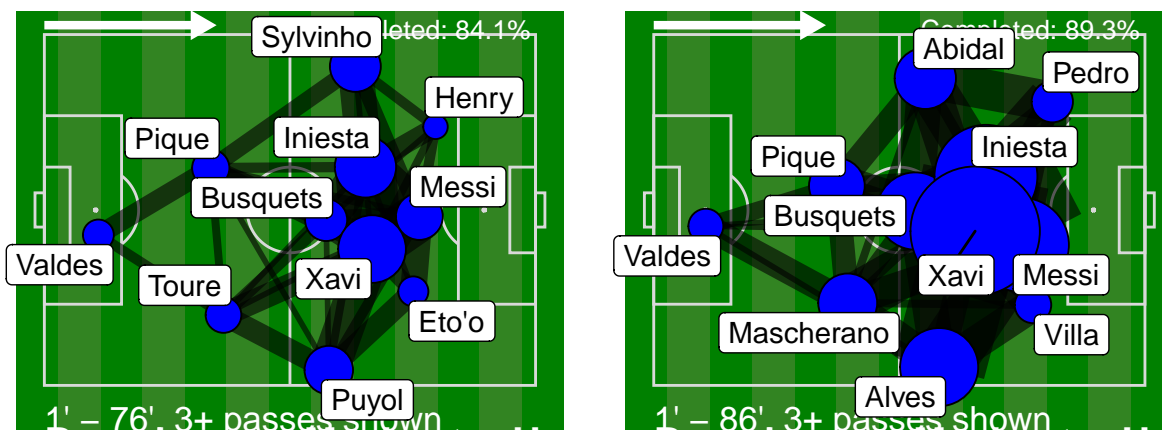
### Liverpool Passing Network

Below is the passing network that I have created for the same Liverpool game which we have been visualising. The passing network further consolidates a lot of what we have already learned from our pass maps and pass flow graphs. The furthest forward node for Liverpool is not their striker, Roberto Firmino, it is instead Mohamed Salah. Salah is known for his goalscoring ability and for scoring lots of poacher's goals in and around the box. To do this, Salah must occupy positions around these areas regularly to ensure that he is in the correct position.

We can see from the passing network that Sadio Mané has a different approach. Mané's node on the passing network is much deeper than the opposing wingers. As we have seen in previous visualisation methods, Mané made much more passes from deeper positions. From the passing network, we can also look at the links between players and how strong they are. Liverpool has 22 undirected connections between players. Andrew Robertson is involved in 7, 32% of all of Liverpool links, connected by 3 or more passes to 70% of his teammates. Surprisingly, we see that Fabinho only has links to Alisson, Robertson and Henderson. Fabinho doesn't have connections to Virgil van Dijk or Joël Matip. Liverpool's centre backs preferred to pass out wide to the full backs than to play through the midfield.



## Barcelona Passing Networks vs Manchester United

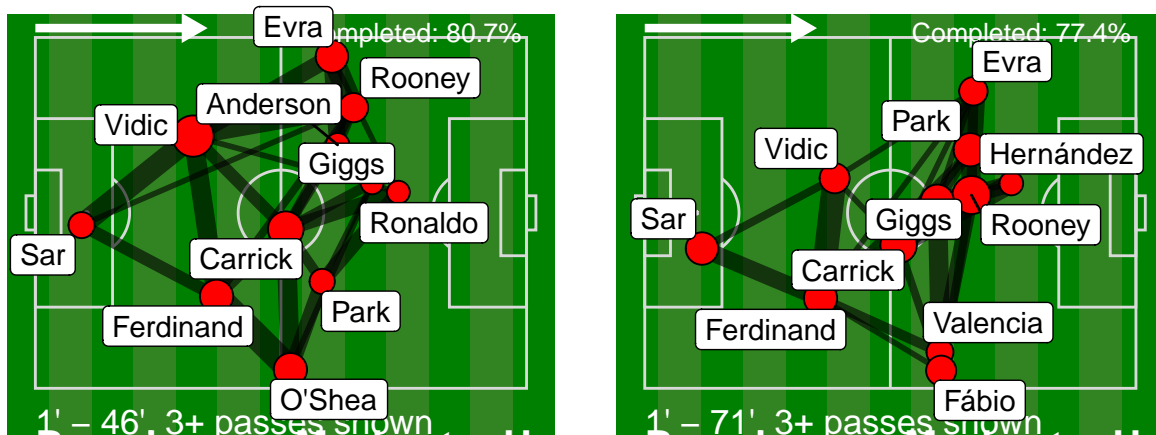


Pep Guardiola took over FC Barcelona's first team in the 2008/09 season with immediate success. Guardiola's famous 'tiki-taka' style helped Barcelona to achieve the first treble in Spanish football history; winning the La Liga, Copa del Rey and UEFA Champions League in the same season. Pep Guardiola has consistently employed the 4-3-3 throughout his managerial career. Looking at both of Barcelona's passing networks, you can see the 4-3-3 clearly. There is a clear structure and they are playing in their way, not changing tactically to cater for Manchester United. There is change of personnel, but no change of structure. The key central spine of players remains for both finals: Valdes, Pique, Busquets, Iniesta, Xavi and Messi were all untouchable for Guardiola; principally the final four. The players surrounding them remain like-for-like changes, which is shown by the similar passing network. In defence, the changed central defender is Javier Mascherano instead of Yaya Touré. Both of whom were previously central midfielders before being converted into ball-playing centre backs. In the full back positions there is still one more defensive and one slightly more offensive player, albeit they are flipped to the opposite side of the pitch. You can also see this in the current Manchester City side with a very offensive minded João Cancelo regularly venturing further up the pitch. Lastly, the extraordinary wingers on either side of Lionel Messi were replaced with new extraordinary wingers.

