-Final Report-

Tottenham Hotspur F.C. Scouting Problem

Youngchan Kim, Kunwoo Kim, Woojae Jeon, Taehyun Yang

I. Background: Overview & Problem Definition

Scenes from the movie *Moneyball*

In the movie *Moneyball* Oakland Athletics' General Manager Billy Beane utilized sabermetrics to evaluate his potential roster by performing data mining on hundreds of individual baseball players, identifying statistics that were highly predictive of how many runs a player would score, a number not typically among those valued by baseball scouts in the day. In doing so, Beane sought lower-cost ballplayers with high on-base percentages, as opposed to high-priced home run hitters with high batting averages. His theory was that players who have a higher on-base percentage would be more valuable, even though they may hit fewer home runs. The efforts proved remarkably effective. Beane was able to utilize predictive analytics to develop a highly successful team despite operating under a tight budget.

Digitization and the pandemic are forcing football clubs to rethink, digitize and improve their scouting processes. Due to the continuously increasing amount of available data and videos, it is nowadays possible to combine a systematic selection with a standardized scouting process, even with limited resources. Data Analytics does not aim to replace scouts. Rather, data analytics provides a valuable complement to the talent identification skills of scouts. A good scouting process consists of several phases and begins with data scouting.

Tottenham Hotspur F.C. is a football club based in London and competes in the Premier League. The club have won the league only twice about 60 years ago and ranked fourth this season(2021-22). Tottenham needs to improve the squad to go higher next year.

Every year after the season, the scouting team needs to find new players to scout and decide which players to release from the original squad. Transfers contain huge risk due to the information asymmetry. Mr.Tot, the scouting manager, should consider two main points to minimize uncertainty and make the best transfer. 1) Whether the player is under-valuated or not, 2) whether the player's performance can contribute to improving team performance.

Mr.Tot wants to implement data analytics in the scouting process. Let's help out Mr.Tot as a research assistant. The scouting process will go through four big steps. 1) First, you will obtain information about players in the EPL market. This contains valuation, ratings, and other performance statistics. Through obtained information you will find players that are under-valuated. 2) Second, you will analyze the team's games this season and see what factors were the crucial to team's victory. 3) Third, you use linear programming to decide players to scout under certain constraints. 4) Last, you simulate next season with the new squad and estimate next year's victory points.

II. Questions

1. First of all, you decided to make a full list of under-valuated players from which you will choose players to scout. Use the given dataset.

    a. Conduct a multiple linear regression analysis using given data sheet. Dependent variable should be 'valuation in 10 thousand pounds' while all the other variables be independent variables. However, you have to figure out which independent variables are valid predicting the dependent variable. Using 10% confidence level, detect independent variables which are not valid.
    b. Conduct a multiple linear regression analysis one more time, after removing invalid variables found in (question 1.a.). How much of the variance of dependent variable is explained by the revised model? If the model doesn't explain well, what would be the reason? Briefly explain your thoughts.
    c. From the regression model built in (b.), calculate each player's predicted value and add it in the list using a new column on the sheet.
    d. Divide each player's value by their predicted value and add it in the list. Sort the list by V/PV in ascending manner. Wipe out ones whose values are over estimated.

2. As a team analyst for Tottenham Hotspur F.C., you would like to know how to improve your team's chances of winning. More specifically, you would like to know which factors in game play the biggest role in winning a game. You are given the data for every Tottenham player's statistics for every game played last season. A portion of the data is shown below:

		Summary																										
								Performance						Expected			SCA		Total					Short			Medium	
ID	Matchweek	Player	#	Nation	Pos	Age	Min	Gls	Ast	PK	PKatt	Sh	SoT	xG	npxG	xA	SCA	GCA	Cmp	Att	Cmp%	TotDist	PrgDist	Cmp	Att	Cmp%	Cmp	Att
12	1	Japhet Tanganga	25	eng ENG	RB	22	82	0	0	0	0	0	0	0	0	0.3	1	0	29	32	90.6	561	256	16	17	94.1	9	9
13	1	Matt Doherty	2	ie IRL	RB	29	8	0	0	0	0	0	0	0	0	0	0	0	3	6	50	44	11	1	1	100	1	3
14	1	Hugo Lloris	1	fr FRA	GK	34	90	0	0	0	0	0	0	0	0	0	0	0	17	27	63	847	675	3	3	100	1	1
15	2	Son Heung-min	7	kr KOR	FW	29	71	0	0	0	0	2	1	0.2	0.2	0	1	0	10	18	55.6	122	12	8	8	100	2	6
16	2	Harry Kane	10	eng ENG	FW	28	19	0	0	0	0	1	1	0.3	0.3	0.1	2	0	7	10	70	81	34	5	6	83.3	2	3
17	2	Steven Bergwijn	23	nl NED	LW	23	89	0	0	0	0	2	2	0.1	0.1	0.3	4	0	19	22	86.4	235	53	13	13	100	5	6
18	2	Harry Winks	8	eng ENG	CM	25	1	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0		0	0
19	2	Lucas Moura	27	br BRA	RW	29	66	0	0	0	0	0	0	0	0	0	0	0	12	15	80	188	96	6	7	85.7	6	8
20	2	Giovani Lo Celso	18	ar ARG	RW	25	24	0	0	0	0	0	0	0	0	0.1	1	0	10	12	83.3	145	50	6	6	100	3	3
21	2	Dele Alli	20	eng ENG	CM,LW	25	90	1	0	1	1	0	0	0.8	0	0	1	1	26	32	81.3	414	127	13	14	92.9	10	10
22	2	Pierre HÃ¸jbjerg	5	dk DEN	CM	26	90	0	0	0	0	1	0	0.1	0.1	0	1	0	42	45	93.3	653	162	24	24	100	15	15
23	2	Oliver Skipp	29	eng ENG	DM	20	90	0	0	0	0	0	0	0	0	0	0	0	36	44	81.8	596	124	20	21	95.2	15	17
24	2	Sergio ReguilÃ³n	3	es ESP	LB	24	90	0	0	0	0	0	0	0	0	0	1	1	29	44	65.9	461	319	14	15	93.3	14	24
25	2	Eric Dier	15	eng ENG	CB	27	90	0	0	0	0	1	1	0.1	0.1	0	0	0	29	38	76.3	691	186	6	7	85.7	18	20
26	2	Davinson SÃ¡nchez	6	co COL	CB	25	90	0	0	0	0	0	0	0	0	0	0	0	30	33	90.9	680	253	4	4	100	22	23
27	2	Japhet Tanganga	25	eng ENG	RB	22	90	0	0	0	0	0	0	0	0	0	0	0	20	34	58.8	289	147	14	20	70	4	10
28	2	Hugo Lloris	1	fr FRA	GK	34	90	0	0	0	0	0	0	0	0	0	1	0	19	29	65.5	728	527	3	3	100	6	6
29	3	Harry Kane	10	eng ENG	FW	28	90	0	0	0	0	1	1	0.1	0.1	0.3	3	0	21	26	80.8	367	91	10	10	100	7	9
30	3	Steven Bergwijn	23	nl NED	LW	23	67	0	0	0	0	2	0	0.1	0.1	0	2	1	21	25	84	293	91	14	14	100	5	6
31	3	Lucas Moura	27	br BRA	RW	29	23	0	0	0	0	1	1	0	0	0	0	0	8	12	66.7	126	48	5	8	62.5	3	4
32	3	Son Heung-min	7	kr KOR	RW,LW	29	87	1	0	0	0	3	2	0.2	0.2	0.1	3	0	32	47	68.1	530	113	19	22	86.4	10	13
33	3	Bryan	11	es ESP	LW	20	3	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0		0	0

a. As the team analyst, you would want to know what areas as a team need improvement based on history of previous matches. In order to do so, you want to know the team sum of all the stats for each match. For example, you want to know how many total goals Tottenham scored for match 2 or the total goal creating action (GCA) done by all team members to obtain insight. Create a table that lists total stats per match. (Keep in mind that some stats such as cmp%[% of completed passes], tkl% [%of tackle done successfully] are in %. Also, you can exclude individual stats: Nation, #, Pos, and Age)