In the passing networks, we can see the central players made the most passes. This is contrary to what we have seen throughout this project with Liverpool's game versus Tottenham. Looking back to Liverpool's passing network, we can see this clearly. The largest node for Barcelona in both games is Xavi; but for Liverpool, the largest node is the full back Andrew Robertson, who is in an entirely different position. Both teams are playing the same formation but their passing networks look nothing alike. Compared to both Liverpool and the following Manchester United nodes, Barcelona's nodes are significantly larger. This is simply because Barcelona made significantly more passes during the game, as is the Pep Guardiola way. Additionally, Barcelona made substitutions later in the game than the other teams, meaning that their passing networks represent a longer period of time in which the passes were made.

Importantly, as we have seen, Barcelona have a clear structure and way of playing. Barcelona also have an intentional symmetry in their passing networks. The network has wide-angled triangles, which creates additional space and ensures that there are always good passing opportunities, Sumpter (2017).

### Manchester United Passing Network vs Barcelona



When looking at Manchester United's passing networks, it is much more difficult to see a clear structure or formation. In 2009 there seems to be a compact cluster of players including Evra, Anderson, Rooney, Giggs and Ronaldo. In 2011, there also seems to be a cluster in the same area. On that occasion, Ji Sung Park replaced Anderson and Hernández replaced Ronaldo. In these clusters, there are no wide-angled triangles that we discussed for Barcelona. This means that a lot of Manchester United's passing lanes could be cut more easily by Barcelona's defensive players. Patrice Evra, one of the players near the cluster, is one of the highest nodes in both of Manchester United's passing networks. This is an odd decision as we know that Lionel Messi likes to link with the right winger in and around the area that Evra has often left vacated in order to make more advanced passes. Messi was awarded Fan's Man of the Match in both the 2009 and 2011 final; and also UEFA Man of the Match in the 2011 for his performance in that area.

Comparing to Liverpool's Fábio, Michael Carrick is much better connected. In the 2009 final, Carrick had 8 connections in just the first half before the first substitution. Fábio only had 3. This shows the difference in style of play between Liverpool of 2019 and Manchester United of 2009. Liverpool typically pass out wide to Trent Alexander-Arnold and Andrew Robertson, whereas Rio Ferdinand and Nemanja Vidic pass the ball more into the midfield.

# A Geometric Approach

## Introducing Voronoi diagrams

### What are Voronoi diagrams?

A Voronoi diagram is a partition of a plane into different sections, determined by the proximity of each point of the plane. Each point (sometimes referred to as seeds, sites, or generators) has a corresponding section surrounding it called a Voronoi cell. A voronoi cell consists of all of the points of the plane that are closer to its seed than any other. The Voronoi diagram of a set of points is dual to its Delaunay triangulation, Wikipedia (2019).

### How do Voronoi diagrams relate to football?

Voronoi diagrams can be used to indicate the areas and level of control a team has over the pitch, given the position of all of their players relative to the opponents. Each seed represents a player, and each Voronoi cell represents the area that the player should be able to control. For example, if the ball was to enter a player's Voronoi cell, they should be able to reach the ball before an opponent.

Of course we cannot be 100% sure that the player whose area the ball ends up in will control the ball; but the Voronoi cells give a good idea of which team controls which area. In my generated examples, there are some variables that remain the same that in reality are not constants. Every player in my imaginary game is stationary, and their Voronoi cells are determined purely by their Euclidean distance relative to each other player.

There are scenarios that we need to consider where this is not entirely accurate. We know that if a prime 2012/13 Theo Walcott was to get a run on his teammate Per Mertesacker in training, it could be a good idea to exploit the space in behind the defender. Walcott has the advantage because of his lightning pace and the fact that he is already making the run whereas Mertesacker may still have to turn around. Likewise, if Chelsea are playing with Olivier Giroud leading the line against Liverpool, they should realise that he may struggle to get in behind Virgil van Dijk and Joe Gomez. Chelsea should instead look to utilise Giroud's other skills.

We can apply Voronoi analysis to football games in various ways. S. Kim discusses some of these different ideas. Comparing the total area of a team's collective Voronoi cells during a game or period of a game, we can determine a "dominant ratio". If a team's Voronoi area is larger than the oppositions, they have 'dominated' that team during that period, Kim (2004). Voronoi analysis can indicate which team controls the most space and has an advantageous position in the game. However, a team which 'dominates' in terms of Voronoi analysis does not guarantee that they will win the game. That is part of the beauty of football.

Another way to use Voronoi diagrams for football game analysis is by taking snapshots of an in-game situation. We can then look at where and how a team could increase their Voronoi area. Analysts will then look at what passing patterns and offensive runs can allow for this increase. A team would like to increase their Voronoi area, however, we know that different areas on the pitch have different levels of importance. Areas in and around the opposition box or beyond the defensive line while remaining onside are the most important.

## Voronoi diagrams of generated examples using Euclidean distance

### Creating the Voronoi diagrams

```
library(ggvoronoi)
library(groupdata2)
library(dplyr)
```

I have installed the packages “ggvoronoi” and “groupdata2”. As the names suggest, the former helps to plot the Voronoi diagrams, and the latter was used to group the input points into the two different teams to add colour to each point to represent the corresponding team.

```
x <- c(5,25,22,22,25,37,33,37,45,47,45,90,47,52,52,47,37,42,37,32,24,32)
y <- c(34,56,39,29,12,42,34,26,55,34,13,34,6,25,40,60,23,34,45,8,34,60)
```

The data points are input manually for each example. Firstly, I roughly plotted the points on paper through trial and error to find what coordinates are appropriate for each point in a specific formation.

```
xy <- data.frame(x, y,
                 distance = sqrt((x-100)^2 + (y-100)^2))
```

The above code forms a data frame using the Euclidean distance between the points.

```
z<-group_factor(xy, 2)
```

In order to change the colours to represent each team, I used the ‘group\_factor’ function applied to the data frame of points. When doing this for the snapshots of Tanguy Ndombele’s goal, I had to alter the grouping method to “greedy” instead of the default method. This is due to the different numbers of players per team.

```
ggplot(xy,aes(x,y, col=c("red", "blue")[z])) +
  stat_voronoi(geom="path") +
  geom_point()+
  ggtitle("4-3-3 vs 4-3-3")
```

I have plotted the points and used the ‘stat\_voronoi’ function from the “ggvoronoi” package. This plots the borders for each Voronoi cell. I have also included titles for each different formation used.