b. You received more data collected by your junior workers (aka student interns) about each match details. A portion of the data is shown below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Date	Time	Round	Venue	Result	GF	GA	Opponent	xG	xGA	Poss	Attendance	Captain	Formation	Referee
2	2021/08/15	(00:30)	1	Home	W	1	0	r City	1.3	1.9	36	58,262	Hugo Lloris	2004/03/03	Taylor
3	2021/08/22	(22:00)	2	Away	W	1	0	Wolves	1.5	1.4	43	30,368	Hugo Lloris	2004/03/03	Attwell
4	2021/08/29	(22:00)	3	Home	W	1	0	Watford	0.9	0.6	56	57,672	Hugo Lloris	4-2-3-1	Marriner
5	2021/09/11	(20:30)	4	Away	L	0	3	Palace	0.1	2.4	38	26,000	Hugo Lloris	4-4-2◆	Moss
6	2021/09/19	(00:30)	5	Home	L	0	3	Chelsea	0.7	1.9	47	60,059	Hugo Lloris	4-3-2-1	Tierney
7	2021/09/26	(00:30)	6	Away	L	1	3	Arsenal	1	0.8	54	59,919	Hugo Lloris	2004/03/03	Pawson
8	2021/10/03	(22:00)	7	Home	W	2	1	Aston Villa	1.7	1.1	58	53,076	Hugo Lloris	4-2-3-1	Kavanagh
9	2021/10/17	(00:30)	8	Away	W	3	2	Utd	1.6	0.5	63	52,214	Hugo Lloris	4-2-3-1	Marriner
10	2021/10/24	(22:00)	9	Away	L	0	1	West Ham	0.7	1.2	61	59,924	Hugo Lloris	4-2-3-1	Tierney
11	2021/10/30	(01:30)	10	Home	L	0	3	r Utd	0.7	1.2	58	60,356	Hugo Lloris	4-2-3-1	Attwell
12	2021/11/07	(23:00)	11	Away	D	0	0	Everton	0.8	0.5	56	39,059	Hugo Lloris	2003/04/03	Kavanagh
13	2021/11/21	(01:30)	12	Home	W	2	1	United	2	1.1	43	58,989	Hugo Lloris	2003/04/03	Marriner
14	2022/02/23	(04:30)	13	Away	L	0	1	Burnley	0.9	1.1	64	19,488	Hugo Lloris	2003/04/03	Scott
15	2021/12/02	(04:30)	14	Home	W	2	0	Brentford	2.4	0.3	47	54,202	Hugo Lloris	2003/04/03	Moss

Using this data, calculate the average stats of all matches won and all matches not won (draw and losses). Then calculate the % difference that ranges from -1 to 1 where a positive number indicates a positive correlation of the stat, the higher the chance of the team winning and a negative number indicates a negative correlation of the stat.

c. Sort the stats in ascending order and provide your insight into how Tottenham generally wins their games, why Tottenham doesn't win some of their games, and how Tottenham can potentially improve their chances of winning. List the 5 most detrimental factors and 5 most helpful factors that Tottenham can possibly work towards increasing their chances of winning.

3. To improve the club's performance, the managerial board asks for your advice on which players to scout during transfer season. The club can scout up to 2 players max for the next season, and wishes to primarily scout players that are currently undervalued in the league (found in question 1). Also, the team wishes to scout players that can improve the club's weaknesses (top 5 stats that have negative correlation with winning) and maintain or increase the team's strong points (top 5

stats that have positive correlation with winning). For this question, you will use the answers from question 1 and question 2 to suggest to the managerial board which new players to scout. Assume that Tottenham's goalkeeper Hugo Lloris has a life-long contract, so you can only scout defenders, midfielders, and forwards.

a. Suppose that you wish to improve the team's core stat derived in question 2, while minimizing the budget. That is, you wish to find the cheapest way to scout players so that upon scouting, the team's aggregate strong stat increases or the team's weakness related stat decreases. There are 2 additional constraints in this problem. 1) You can scout up to 2 players from the undervalued players pool derived in question 1. 2) Tottenham uses a 3-4-3 formation, meaning if you wish to scout a midfielder, you have to let go of an existing midfielder from the club. You cannot change the formation. Which players should Tottenham seek to scout during the transfer season?
b. Now suppose you get greedy with your improvements. The club wishes to scout players such that all your strong stats improve by 5%, while your weak stat is decreased by 5%. Which players should Tottenham sell and scout to achieve this with the cheapest budget?
c. Now suppose the club does not care about minimizing budget. You can scout any players from the undervalued players pool. Re-solve the problem to find out the maximum team stat the team can achieve (Note that the 2 additional constraints still hold).

4. With the new squad derived in question 3, you will use simulation to predict Tottenham's league points next season. Assume that there are 38 matches per season, and the team/players' individual stats will not change next season (it is safe to use only 2021-22 season data for simulation). Follow these simulation steps.

[Simulation Steps]

1) Extract the new squad's players info data to a new spreadsheet. This spreadsheet should contain. The players' info contains each match data of the player. Only extract significant stats derived in question 2. For example, if the new squad contains Harry Kane, you will need to extract 10 columns of Kane's 2021-22 stats per match. Note that not all players appeared in all 38 matches.
2) Create a pivot-table to calculate each player's average and standard deviation per significant stat. The pivot-table will contain 10 rows of player names and 20 columns in total for the average and standard deviation of each stat.
3) Now aggregate the player's stat info. For example if Harry Kane's Shots on Target was 2 on average and Son's Shots on Target was 4 on average, we can assume that the total squad's Shots on Target is 6, if other player's Shots on Target is 0. Use the SUM function to aggregate the averages, and DEVSQ function to aggregate the standard deviations.
4) Simulate 10 columns worth of stat info for 38 games. Use averages and standard deviations calculated in step 3. For negative values, change it to 0.
5) Create a pivot-table of Tottenham's teams stat derived in question 2-b, but with only the relevant 10 stats needed. The cells in this pivot-table is the average of each stat per match outcome.
6) You wish to use the averages found in the previous step as threshold points to determine the match outcomes. For example, suppose the average of all the 10 stats in games won is 5. You look into the individual games-won info. Won-Game1's stat value is 6 for all 10 stats. Since there are 5 positive correlated stats and 5 negative correlated stats, you can say that Won-Game1 has gone past 5 of the threshold points (For negative correlated stats, you need to go below the team average to say you broke the threshold point). Calculate the number of

threshold points broken by each game won, and calculate the average of this. Do this for losses as well.

7) Determine if the games were won, lost or drawn by using data from step 6 and applying it to step 4. In step 4 you calculated the simulated results of stats in 38 games with the new squad. In step 6, you calculated the threshold points for Tottenham to break in order to win (or lose). If none of the win and lose threshold points were broken, assume that match was drawn.
8) Calculate the total league points after 38 games. 3 points are earned when winning, 1 point is earned when drawing and none are earned when losing.
9) Simulate the above steps 1000 times. Calculate the average total league points Tottenham earns with the new squad.

a. Use the simulation steps above with the new squad obtained in question 3-a. Chelsea, the team one standing above Tottenham, had a total of 74 league points during the 2021-22 season. With the new squad, could Tottenham beat Chelsea the next season?

III. Solutions

1.a. In 10% confidence level, MP and Starts come out to be invalid.

	A	B	C	D	E	F	G	H	I	J	K
24		*Coefficients*	*Std Err*	*LCL*	*UCL*	*t Stat*	*p-value*	*H0 (10%)*	*VIF*	*TOL*	*Beta*
25	**Intercept**	-15,200.9852	3,108.1507	-20,334.6696	-10,067.3007	-4.8907	1.9178E-6	*Rejected*			
26	**Age**	-116.5613	29.4442	-165.1938	-67.9288	-3.9587	0.0001	*Rejected*	1.0757	0.9296	-0.1850
27	**MP**	-20.6263	29.1078	-68.7031	27.4506	-0.7086	0.4793	*Accepted*	6.3738	0.1569	-0.0806
28	**Starts**	16.3565	25.4065	-25.6071	58.3200	0.6438	0.5204	*Accepted*	6.6792	0.1497	0.0750
29	**Gls**	196.1289	32.0745	143.1520	249.1058	6.1148	4.2563E-9	*Rejected*	1.8314	0.5460	0.3728
30	**Ast**	163.0644	51.0059	78.8187	247.3101	3.1970	0.0016	*Rejected*	1.8573	0.5384	0.1963
31	**Cmp%**	74.4483	14.3862	50.6868	98.2098	5.1750	5.0602E-7	*Rejected*	1.2814	0.7804	0.2639
32	**Rating**	2,093.2385	482.3264	1,296.5874	2,889.8895	4.3399	2.1581E-5	*Rejected*	2.4983	0.4003	0.3090
33	*T (10%)*	1.6517									
34	*LCL - Lower limit of the 90% confidence interval*										
35	*UCL - Upper limit of the 90% confidence interval*										

1.b.