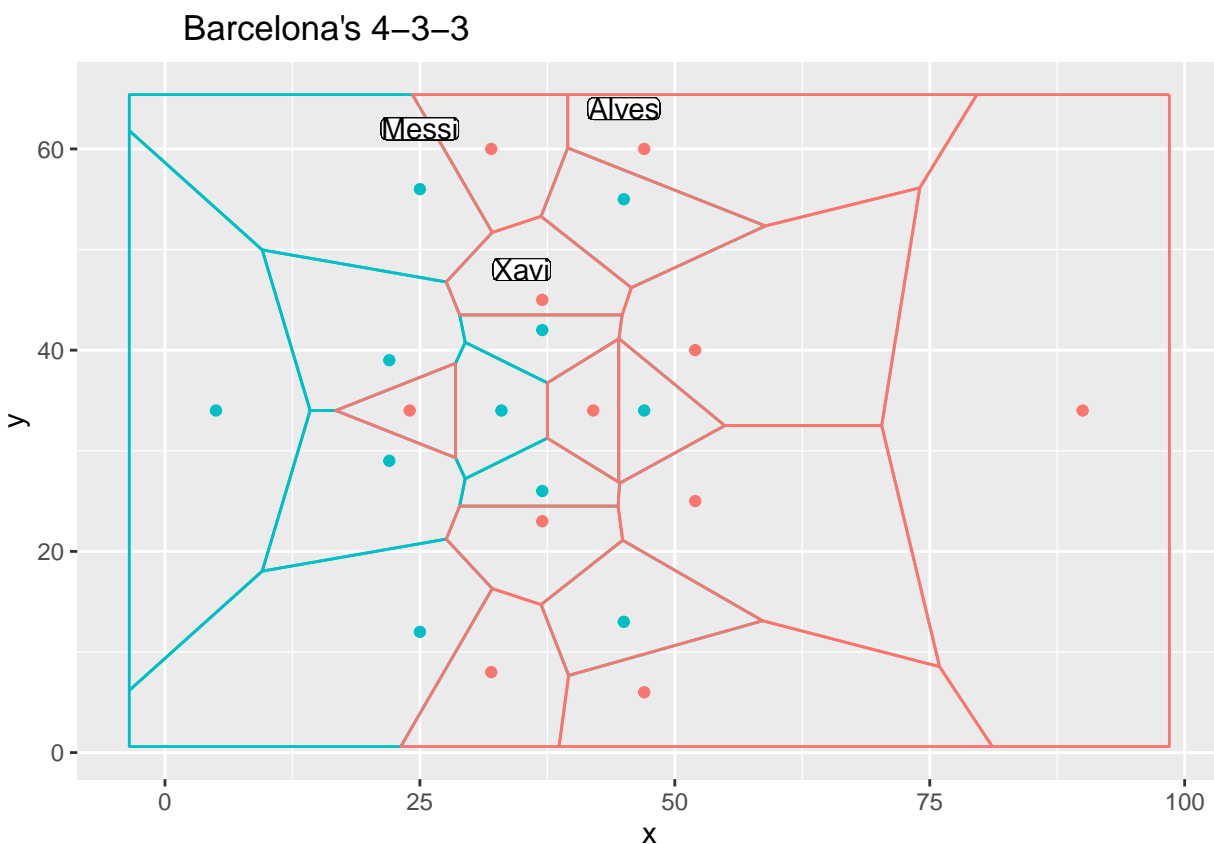
```
geom_label(
  label="Lautaro",
  x=25,
  y=23,
  label.padding = unit(0.008, "lines"),
  label.size = 0.005,
  color = "black")
```

Lastly, shown here is an example for the code that I have used to label key players in various situations.

## Comparing formations using Voronoi diagrams

I have formulated three examples where teams are in very clearly structured formations. In all cases, the red team is attacking with possession from right to left, with the defensive blue team being set up in a 4-3-3 formation in all three examples. I have imagined these examples as being situations where the red team has the ball and are trying to break down the opposition. Using S. Kim's definition, the red team 'dominates', Kim (2004). In the first phases, the blue team are mostly positioned well defensively.

In each example, I have chosen a team that have embodied that specific formation at some period in time. I have then produced examples to show how they can use passing to increase their Voronoi area. These are not carbon copies of real gameplay. However, they are examples of ways that I understand these teams play or played.