	A	B	C	D	E	F	G	H	I	J	K
7	**Regression Statistics**										
8	*R*		0.7378	***R-Squared***		**0.5444**	*Adjusted R-Squared*		0.5343		
9	*MSE*		2,049,237.9242	***S***		**1,431.5160**	*MAPE*		55.8474		
10	*Durbin-Watson (DW)*		2.0444	*Log likelihood*		-2,011.9798					
11	*Akaike inf. criterion (AIC)*		17.3964	*AICc*		17.3975					
12	*Schwarz criterion (BIC)*		17.4855	*Hannan-Quinn criterion (HQC)*		17.4323					
13	*PRESS*		491,121,270.2896	*PRESS RMSE*		1,454.9577	*Predicted R-Squared*		0.5168		
24		*Coefficients*	*Std Err*	*LCL*	*UCL*	*t Stat*	*p-value*	*H0 (10%)*	*VIF*	*TOL*	*Beta*
25	**Intercept**	-16,375.8915	2,541.5016	-20,573.4961	-12,178.2868	-6.4434	########	*Rejected*			
26	**Age**	-115.4899	29.2254	-163.7593	-67.2205	-3.9517	0.0001	*Rejected*	1.0668	0.9374	-0.1833
27	**Gls**	190.2868	30.6910	139.5968	240.9768	6.2001	2.6528E-9	*Rejected*	1.6880	0.5924	0.3617
28	**Ast**	154.4567	49.3339	72.9756	235.9377	3.1308	0.0020	*Rejected*	1.7492	0.5717	0.1859
29	**Cmp%**	75.1166	14.2676	51.5519	98.6813	5.2648	3.2607E-7	*Rejected*	1.2688	0.7882	0.2663
30	**Rating**	2,232.7088	418.4783	1,541.5400	2,923.8776	5.3353	2.3134E-7	*Rejected*	1.8932	0.5282	0.3296
31	*T (10%)*	1.6516									
32	*LCL - Lower limit of the 90% confidence interval*										
33	*UCL - Upper limit of the 90% confidence interval*										

The model explains about 55% the variance of the independent variable, that is quite low. The reason for this may be the nature of player's valuation where non-performance features such as fame or popularity comes into account as much as the performance factors considered in the model.

1.c.-d. Regression model should be *"Valuation in 10 thousand pounds = - 16,375.8915 - 115.4899 * Age + 190.2868 * Gls + 154.4567 * Ast + 75.1166 * Cmp% + 2,232.7088 * Rating"*.

You can feed excel function to calculate PV, predicted value of each players as presented in column P. In turn you can feed another excel function to divide 'Valuation in 10 thousand pounds' column by 'PV' column. Sorting can be done by giving a filter to the headers of the columns.

2.a. Create a table with the same exact labels but excluding the players' names and their individual information. Create a new label for match number and fill the rest of the table with this formula: =SUMIF($B$5:$B$535,$F540,H$5:H$535). This formula sums the array (H$5:H$535) that contains the stats that you have to sum up, only if the match number ($B$5:$B$535) is equal to the match number we are looking for ($F540). Remember that we have to account for data with %. Therefore for datas with % we add a /COUNTIF($B$5:$B$535,F540). This divides the total % by number of player ran in each match so that we calculate the mean % of the desired stats. The result will look like the following:

Summary												
		Performance						Expected			SCA	
Match number	Min	Gls	Ast	PK	PKatt	Sh	SoT	xG	npxG	xA	SCA	GCA
1	990	1	1	0	0	14	2	1.2	1.2	0.9	21	2
2	990	1	0	1	1	7	5	1.6	0.8	0.5	12	2
3	990	1	0	0	0	15	8	0.9	0.9	0.6	20	2
4	957	0	0	0	0	2	1	0.1	0.1	0	4	0
5	990	0	0	0	0	9	2	0.6	0.6	0.3	13	0
6	990	1	1	0	0	10	4	1.1	1.1	0.9	15	2
7	990	1	1	0	0	17	7	1.6	1.6	1.5	33	2
8	990	3	3	0	0	14	4	1.7	1.7	1.5	23	6
9	990	0	0	0	0	7	4	0.6	0.6	0.7	9	0
10	990	0	0	0	0	9	0	0.7	0.7	0.6	16	0
11	990	0	0	0	0	8	0	0.7	0.7	0.6	15	0
12	990	2	1	0	0	13	4	2.2	2.2	1.2	19	3
13	990	0	0	0	0	12	3	1	1	0.9	23	0
14	990	1	1	0	0	11	6	2.3	2.3	2	19	2
15	990	3	2	0	0	17	7	1.8	1.8	1.4	27	6
16	990	2	2	0	0	17	7	2.3	2.3	1	26	3
17	990	3	3	0	0	27	8	3.7	3.7	2.3	43	4
18	990	2	1	0	0	11	5	3.3	3.3	2.2	20	4
19	990	3	3	0	0	17	6	2.6	2.6	2.3	31	6
20	990	1	0	1	1	20	9	2.4	1.6	1.4	33	2
21	990	1	1	0	0	21	9	1.8	1.8	1.3	33	2
22	990	3	1	1	1	15	5	3.2	2.4	1.7	27	3
23	990	0	0	0	0	6	3	0.4	0.4	0.4	10	0
24	990	1	1	0	0	8	3	1.4	1.4	1.4	13	2
25	990	0	0	0	0	17	7	1.3	1.3	1	23	0
26	990	3	3	0	0	6	5	1.9	1.9	1.8	12	6
27	990	4	4	0	0	15	10	2.3	2.3	2.2	26	7
28	990	4	4	0	0	14	7	3	3	2.8	25	8
29	990	1	0	1	1	9	1	1.4	0.6	0.6	18	2
30	990	2	2	0	0	17	4	2.3	2.3	2.3	27	4
31	990	5	4	0	0	19	6	3.1	3.1	2.6	37	10
32	990	4	3	0	0	11	5	1	1	0.8	14	7
33	990	0	0	0	0	5	0	0.4	0.4	0.4	8	0
34	990	0	0	0	0	10	1	0.8	0.8	0.4	17	0
35	990	3	3	0	0	13	7	1.4	1.4	0.8	22	6
36	990	1	1	0	0	9	3	1	1	1.1	17	2
37	990	1	0	1	1	20	6	2.1	1.4	1.2	36	2
38	990	5	4	0	0	19	12	3.6	3.6	3.1	30	7
									Formula= =SUMIF($B$5:$B$535,$F540,P$5:P$535)			

	Pass															
			Total				Short			Medium			Long			
Match number	Cmp	Att	Cmp%	TotDist	PrgDist	Cmp	Att	Cmp%	Cmp	Att	Cmp%	Cmp	Att	Cmp%	Ast	xA
1	240	316	74.31428571	5030	2070	99	112	82.87857143	92	111	69.75714286	41	79	47.32857143	1	
2	289	376	70.65	5283	2090	136	148	86.15	122	151	73.60714286	28	63	28.49285714	0	
3	438	516	75.79230769	8156	2200	202	218	84.88461538	187	205	81.23846154	46	83	47.19230769	0	
4	245	320	76.23076923	4820	1766	106	121	75.26923077	99	114	87.29230769	36	71	47.22307692	0	
5	406	494	81.92857143	7388	2121	179	206	88.57857143	155	170	89.94285714	55	88	64.16428571	0	
6	430	508	83.45	8394	2454	186	199	91.65	171	188	89.64285714	65	105	63.12857143	1	
7	379	474	80.58461538	6873	2495	174	194	92.50769231	158	190	73.61538462	39	69	49.11538462	1	
8	542	641	83.63636364	9345	3076	266	292	92.13636364	220	252	82.17272727	49	82	64.54545455	3	
9	561	662	75.15714286	10811	2759	208	225	86.17142857	293	321	74.66428571	58	104	56.07142857	0	
10	488	597	80.83571429	8505	2355	228	249	91.85	191	222	85.35	56	88	65.22857143	0	
11	486	579	77.59285714	9295	2892	189	208	85.92142857	228	258	78.50714286	59	86	58.82857143	0	
12	329	427	70.39285714	6000	2367	138	160	80.72142857	151	180	73.05	37	71	48.91428571	1	
13	581	693	80.77857143	11549	2775	216	250	86.88571429	272	298	87.39285714	86	117	58.10714286	0	
14	377	497	73.27142857	7022	2234	162	187	76.13571429	166	201	76.35	43	82	40.81428571	1	
15	404	482	81.43571429	7491	2356	197	218	90.2	135	161	81.22142857	65	87	60.17857143	2	
16	505	579	80.10714286	8991	2491	243	261	86.34285714	185	212	77.15	64	83	60.37857143	2	
17	533	618	85.53571429	10492	2794	188	213	87.69285714	258	288	83.92142857	77	99	63.35	3	
18	338	434	74.73571429	6792	2734	140	163	87.92142857	135	165	79.51428571	57	87	52.01428571	1	
19	571	637	86.45714286	11114	3080	223	245	89.88571429	270	290	83.61428571	70	90	70.70714286	3	
20	492	592	79.45	9319	3233	211	233	88.25	218	246	84.61428571	58	95	57.22142857	0	
21	668	772	86.45	12241	3682	300	329	90.17142857	287	317	90.32142857	70	98	65.55714286	1	
22	514	576	87.00714286	10055	2329	196	210	86.53571429	242	255	92.32857143	73	101	61.13571429	1	
23	267	351	70.05	5249	2010	112	130	81.41428571	106	132	67.49285714	44	74	54.48571429	0	
24	371	463	79.09285714	7225	2502	152	172	82.07857143	161	185	87.27857143	53	92	47.62142857	1	
25	499	611	79.87142857	9417	2760	205	237	77.45714286	220	258	86.69285714	65	90	49.86428571	0	
26	258	322	70.75714286	5055	1987	107	118	77.49285714	105	119	75.32142857	40	71	45.39285714	3	
27	427	488	81.83571429	8368	2930	161	174	85.74285714	195	210	79.55	61	85	50.2	4	
28	591	670	86.77857143	10777	2916	247	266	92.74285714	286	310	88.23571429	54	76	51.74285714	4	
29	523	610	82.18571429	9687	2467	217	236	81.74285714	246	266	85.32857143	53	83	52.20714286	0	
30	555	628	86.74285714	10199	2737	239	251	87.97857143	247	267	92.35	66	97	52.82142857	2	
31	554	616	89.39285714	10488	2629	196	206	95.56428571	297	318	91.73571429	53	80	57.26428571	4	
32	409	473	81.45	7703	2199	184	197	93.05	164	183	68.17857143	54	72	71.65714286	3	
33	403	493	77.50714286	7941	2023	134	152	84.48571429	208	226	76.79285714	53	91	58.24285714	0	
34	452	540	74.31538462	8128	2295	193	212	84.14615385	218	245	79.08461538	34	63	49.50769231	0	
35	350	440	80.15714286	7514	2548	116	140	82.16428571	164	187	87.81428571	68	96	60.33571429	3	
36	285	385	77.12142857	5959	2058	121	135	83.51428571	101	120	69.95	60	107	48.59285714	1	
37	598	701	80.46923077	12239	3298	210	237	87.18461538	284	314	74.46153846	97	128	60.33076923	0	
38	556	626	85.82142857	10874	2981	212	225	93.62857143	261	286	88.52142857	76	103	57.93571429	4	
			Formula= =SUMIF($B$5:$B$535,$F540,U$5:U$535)													

2.b. Using the new data provided , create a new label indicating whether each match was a win or not a win. Using a filter, arrange the games by win/ not wins. Calculate the area of wins by using =AVERAGE(E539:E560) where (E539:E560) is the area of the wins. Do the same for not wins. Then use =(E578-E579)/(MAX(E578:E579)) for the % differences. (E578-E579)

			Performance						Expected			SCA	
result		Min	Gls	Ast	PK	PKatt	Sh	SoT	xG	npxG	xA	SCA	GCA
W	1	990	1	1	0	0	14	2	1.2	1.2	0.9	21	2
W	2	990	1	0	1	1	7	5	1.6	0.8	0.5	12	2
W	3	990	1	0	0	0	15	8	0.9	0.9	0.6	20	2
W	7	990	1	1	0	0	17	7	1.6	1.6	1.5	33	2
W	8	990	3	3	0	0	14	4	1.7	1.7	1.5	23	6
W	12	990	2	1	0	0	13	4	2.2	2.2	1.2	19	3
W	14	990	1	1	0	0	11	6	2.3	2.3	2	19	2
W	15	990	3	2	0	0	17	7	1.8	1.8	1.4	27	6
W	16	990	2	2	0	0	17	7	2.3	2.3	1	26	3
W	17	990	3	3	0	0	27	8	3.7	3.7	2.3	43	4
W	19	990	3	3	0	0	17	6	2.6	2.6	2.3	31	6
W	21	990	1	1	0	0	21	9	1.8	1.8	1.3	33	2
W	22	990	3	1	1	1	15	5	3.2	2.4	1.7	27	3
W	26	990	3	3	0	0	6	5	1.9	1.9	1.8	12	6
W	27	990	4	4	0	0	15	10	2.3	2.3	2.2	26	7
W	28	990	4	4	0	0	14	7	3	3	2.8	25	8
W	30	990	2	2	0	0	17	4	2.3	2.3	2.3	27	4
W	31	990	5	4	0	0	19	6	3.1	3.1	2.6	37	10
W	32	990	4	3	0	0	11	5	1	1	0.8	14	7
W	35	990	3	3	0	0	13	7	1.4	1.4	0.8	22	6
W	37	990	1	0	1	1	20	6	2.1	1.4	1.2	36	2
W	38	990	5	4	0	0	19	12	3.6	3.6	3.1	30	7
L	4	957	0	0	0	0	2	1	0.1	0.1	0	4	0
L	5	990	0	0	0	0	9	2	0.6	0.6	0.3	13	0
L	6	990	1	1	0	0	10	4	1.1	1.1	0.9	15	2
L	9	990	0	0	0	0	7	4	0.6	0.6	0.7	9	0
win average		990	2.545455	2.090909	0.136364	0.136364	15.40909	6.363636	2.163636	2.059091	1.627273	25.59091	4.545455
not win average		987.9375	0.4375	0.25	0.125	0.125	9.5	2.875	1.075	0.975	0.80625	15.875	0.875
% difference		0.002083	0.828125	0.880435	0.083333	0.083333	0.383481	0.548214	0.503151	0.52649	0.504539	0.379663	0.8075
		Formula for E 578 =AVERAGE(E539:E560)				Formula for E579~ =AVERAGE(E561:E576)				Formula for E580~ =(E578-E579)/(MAX(E578:E579))			