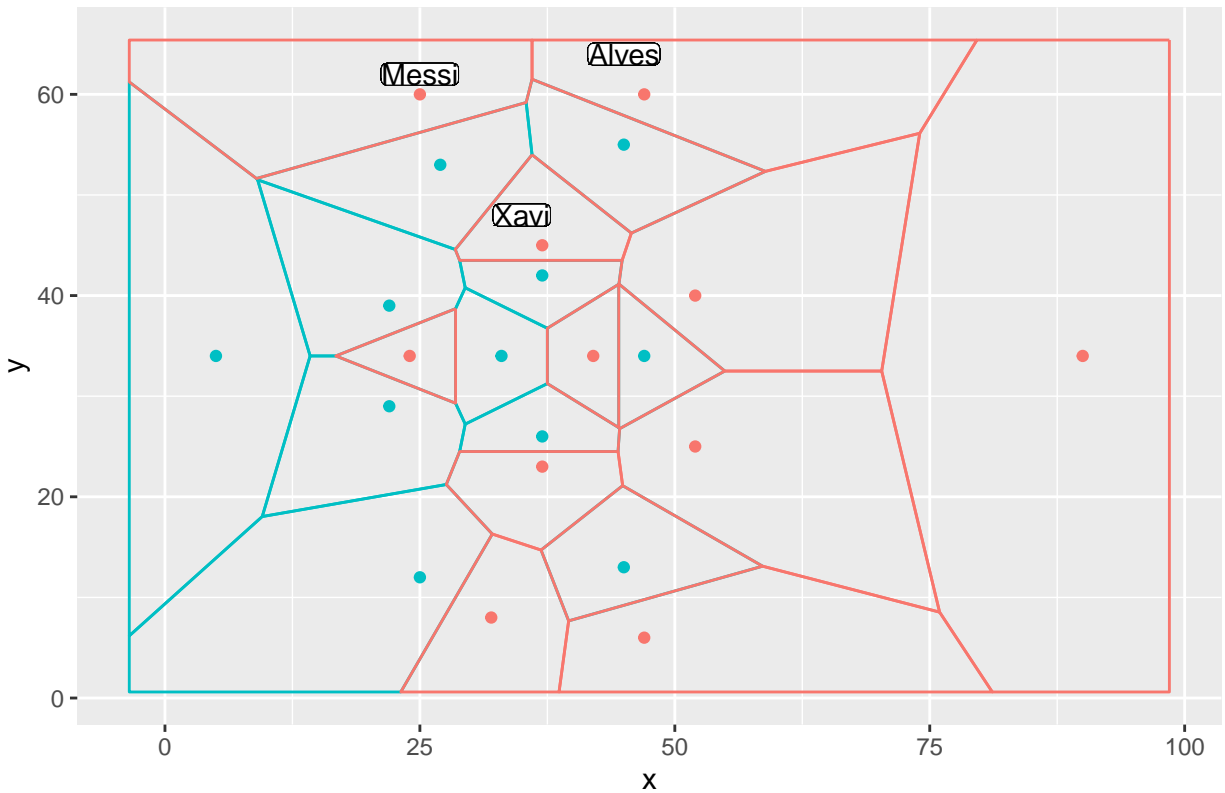


Should you play a midfield three? In the above example, both teams have three in midfield and are competing fiercely for pitch control in the middle of the park. You may hear “Win the battle in midfield and you will win the game”. But at what cost should you do this? There must be a point where gaining more control over a slightly less valuable area is worth it. When going against a team using the same formation, you must be confident in your team's ability. Otherwise, you may look to use a different formation to attack their weak points.

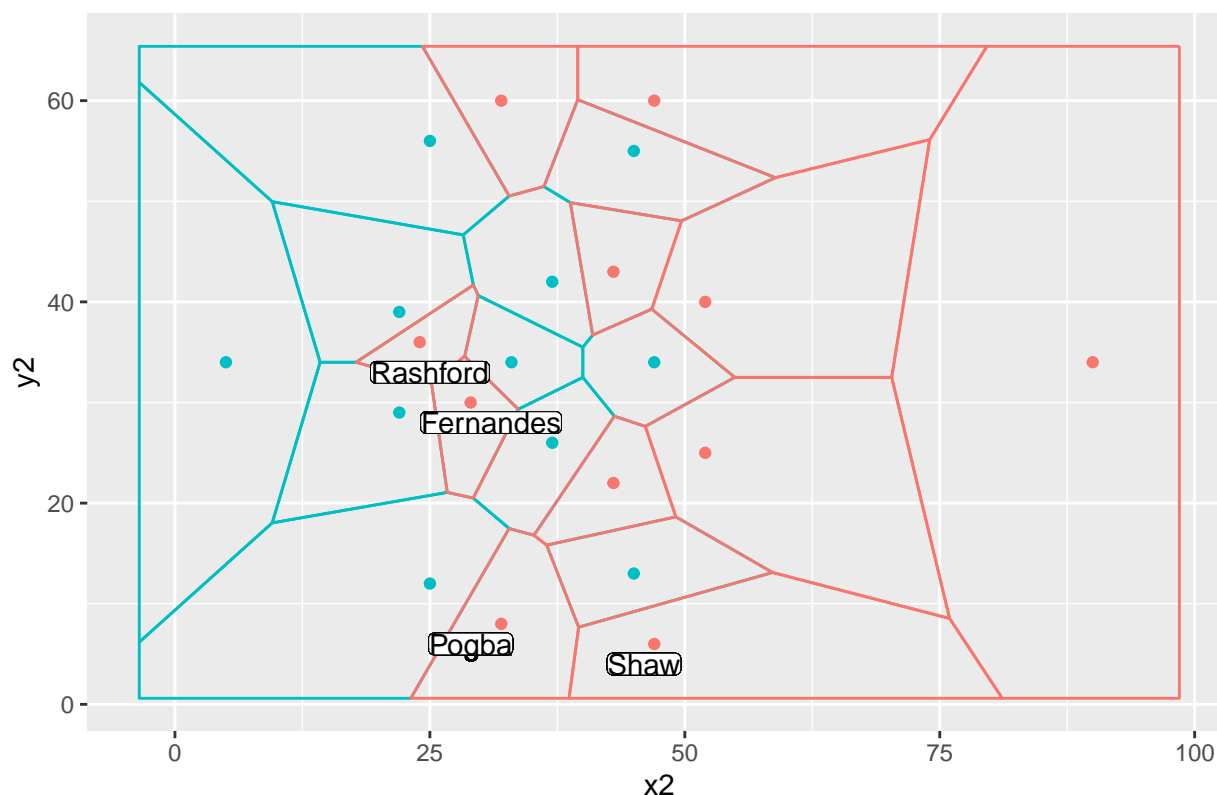
From the Voronoi diagram, we can see that the red team is pushed up quite high in the pitch, giving them more pitch control than the blue team. However, the blue team is set up well defensively with most players being goal side of their men, blocking most of red teams Voronoi cells from gaining any area beyond them. In the 4-3-3 formation, Guardiola's formation of choice, you can see lots of triangles between players. Famously, Guardiola's teams will use this tessellation to make quick 'tiki taka' passes between players, hoping to lure opposition players out of position. In this case, we can imagine quick passes between Lionel Messi, Xavi and

Dani Alves on the right side of the pitch (labelled on the Voronoi diagram). If the blue team's left back gets dragged in to try to intercept a pass, Messi can make a run beyond him to increase the area of his Voronoi cell. This is exactly what Pep Guardiola's players are trying to do when they are playing those short quick passes. In the next Voronoi diagram, we can see what happens to Lionel Messi's Voronoi cell in such a scenario.