				Total				Short			Medium			Long	
result		Cmp	Att	Cmp%	TotDist	PrgDist	Cmp	Att	Cmp%	Cmp	Att	Cmp%	Cmp	Att	Cmp%
W	1	240	316	74.31429	5030	2070	99	112	82.87857	92	111	69.75714	41	79	47.32857
W	2	289	376	70.65	5283	2090	136	148	86.15	122	151	73.60714	28	63	28.49286
W	3	438	516	75.79231	8156	2200	202	218	84.88462	187	205	81.23846	46	83	47.19231
W	7	379	474	80.58462	6873	2495	174	194	92.50769	158	190	73.61538	39	69	49.11538
W	8	542	641	83.63636	9345	3076	266	292	92.13636	220	252	82.17273	49	82	64.54545
W	12	329	427	70.39286	6000	2367	138	160	80.72143	151	180	73.05	37	71	48.91429
W	14	377	497	73.27143	7022	2234	162	187	76.13571	166	201	76.35	43	82	40.81429
W	15	404	482	81.43571	7491	2356	197	218	90.2	135	161	81.22143	65	87	60.17857
W	16	505	579	80.10714	8991	2491	243	261	86.34286	185	212	77.15	64	83	60.37857
W	17	533	618	85.53571	10492	2794	188	213	87.69286	258	288	83.92143	77	99	63.35
W	19	571	637	86.45714	11114	3080	223	245	89.88571	270	290	83.61429	70	90	70.70714
W	21	668	772	86.45	12241	3682	300	329	90.17143	287	317	90.32143	70	98	65.55714
W	22	514	576	87.00714	10055	2329	196	210	86.53571	242	255	92.32857	73	101	61.13571
W	26	258	322	70.75714	5055	1987	107	118	77.49286	105	119	75.32143	40	71	45.39286
W	27	427	488	81.83571	8368	2930	161	174	85.74286	195	210	79.55	61	85	50.2
W	28	591	670	86.77857	10777	2916	247	266	92.74286	286	310	88.23571	54	76	51.74286
W	30	555	628	86.74286	10199	2737	239	251	87.97857	247	267	92.35	66	97	52.82143
W	31	554	616	89.39286	10488	2629	196	206	95.56429	297	318	91.73571	53	80	57.26429
W	32	409	473	81.45	7703	2199	184	197	93.05	164	183	68.17857	54	72	71.65714
W	35	350	440	80.15714	7514	2548	116	140	82.16429	164	187	87.81429	68	96	60.33571
W	37	598	701	80.46923	12239	3298	210	237	87.18462	284	314	74.46154	97	128	60.33077
W	38	556	626	85.82143	10874	2981	212	225	93.62857	261	286	88.52143	76	103	57.93571
L	4	245	320	76.23077	4820	1766	106	121	75.26923	99	114	87.29231	36	71	47.22308
L	5	406	494	81.92857	7388	2121	179	206	88.57857	155	170	89.94286	55	88	64.16429
L	6	430	508	83.45	8394	2454	186	199	91.65	171	188	89.64286	65	105	63.12857
L	9	561	662	75.15714	10811	2759	208	225	86.17143	293	321	74.66429	58	104	56.07143
win average		458.5	539.7727	80.86544	8695.909	2613.136	190.7273	209.1364	87.35418	203.4545	227.5909	81.11439	57.77273	86.13636	55.24505
not win average		426.6875	520.75	78.14396	8154.938	2450.25	174.8125	195.5	84.83355	188.875	213.375	81.84633	55.75	90.0625	55.15683
% difference		0.069384	0.035242	0.033654	0.06221	0.062334	0.083443	0.065203	0.028855	0.07166	0.062463	-0.00894	0.035012	-0.04359	0.001597

represents the win average- loss average and (MAX(E578:E579) will get the bigger number from win or loss. The data will look as such:

2.c. Transpose the data above and ascend the % differences to gain insight in what stats are most important in winning a game vs what stats are detrimental to the team.

		% differe
Corner Kicks	Str	-0.81818
Blocks	ShSv	-0.75758
Outcomes	Int	-0.55669
Outcomes	Blocks	-0.47419
Dribbles	Succ%	-0.42553
Body Parts	Head	-0.34065
Pass Types	Dead	-0.28252
Performance	PKcon	-0.27273
Body Parts	TI	-0.26621
Corner Kicks	Out	-0.20455
Outcomes	Off	-0.18881
Pass Types	FK	-0.17552
Blocks	Sh	-0.17125
Carries	Mis	-0.15636
Carries	Dis	-0.1342
Long	CrsPA	-0.12727
Pressures	Att 3rd	-0.12377

Performance	PKwon	0.3125
Long	KP	0.34728
Carries	CPA	0.36458
SCA	SCA	0.37966
Performanc	Sh	0.38348
Dribbles	Megs	0.4375
Expected	xG	0.50315
Expected	xA	0.50454
Long	xA	0.50454
Expected	npxG	0.52649
Blocks	Err	0.54167
Performanc	SoT	0.54821
Pass Types	TB	0.70745
SCA	GCA	0.8075
Long	Ast	0.88043
Performanc	Gls	0.91813
Performanc	Ast	0.93487

Here on the left we see the top list of qualities that the more often seen, the lower the chances of the team winning (will be defined as detrimental) vs on the right top list of qualities that have a positive correlation with winning (will be defined as helpful). On the tail end of detrimental attributes are in order, # of straight corner kicks (str), # of times a shot was blocked by a target standing in its path (shsv) , # of passes intercepted(int) ,# of passes Blocked by the opponent who was standing it the path (blocks), and % of dribbles succeeded (succ%). An interesting insight is that the # of straight corner kicks seems to be extremely detrimental to a team's victory. Although used as an offensive tool, corner kicks that are kicked in a straight motion might have to do with the FW's confidence or level of play. Another interesting point is shsv. Although if a Tottenham defender stands in the path of the opponent's shot might be a good thing in isolation, looking at the bigger picture it might not be a reliable defense strategy. More rigid defense tactic that pressures the opponent's offense might be a more 'skillful' type of defense that doesn't rely too heavily on the luck factor that comes from simply blocking by standing in its path.

Most helpful 5 stats are number of two offensive actions directly leading to a goal such as passes/dribbles/drawing fouls (Goal creating actions: GCA), # of Long passes that led to an assist (AST), # of Completed pass sent between back defenders into open space (TB), # of shots on target (SoT), actions leading to an opponent's error (Err). Since goals and assists are results of a winning game and these statistics don't provide insight to the game, we excluded them from potential factors. Another interesting insight is # of completed pass sent between back to defenders. Giving a pass to the backspace demonstrates a team's ability to use the field widely and proficiently penetrate through an empty space. This is why TB stats might be so important in winning a game.

3.a.

The constraints and objective function are developed below. We let go of Son Heung-Min and Bentancur, and scout Jarrod Bowen and Philipppe Coutinho.

OBJ				Min	35900	=SUMPRODUCT(B26:B163,E26:E163)
Constraints						
Squad	FW Max	3	=	3		=B26+B27+B33+SUM(B75:B115)
	MF Max	3	=	3		=SUM(B116:B163)+B28+B29+B30
	DF Max	4	=	4		=SUM(B36:B74)+B31+B32+B34+B35
Stat	SoT	159	>=	152.25	145	=SUMPRODUCT($B$26:$B$163,L26:L163)*1.05
	Ast	41	>=	36.75	35	=SUMPRODUCT($B$26:$B$163,M26:M163)*1.05
	TB	42	>=	37.8	36	=SUMPRODUCT($B$26:$B$163,N26:N163)*1.05
	GCA	91	>=	85.05	81	=SUMPRODUCT($B$26:$B$163,O26:O163)*1.05
	Err	0	>=	0	0	=SUMPRODUCT($B$26:$B$163,P26:P163)*1.05
	Succ%	504.9	<=	546		=SUMPRODUCT($B$26:$B$163,Q26:Q163)
	Str	4	<=	4		=SUMPRODUCT($B$26:$B$163,R26:R163)
	Int	325	<=	331		=SUMPRODUCT($B$26:$B$163,S26:S163)
	Blocks	297	<=	300		=SUMPRODUCT($B$26:$B$163,T26:T163)
	ShSv	4	<=	4		=SUMPRODUCT($B$26:$B$163,U26:U163)
	MAX 2	2	<=	2		=SUM(B36:B163)

3.b.

The constraints and objective function are developed below. We let go of Emerson and Bergwijn, and scout Rob Holding and Michail Antonio.

OBJ				Min	33000	=SUMPRODUCT(B26:B163,E26:E163)
Constraints						
Squad	FW Max	3	=	3	C5	=B26+B27+B33+SUM(B75:B115)
	MF Max	3	=	3	C6	=SUM(B116:B163)+B28+B29+B30
	DF Max	4	=	4	C7	=SUM(B36:B74)+B31+B32+B34+B35
Stat	SoT	145	>=	145	C8	=SUMPRODUCT($B$26:$B$163,L26:L163)
	Ast	37	>=	35	C9	=SUMPRODUCT($B$26:$B$163,M26:M163)
	TB	39	>=	36	C10	=SUMPRODUCT($B$26:$B$163,N26:N163)
	GCA	86	>=	81	C11	=SUMPRODUCT($B$26:$B$163,O26:O163)
	Err	0	>=	0	C12	=SUMPRODUCT($B$26:$B$163,P26:P163)
	Succ%	519.7	<=	546	C13	=SUMPRODUCT($B$26:$B$163,Q26:Q163)
	Str	4	<=	4	C14	=SUMPRODUCT($B$26:$B$163,R26:R163)
	Int	329	<=	331	C15	=SUMPRODUCT($B$26:$B$163,S26:S163)
	Blocks	296	<=	300	C16	=SUMPRODUCT($B$26:$B$163,T26:T163)
	ShSv	4	<=	4	C17	=SUMPRODUCT($B$26:$B$163,U26:U163)
	MAX 2	2	<=	2	C18	=SUM(B36:B163)

3.c.

The constraints and objective function are developed below. We let go of Pierre Højbjerg and Emerson, and scout Nathan Aké and Mohamed Elneny.

OBJ	Max	-635.1					B2	=SUMPRODUCT($B$26:$B$163,L26:L163) +
								SUMPRODUCT($B$26:$B$163,M26:M163) +
Constraints								SUMPRODUCT($B$26:$B$163,N26:N163) +
	FW Max	3	=	3				SUMPRODUCT($B$26:$B$163,O26:O163) -
Squad	MF Max	3	=	3				SUMPRODUCT($B$26:$B$163,Q26:Q163) -
	DF Max	4	=	4				SUMPRODUCT($B$26:$B$163,R26:R163) -
								SUMPRODUCT($B$26:$B$163,S26:S163) -
								SUMPRODUCT($B$26:$B$163,T26:T163) -
							C5	=B26+B27+B33+SUM(B75:B115)
							C6	=SUM(B116:B163)+B28+B29+B30
							C7	=SUM(B36:B74)+B31+B32+B34+B35

4.a. The solution for each step is below.

1)

Player	Pos	Ast	SoT	GCA	TB	Str	Int	Blocks	ShSv	Err	Succ%
Son HeungMin	FW	0	1	0	0	0	0	0	0	0	100
Son HeungMin	FW	0	1	0	0	0	1	1	0	0	100
Son HeungMin	FW	0	2	0	0	0	2	2	0	0	75
Son HeungMin	FW	0	1	0	0	0	1	0	0	0	66.7
Son HeungMin	FW	0	2	0	0	0	0	1	0	0	0
Son HeungMin	FW	1	1	1	0	0	2	1	0	0	60
Son HeungMin	FW	0	1	0	0	0	1	0	0	0	0
Son HeungMin	FW	0	1	0	0	0	1	1	0	0	100
Son HeungMin	FW	0	0	0	0	0	0	4	0	0	50
Son HeungMin	FW	0	0	0	0	0	2	2	0	0	100
Son HeungMin	FW	0	0	0	0	0	2	1	0	0	62.5
Son HeungMin	FW	0	2	0	0	0	0	1	0	0	0
Son HeungMin	FW	1	1	2	0	0	2	0	0	0	50
Son HeungMin	FW	0	1	0	0	0	1	2	0	0	0
Son HeungMin	FW	0	2	0	0	0	1	0	0	0	100
Son HeungMin	FW	0	2	1	1	0	1	5	0	0	100
Son HeungMin	FW	1	1	2	0	0	2	4	0	0	71.4
Son HeungMin	FW	0	1	0	0	0	0	0	0	0	40
Son HeungMin	FW	0	2	0	0	3	3	2	0	0	0

2)

Player Pool

Row Labels	Average of Ast	Average of SoT	Average of GCA	Average of TB	Average of Str	Average of Int	Average of Blocks	Average of ShSv	Average of Err
Ben Davies	0.034482759	0.24137931	0.137931034	0	0	1.379310345	1.172413793	0.068965517	0
Dejan Kulusevski	0.375	0.5625	0.75	0	0	0.8125	1.375	0	0
Emerson	0.032258065	0.129032258	0.129032258	0	0	1.193548387	1.838709677	0	0
Eric Dier	0	0.114285714	0.114285714	0.114285714	0	1.028571429	0.457142857	0.028571429	0
Harry Kane	0.243243243	1.378378378	0.567567568	0.513513514	0	1.72972973	1.756756757	0.027027027	0
Jarrod Bowen	0.277777778	0.916666667	0.583333333	0.138888889	0.111111111	1.055555556	1.305555556	0	0
Philipe Coutinho	0.157894737	0.842105263	0.315789474	0.157894737	0	1.210526316	0.894736842	0	0
Pierre Højbjerg	0.055555556	0.277777778	0.138888889	0.194444444	0	1.611111111	0.5	0	0
Ryan Sessegnon	0.133333333	0.066666667	0.266666667	0	0	0.8	1.333333333	0	0
Steven Bergwijn	0.04	0.4	0.12	0.04	0	0.4	0.24	0	0
Grand Total	**0.125448029**	**0.519713262**	**0.301075269**	**0.139784946**	**0.014336918**	**1.186379928**	**1.082437276**	**0.014336918**	**0**

Row Labels	Average of Succ%2	StdDev of Ast2	StdDev of SoT2	StdDev of GCA2	StdDev of TB2	StdDev of Str2	StdDev of Int2	StdDev of Blocks2	StdDev of ShSv2	St
Ben Davies	14.36896552	0.185695338	0.43549417	0.350931203	0	0	1.497946049	1.167077104	0.257880715	
Dejan Kulusevski	38.53125	0.619139187	0.81394103	1	0	0	1.376892637	1.147460965	0	
Emerson	30	0.179605302	0.340777101	0.427546137	0	0	1.249516035	1.485412944	0	
Eric Dier	0	0	0.322802851	0.322802851	0.322802851	0	1.014185106	0.657215926	0.169030851	
Harry Kane	57.92972973	0.547996624	1.186763839	0.800712896	1.145562007	0	1.465314692	1.382538033	0.164398987	
Jarrod Bowen	32.08055556	0.614636297	0.937321412	0.967323258	0.424451093	0.398409536	1.040451672	1.237957868	0	
Philipe Coutinho	49.12631579	0.501459857	0.688247202	0.671038298	0.374634325	0	1.315672509	0.87526103	0	
Pierre Højbjerg	55.09166667	0.232310684	0.51331478	0.424451093	0.524782646	0	1.049565292	0.878310066	0	
Ryan Sessegnon	13.88666667	0.351865775	0.25819889	0.457737708	0	0	0.941123948	1.2344268	0	
Steven Bergwijn	15.6	0.2	0.645497224	0.331662479	0.2	0	0.645497224	0.435889894	0	
Grand Total	**31.45698925**	**0.400581048**	**0.817175292**	**0.647934398**	**0.528158151**	**0.146206865**	**1.229593493**	**1.216055575**	**0.119089053**	

3)

Aggregate AVG									
1.34954547	4.928792035	3.123494937	1.159027298	0.111111111	11.22085287	10.87364882	0.124563973	0	31.45698925
=SUM(B13:B22)	=SUM(C13:C22)								

Aggregate SD							
0.419042848	0.807118538	0.625891873	1.15707464	0.142857143	0.647082032	0.984134518	0.087136102
=DEVSQ(L13:L22)	=DEVSQ(M13:M22)						

4)