Barcelona's 4-3-3



## Manchester United's 4-2-3-1

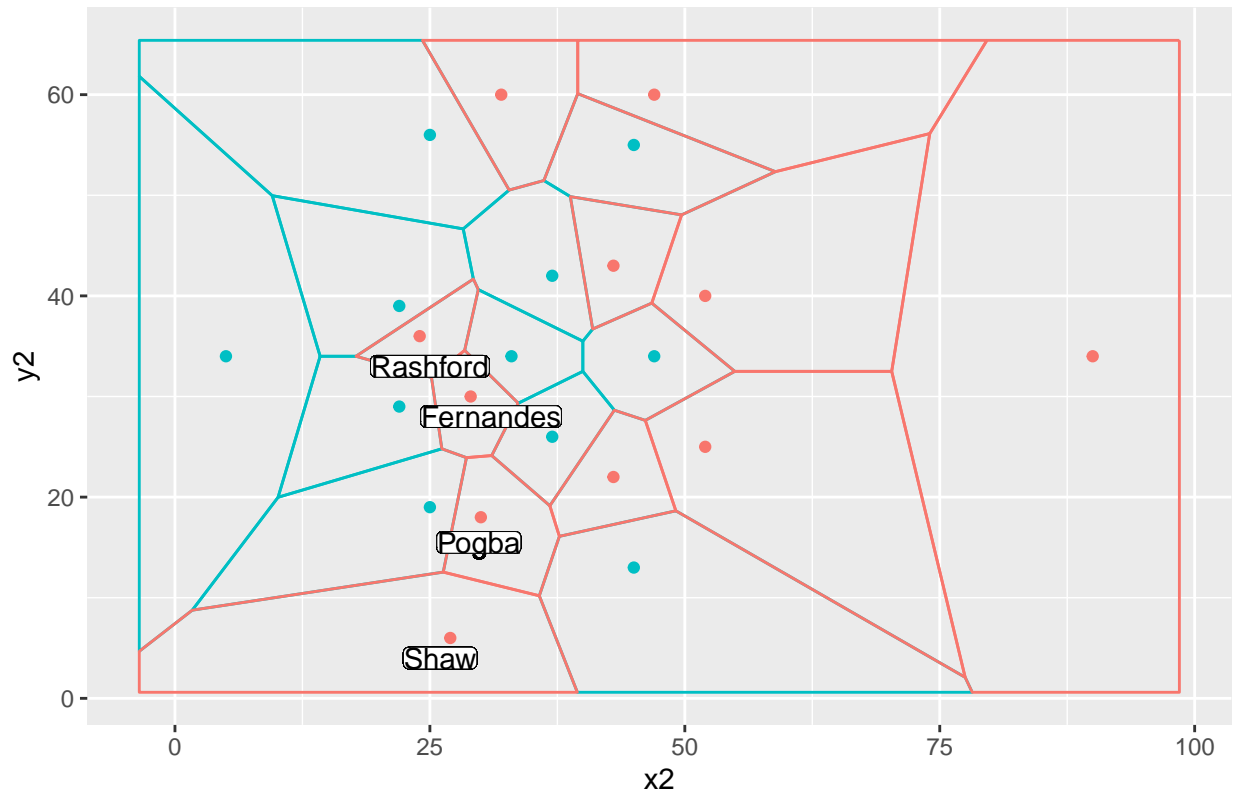


When the offensive team instead deploys a 4-2-3-1 formation, they are trying to gain more pitch control between the midfield and the defence. 4-2-3-1 has arguably been the most popular formation over the last decade. The 4-2-3-1 allows for passing through triangles, breaking down opponents more effectively than other formations where most passing options are sideways or risky long balls; like the 4-4-2. We can see in the Voronoi diagram that the defensive 4-3-3 formation controls a lot of the midfield. However, if the attacking team can get the ball into the number 10 and bring support from wide players, there is plenty of space to be created and found.

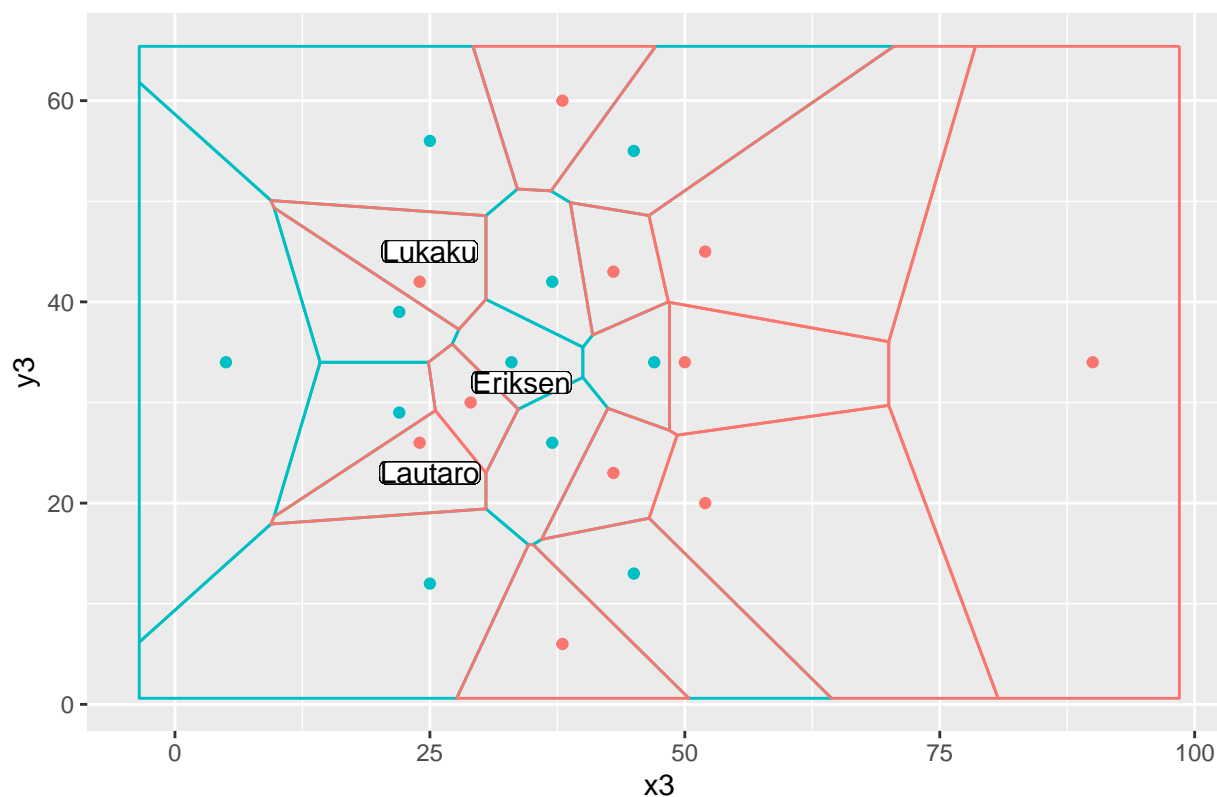
In the 4-2-3-1 example above I have chosen to represent Ole Gunnar Solskjær's Manchester United. After a difficult period under ex-manager José Mourinho, Luke Shaw has enjoyed a resurgence for Manchester United. I have shown an example below of how Solskjær sets up Manchester United in the 4-2-3-1 formation to create opportunities for Luke Shaw. Manchester United's record £95 million signing Paul Pogba has also found a place in this formation, often on the left side of midfield. Below, I have shown how Pogba drifts inside, creating space for Luke Shaw to run into. Luke Shaw can then gain control over a large area on the left flank, which you can see from the following Voronoi diagram. Luke Shaw then often gets played through, given an opportunity to cross the ball to Marcus Rashford or Edinson Cavani.