	Need to eliminate negative values, assume negative values corresponds to 0									
Game Num	Ast	SoT	GCA	TB	Str	Int	Blocks	ShSv	Err	Succ%
1	1.521052566	5.128683705	3.280312251	1.87011788	0.063816452	12.01179049	10.5744338	0.17374372	0	48.00297602
2	1.370606791	4.165819666	2.692874713	1.640744553	0.09940537	12.05460531	10.65828765	0.138624825	0	-17.02178552
3	1.256006522	4.600127552	1.801032474	0.973855192	0.063798118	11.07411732	11.43240539	0.108372039	0	22.31747832
4	1.100563635	4.567574956	2.828749856	-0.192881022	-0.052186746	10.93189199	10.58119827	0.137312453	0	20.79226859
5	1.508517862	5.606547495	3.482734437	-0.302408834	0.221580637	12.08187723	10.51099181	0.104957849	0	96.02034965
6	1.297859446	4.784182984	2.814058906	-0.686877822	0.161369908	11.43792739	10.32460396	0.050624597	0	-35.3673063
7	1.546077433	4.335679667	3.5850388	-0.780183075	0.146956743	12.59377793	10.19025753	0.051932841	0	61.66357897
8	1.095786527	4.055950875	3.523381121	-1.367267072	-0.015437678	11.24468122	12.13576272	0.077852608	0	50.68366489
9	0.76173795	4.584196017	3.299137514	-0.639094751	0.413221764	10.42754567	11.00206589	-0.068219627	0	36.70511426
10	1.509806254	5.823433722	3.619692628	3.229935196	0.105228496	11.64664041	11.61022525	0.117716399	0	4.420591332
11	1.097290891	3.843085138	3.271969376	0.57153252	0.197142803	10.49179517	11.04151362	0.057764114	0	-41.71076995
12	1.324225935	5.401775932	2.926909423	1.459828581	0.367830019	11.52115767	9.26048608	0.160791587	0	46.46892677
13	1.170944091	4.518139891	2.808216574	-0.227829067	0.285904496	11.01614819	9.96410756	0.282410076	0	62.69096056
14	1.381823333	5.27658812	2.934127052	0.617953656	0.073495444	9.599520636	9.954004922	0.175046297	0	74.3775302
15	1.694890111	5.803141437	3.913668589	2.284622354	0.151382902	11.38791819	11.3985357	0.115925638	0	-24.27420322
16	1.886783622	5.84314353	3.45690448	0.53476462	-0.053086729	11.71779422	11.78668286	0.124983432	0	5.109701798
17	1.319039535	4.71655607	2.879283398	-0.643496417	0.339280424	11.15180696	10.60554073	0.202828135	0	56.48574629
18	1.035453441	5.075045559	2.905980522	0.546403693	0.238290098	11.09395892	11.43033633	0.027038415	0	61.21592604
19	1.029305336	2.998385133	3.88099165	0.111791598	0.262917118	10.79415021	8.825872875	0.161981169	0	70.49641117
20	1.81110564	4.76696427	2.789868633	0.037508897	0.199572876	11.54744673	10.66511246	0.098686966	0	5.877174296
21	1.088461957	4.860334099	3.433315623	-1.057888272	0.157183281	11.59563767	12.24940107	0.065185419	0	56.34105951
22	1.569014838	5.353890974	2.477872492	1.93577945	0.193616602	10.83747162	9.444938555	0.139833456	0	59.88499185
23	0.808171925	4.544704634	3.263901688	2.127113119	0.123320873	10.67092065	10.66034073	0.14141914	0	25.06070375
24	1.431470778	4.973410412	2.428057843	1.989027137	0.081255101	11.27512191	12.57766673	0.191382887	0	-16.26257262
25	1.781355133	3.962665781	2.929570145	1.692874703	0.209939048	11.10829834	10.51517876	0.317531064	0	33.00204602
26	0.74815425	5.391519254	1.313109577	1.988800675	0.017694619	11.07611906	10.20285634	0.286521128	0	-33.85040419
27	1.094581224	5.17065716	3.53749204	-0.178573104	0.116427404	11.12878849	9.713485433	0.270489672	0	37.45027797
28	0.769142981	5.96039611	1.854174384	1.274066197	0.028491595	10.78012337	11.23616163	0.039112577	0	18.45389656
29	1.205817547	5.81115932	2.918267301	0.672108071	0.274197518	10.30975982	11.79593918	0.127890755	0	72.35235645
30	1.656703789	5.738241557	3.453667257	0.377204107	0.086343535	11.23863419	11.50322394	0.228568286	0	86.72172635
31	1.103016401	5.678440143	2.98777493	2.49786976	0.140649535	12.22004445	10.89718225	0.174144551	0	37.43411286
32	0.753541484	6.008401181	2.4553674	2.28237505	0.026510866	11.53736416	12.18161622	0.078873888	0	-2.825681189
33	1.504699847	4.752596533	3.145051428	1.839628094	-0.027825891	11.52874344	11.56909519	0.107379416	0	74.86450297
34	0.792894978	4.420032909	3.931026349	0.861294954	0.125679277	11.35487463	11.13835699	0.121694975	0	36.03028026
35	1.679370669	5.569554407	3.996182732	1.02726565	0.183812062	10.89919599	9.025885876	0.075211032	0	45.47059654
36	1.474704942	3.288821601	2.836537927	1.716765845	0.316819296	10.99765364	11.07731823	0.226403389	0	39.60346761
37	1.432757328	4.390399891	1.770886181	2.980880833	0.420182297	12.37291326	9.19988656	0.329389271	0	39.88614242
38	0.678111017	3.598812675	2.973485401	1.172513269	0.020712271	11.53640407	10.81707562	0.192761111	0	65.14679629

	No Negative Values									
Game Num	Ast	SoT	GCA	TB	Str	Int	Blocks	ShSv	Err	Succ%
1	1.521052566	5.128683705	3.280312251	1.87011788	0.063816452	12.01179049	10.5744338	0.17374372	0	48.00297602
2	1.370606791	4.165819666	2.692874713	1.640744553	0.09940537	12.05460531	10.65828765	0.138624825	0	0
3	1.256006522	4.600127552	1.801032474	0.973855192	0.063798118	11.07411732	11.43240539	0.108372039	0	22.31747832
4	1.100563635	4.567574956	2.828749856	0	0	10.93189199	10.58119827	0.137312453	0	20.79226859
5	1.508517862	5.606547495	3.482734437	0	0.221580637	12.08187723	10.51099181	0.104957849	0	96.02034965
6	1.297859446	4.784182984	2.814058906	0	0.161369908	11.43792739	10.32460396	0.050624597	0	0
7	1.546077433	4.335679667	3.5850388	0	0.146956743	12.59377793	10.19025753	0.051932841	0	61.66357897
8	1.095786527	4.055950875	3.523381121	0	0	11.24468122	12.13576272	0.077852608	0	50.68366489
9	0.76173795	4.584196017	3.299137514	0	0.413221764	10.42754567	11.00206589	0	0	36.70511426
10	1.509806254	5.823433722	3.619692628	3.229935196	0.105228496	11.64664041	11.61022525	0.117716399	0	4.420591332
11	1.097290891	3.843085138	3.271969376	0.57153252	0.197142803	10.49179517	11.04151362	0.057764114	0	0
12	1.324225935	5.401775932	2.926909423	1.459828581	0.367830019	11.52115767	9.26048608	0.160791587	0	46.46892677
13	1.170944091	4.518139891	2.808216574	0	0.285904496	11.01614819	9.96410756	0.282410076	0	62.69096056
14	1.381823333	5.27658812	2.934127052	0.617953656	0.073495444	9.599520636	9.954004922	0.175046297	0	74.3775302
15	1.694890111	5.803141437	3.913668589	2.284622354	0.151382902	11.38791819	11.3985357	0.115925638	0	0
16	1.886783622	5.84314353	3.45690448	0.53476462	0	11.71779422	11.78668286	0.124983432	0	5.109701798
17	1.319039535	4.71655607	2.879283398	0	0.339280424	11.15180696	10.60554073	0.202828135	0	56.48574629
18	1.035453441	5.075045559	2.905980522	0.546403693	0.238290098	11.09395892	11.43033633	0.027038415	0	61.21592604
19	1.029305336	2.998385133	3.88099165	0.111791598	0.262917118	10.79415021	8.825872875	0.161981169	0	70.49641117
20	1.81110564	4.76696427	2.789868633	0.037508897	0.199572876	11.54744673	10.66511246	0.098686966	0	5.877174296
21	1.088461957	4.860334099	3.433315623	0	0.157183281	11.59563767	12.24940107	0.065185419	0	56.34105951
22	1.569014838	5.353890974	2.477872492	1.93577945	0.193616602	10.83747162	9.444938555	0.139833456	0	59.88499185
23	0.808171925	4.544704634	3.263901688	2.127113119	0.123320873	10.67092065	10.66034073	0.14141914	0	25.06070375
24	1.431470778	4.973410412	2.428057843	1.989027137	0.081255101	11.27512191	12.57766673	0.191382887	0	0
25	1.781355133	3.962665781	2.929570145	1.692874703	0.209939048	11.10829834	10.51517876	0.317531064	0	33.00204602
26	0.74815425	5.391519254	1.313109577	1.988800675	0.017694619	11.07611906	10.20285634	0.286521128	0	0
27	1.094581224	5.17065716	3.53749204	0	0.116427404	11.12878849	9.713485433	0.270489672	0	37.45027797
28	0.769142981	5.96039611	1.854174384	1.274066197	0.028491595	10.78012337	11.23616163	0.039112577	0	18.45389656
29	1.205817547	5.81115932	2.918267301	0.672108071	0.274197518	10.30975982	11.79593918	0.127890755	0	72.35235645
30	1.656703789	5.738241557	3.453667257	0.377204107	0.086343535	11.23863419	11.50322394	0.228568286	0	86.72172635
31	1.103016401	5.678440143	2.98777493	2.49786976	0.140649535	12.22004445	10.89718225	0.174144551	0	37.43411286
32	0.753541484	6.008401181	2.4553674	2.28237505	0.026510866	11.53736416	12.18161622	0.078873888	0	0
33	1.504699847	4.752596533	3.145051428	1.839628094	0	11.52874344	11.56909519	0.107379416	0	74.86450297
34	0.792894978	4.420032909	3.931026349	0.861294954	0.125679277	11.35487463	11.13835699	0.121694975	0	36.03028026
35	1.679370669	5.569554407	3.996182732	1.02726565	0.183812062	10.89919599	9.025885876	0.075211032	0	45.47059654
36	1.474704942	3.288821601	2.836537927	1.716765845	0.316819296	10.99765364	11.07731823	0.226403389	0	39.60346761
37	1.432757328	4.390399891	1.770886181	2.980880833	0.420182297	12.37291326	9.19988656	0.329389271	0	39.88614242
38	0.678111017	3.598812675	2.973485401	1.172513269	0.020712271	11.53640407	10.81707562	0.192761111	0	65.14679629