# Manchester United's 4-2-3-1



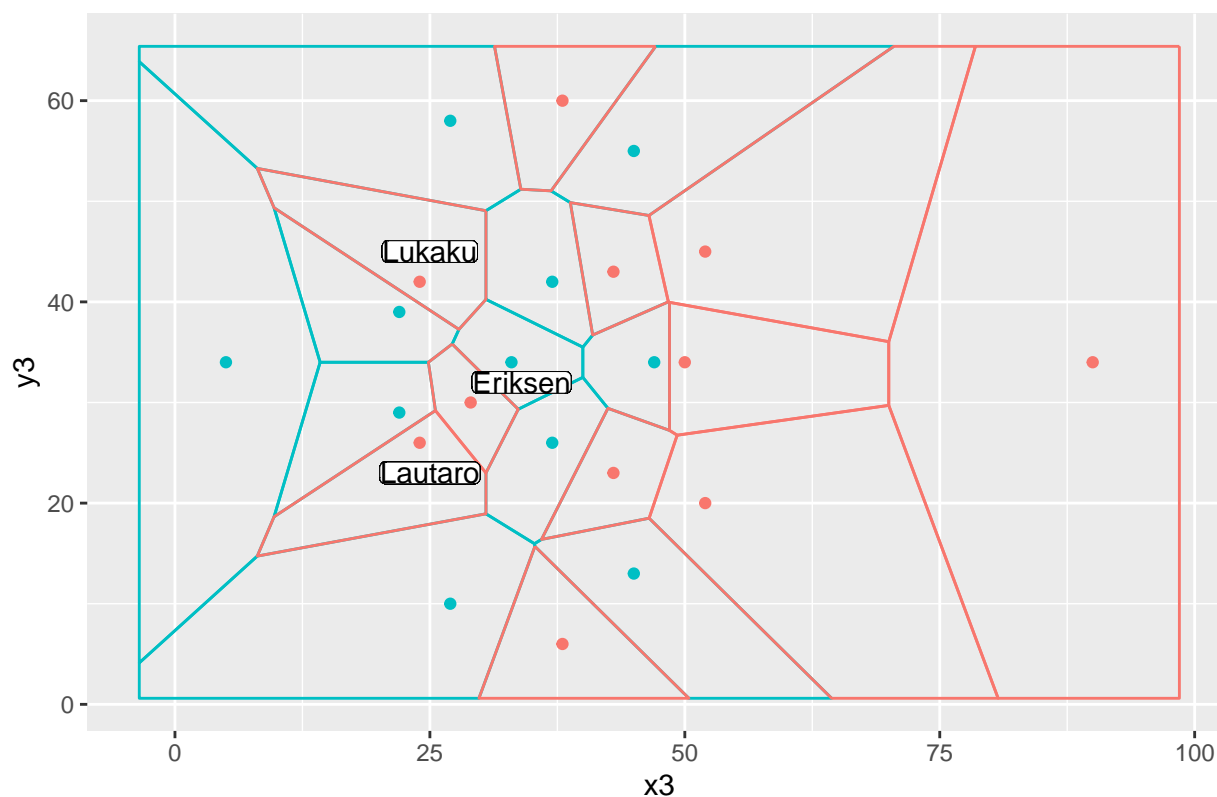
## Inter Milan's 3-5-2



Most popular in Italy, the 3-5-2 formation is a very interesting formation. Antonio Conte, the modern mastermind of the formation, recently lead Inter Milan to their first Serie A title for 11 years. The 3-5-2 formation allows for two strikers and a number 10 in behind, finding as much pitch control as possible high up the pitch. You can see from the Voronoi diagram that Romelu Lukaku and Lautaro Martínez' Voronoi cells already leak past the two centre backs. With two strikers and an attacking midfielder, there is a lot of pressure on the oppositions centre backs. As you can see in the second Voronoi diagram; if the full backs of the defensive team are even slightly dragged towards the attacking teams wing backs, a lot of space can open up for the strikers who can be left 2 vs 2 with the defenders. This can leave the oppositions defenders very exposed.

As I move forward with independent research in the future, I would like to further investigate the value of different Voronoi areas on a football pitch. I think there is something to be seen in quantifying the value of each area, finding the optimal spatial composition (big or small) to counter your oppositions formation.

## Inter Milan's 3–5–2



## Snapshots of Voronoi diagrams from real game situations

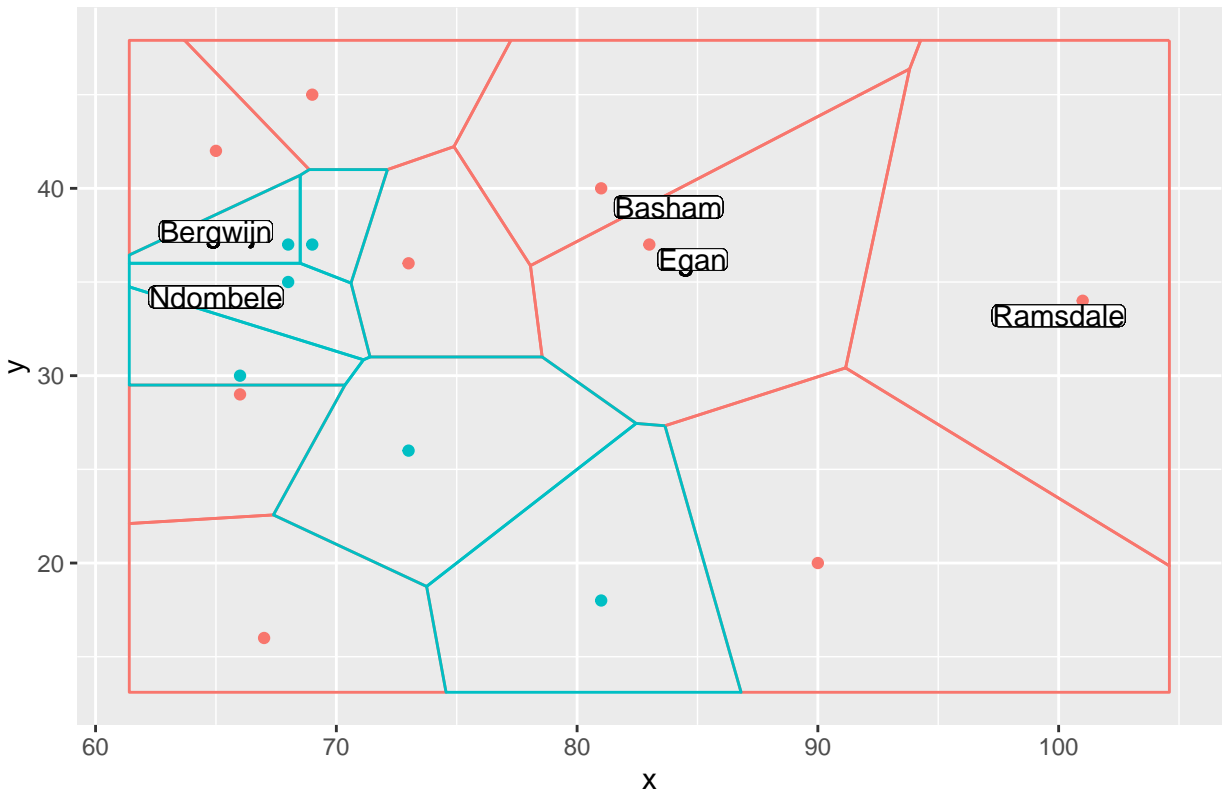
### Tanguy Ndombele's goal vs Sheffield

I have created Voronoi diagrams for two snapshots in the build-up to Tanguy Ndombele's goal against bottom of the table Sheffield United. This goal is a perfect example because it shows the effect that simple passing actions can have on pitch control. After a one-two between Steven Bergwijn and Ndombele, Tottenham gain pitch control over a key central area of the pitch. This gain of pitch control allows them to play a cute chip into the space which leads to a big chance which Ndombele converts impressively. Due to copyright reasons, I have removed pictures of these snapshots. I would recommend watching the goal on Youtube at <https://www.youtube.com/watch?v=V0NvijWPpq8>, Football (2021), or by searching 'Ndombele scores stunning lob in Spurs win'. I believe this would be beneficial to supplement any reader's understanding of these Voronoi diagrams.

Note that the Voronoi diagrams are not centred with the goal. Sheffield's goalkeeper Aaron Ramsdale is a good reference point to where the net is.

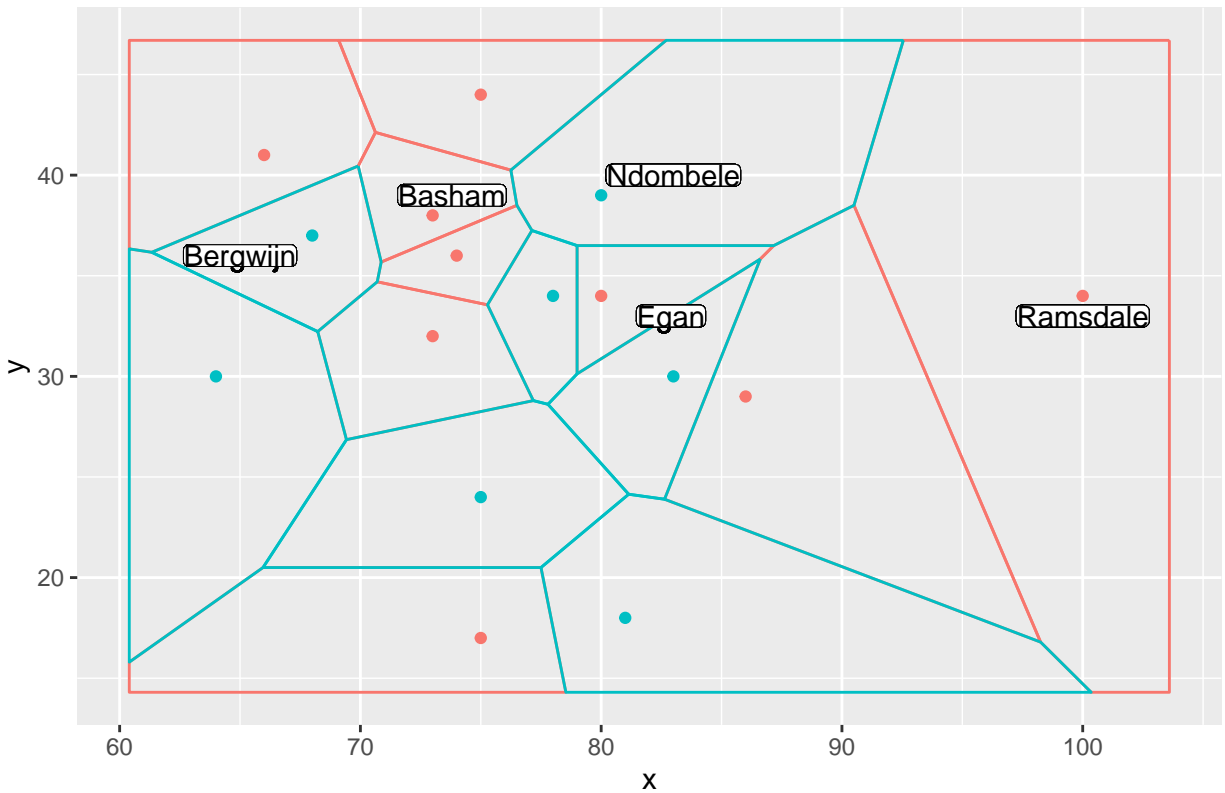
The following Voronoi diagram shows the moment where Tottenham regain possession through Reguilon.