5)

Spurs Data										
Row Labels	**Average of Ast**	**Average of SoT**	**Average of GCA**	**Average of TB**	**Average of Str**	**Average of Int2**	**Average of Blocks**	**Average of ShSv**	**Average of Err2**	**Average of Succ%**
D	0.4	3.6	1.6	1.4	0	15.8	20.2	0.2	0	31.9032967
L	0.181818182	2.545454545	0.545454545	0.272727273	0.363636364	14.54545455	14.72727273	0.181818182	0.090909091	32.80689311
W	2.090909091	6.363636364	4.545454545	2.136363636	0.045454545	14.22727273	15.86363636	0.045454545	0.136363636	25.51452411
Grand Total	**1.315789474**	**4.894736842**	**3**	**1.5**	**0.131578947**	**14.52631579**	**16.10526316**	**0.105263158**	**0.105263158**	**28.466101**

6)

Win Threshold Point Calculation													
Ast	SoT	GCA	TB	Str	Int	Blocks	ShSv	Err	Succ%				
2.090909091	6.36363636	4.54545455	2.13636364	0.04545455	14.2272727	15.8636364	0.04545455	0.13636364	25.5145241	SUM			
0	0	0	0	1	0	0	1	0	0	2		P420	=IF(E420>$P$419,1,0)
0	0	0	1	1	0	0	1	0	1	4		Y420	=IF(N420<$Y$419,1,0)
0	1	0	0	1	0	0	1	0	1	4		Z420	=SUM(P420:Y420)
0	1	0	0	1	1	1	1	0	0	5			
1	0	1	0	1	1	1	1	0	1	7			
0	0	0	0	1	0	1	1	0	1	4			
0	0	0	1	1	1	0	1	0	1	5			
0	1	1	0	1	0	1	1	0	0	5			
0	1	0	0	1	1	0	1	0	1	5			
1	1	0	1	1	0	0	1	0	0	5			
1	0	1	0	1	1	1	1	0	1	7			
0	1	0	0	1	1	1	1	0	0	5			
0	0	0	0	1	1	1	1	0	1	5			
1	0	1	1	1	0	0	1	1	1	7			
1	1	1	1	1	0	0	0	1	0	6			
1	1	1	0	1	1	0	1	0	1	7			
0	0	0	1	1	1	1	1	0	1	6			
1	0	1	0	0	1	1	1	0	1	6			
1	0	1	0	1	0	0	1	1	1	6			
1	1	1	0	1	1	0	1	0	0	6			
0	0	0	0	1	1	1	1	0	0	4			
1	1	1	1	1	0	1	1	0	1	8	AVG	5.40909091	=AVERAGE(Z420:Z441)

Lose Threshold Point Calculation													
Ast	SoT	GCA	TB	Str	Int	Blocks	ShSv	Err	Succ%				
0.181818182	2.54545455	0.54545455	0.27272727	0.36363636	14.5454545	14.7272727	0.18181818	0.09090909	32.8068931	SUM			
1	1	1	1	0	1	1	0	1	1	8		P447	=IF(E442<$P$446,1,0)
1	1	1	0	0	1	1	1	1	1	8		Y447	=IF(N442>$Y$446,1,0)
0	0	0	0	0	0	0	0	1	0	1		Z447	=SUM(P449:Y449)
1	0	1	1	0	0	0	0	1	1	5			
1	1	1	1	1	1	0	0	1	0	7			
1	0	1	1	0	0	1	0	1	1	6			
1	0	1	1	0	1	1	0	1	1	7			
0	0	0	1	0	1	0	1	1	0	4			
1	0	1	0	1	1	0	0	1	0	5			
1	1	0	1	0	0	0	0	1	1	5			
1	1	1	1	0	0	1	0	0	1	6	AVG	5.63636364	=AVERAGE(Z447:Z457)

7)

M91	=IF(B91<GETPIVOTDATA("Average of Ast",$A$39,"W/L","L"),0,IF(B91<GETPIVOTDATA("Average of Ast",$A$39,"W/L","D"),1,3))
Q91	=IF(F91<GETPIVOTDATA("Average of Str",$A$39,"W/L","W"),1,IF(F91<GETPIVOTDATA("Average of Str",$A$39,"W/L","L"),3,1))
W91	=COUNTIF(M91:V91,3)

W Threshold	L Threshold
5.40909091	5.63636364

Calculate Thresholds

Ast	SoT	GCA	TB	Str	Int	Blocks	ShSv	Err	Succ%	Count 3	Count 0	WLD
3	3	3	0	3	3	0	0	0	0	5	5	1
3	3	3	0	3	3	0	0	0	3	6	4	3
3	3	3	3	3	3	0	3	0	0	7	3	3
3	3	3	3	3	3	0	3	0	0	7	3	3
3	3	3	1	1	3	0	0	0	3	5	3	1
3	3	3	1	3	3	0	0	0	3	6	3	3
3	3	3	1	3	3	0	0	0	1	5	3	1
3	3	3	1	1	3	0	3	0	1	5	2	1
3	3	3	1	1	3	0	0	0	0	4	4	1
3	3	3	3	1	3	0	0	0	0	5	4	1
3	3	3	3	1	3	0	0	0	3	6	3	3
3	3	3	0	1	3	0	0	0	1	4	4	1
3	3	3	3	1	3	0	0	0	1	5	3	1
3	3	3	0	3	3	0	0	0	3	6	4	3
3	3	3	0	3	3	0	3	0	0	6	4	3
3	3	3	1	3	3	0	0	0	3	6	3	3
3	1	3	1	1	3	0	3	0	0	4	3	1
3	3	3	1	3	3	0	0	0	0	5	4	1
3	3	3	3	3	3	0	0	0	0	6	4	3
3	3	3	1	1	3	0	0	0	3	5	3	1
3	3	3	3	1	3	0	0	0	0	5	4	1
3	3	3	3	3	3	0	0	0	3	7	3	3
3	3	3	0	3	3	0	0	0	3	6	4	3
3	3	3	1	3	3	0	0	0	0	5	4	1
3	3	3	3	3	3	0	3	0	0	7	3	3
3	3	3	3	3	3	0	3	0	0	7	3	3
3	3	3	0	3	3	0	3	0	3	7	3	3
3	3	3	3	1	3	0	3	0	3	7	2	3
3	3	3	3	1	3	0	0	0	0	5	4	1
3	3	3	3	3	3	0	0	0	3	7	3	3
3	3	3	1	3	3	0	3	0	3	7	2	3
3	3	3	0	3	3	0	0	0	0	5	5	1
3	3	3	3	3	3	0	0	0	0	6	4	3
3	3	3	0	1	3	0	0	0	3	5	4	1
3	3	3	3	1	3	0	0	0	3	6	3	3
3	3	3	0	3	3	0	0	0	3	6	4	3
3	3	3	3	1	3	0	0	0	3	6	3	3
3	3	1	1	1	3	0	0	0	3	4	3	1

Tottenham earns 81 points after the new squad. This is higher than Chelsea's points so we can assume Tottenham can have better standings the next season.

Simulation		Points
1	SimTable	80
2	0	74
3	0.001002	78
4	0.00200401	80
5	0.00300601	74
6	0.00400802	74
7	0.00501002	84
8	0.00601202	82
9	0.00701403	88
999	0.998998	76
1000	1	84

MIN POINTS
81.04