## Build up to Ndombele Goal before one-two



As we can see from this snapshot; Tottenham have some options. Sergio Reguilon intercepts the ball after a mix up between Sheffield's Oliver Norwood and Ethan Ampadu. In an attempt to quickly recover the ball, Chris Basham rushes out from the defensive line. In doing this, Basham leaves a space in behind that Tottenham can take advantage of. On the right side of the pitch, there is Harry Kane and Serge Aurier. However, if the ball was played out wide, it would allow Sheffield United's defense some time to recover and regain their shape. After a couple of sharp, short passes by Tottenham; Tanguy Ndombele passes to Steven Bergwijn and immediately makes a darting run into the space vacated by Basham. Ndombele then gains control over that area, which is represented by his next Voronoi cell. The next Voronoi diagram shows the moment where Steven Bergwijn is playing the smart return ball over Sheffield's defence to Ndombele swiftly after receiving it.

## Build up to Ndombele Goal after one-two



In the new Voronoi diagram, you can see the positions of Basham and Ndombele's corresponding cells has changed drastically. After the one-two has been played, and Ndombele makes the run into the space vacated by Basham, Ndombele's new Voronoi cell is much larger than before. This is due to both his movement and the mistake of Basham in his eager attempt to recover possession. With Ndombele controlling a key central area beyond most of Sheffield's erratic defensive line, Tottenham have a great opportunity if they can capitalise on it. To the delight of Tottenham fans, Bergwijn's pass is weighted well and Ndombele smartly chips the ball over Aaron Ramsdale with the outside of his right boot.

## Discussions and Conclusions

In this project, I have introduced ways that we can visualise aspects of a football game; predominantly discussing passing. Visual maps and graphs are an increasingly central facet of football analysis. Presenting these graphs continually proves to be an effective way to communicate information about a game or season. Analysis of data in football is developing rapidly and it is important that analysts can efficiently express their findings to non-analysts: football coaches, players and pundits. I believe that the best way to do this is by preparing visual aids like pass maps, pass flow graphs, passing networks and Voronoi diagrams.

## A Note Regarding References

In preparation and parallel to this project, I have read the following books: *Soccermatics*, Sumpter (2017), *Zonal Marking*, Cox (2019), *Data Analytics in Football*, Memmert and Raabe (2018) and *Football Hackers*, Biermann (2019). I have made every effort to reference these books where applicable.

## Bibliography

- Biermann, Christoph. 2019. *FOOTBALL Hackers : The Science and Art of a Data Revolution*. Blink Publishing.
- Buldú, J. M., J. Busquets, I. Echegoyen, and F. Seirul.lo. 2019. “Defining a Historic Football Team: Using Network Science to Analyze Guardiola’s F.c. Barcelona.” *Scientific Reports* 9 (September). <https://doi.org/10.1038/s41598-019-49969-2>.
- Cox, Michael. 2019. *Zonal Marking : From Ajax to Zidane, the Making of Modern Soccer*. Bold Type Books.
- “FCrSTATS/Sbpitch.” 2021. GitHub. <https://github.com/FCrSTATS/SBpitch>.
- Football, Sky Sports. 2021. “Ndombele Scores Stunning Lob in Spurs Win!| Sheff Utd 1-3 Tottenham | Premier League Highlights.” [www.youtube.com](http://www.youtube.com). <https://www.youtube.com/watch?v=V0NvijWPpq8>.
- Gallagher, Joe. 2021. “SoccerPassmap: Draw a Passing Network Using Statsbomb Data in Jo-gall/Soccermatics: Visualise Tracking and Event Data from Soccer Matches.” [rdrr.io](http://rdrr.io). <https://rdrr.io/github/JoGall/soccermatics/man/soccerPassmap.html>.
- Kim, S. 2004. “Voronoi Analysis of a Soccer Game.” *Nonlinear Analysis: Modelling and Control* 9 (July): 233–40. <https://doi.org/10.15388/NA.2004.9.3.15154>.
- Memmert, Daniel, and Dominik Raabe. 2018. *Data Analytics in Football : Positional Data Collection, Modelling and Analysis*. Routledge.
- Mishra, Abhishek Amol. 2021. “Creating a Pass Flow Graph in R.” [www.youtube.com](http://www.youtube.com). [https://www.youtube.com/watch?v=uOE2\\_UIRniE](https://www.youtube.com/watch?v=uOE2_UIRniE).
- Nakagawara, Ryo. 2019. “Visualizing Soccer with Statsbomb Data and R, Part 1: Simple xG and Pass Partner Plots!” [ryo-n7.github.io](http://ryo-n7.github.io). <https://ryo-n7.github.io/2019-08-21-visualize-soccer-statsbomb-part-1/>.
- Sumpter, David. 2017. *Soccermatics*. Bloomsbury Publishing Ltd.
- Wikipedia. 2019. “Voronoi Diagram.” Wikipedia; Wikimedia Foundation. [https://en.wikipedia.org/wiki/Voronoi\\_diagram](https://en.wikipedia.org/wiki/Voronoi_diagram).