HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 4 (2022)

Testing the Validity of Markov Chains in Analyzing Soccer Gameplay through Barcelona's 2020-2021 La Liga Season

Kevin Yuan[1], Isabella Halaby[1] and Sema Duzyol[#]

[1] Fulton Science Academy Private School, Alpharetta, GA, USA
[#]Advisor

ABSTRACT

Statistical modeling has been applied to most or all modern-day sports in order to analyze gameplay and offer teams an upper hand in games. Though games such as baseball have had mass changes due to Sabermetrics, the empirical analysis of baseball, there still remain numerous components of soccer where mathematics can be more deeply integrated. Though current literature and research like Sanyal's "Who will receive the ball? Predicting recipients in soccer videos" do provide some form of analysis, many of these methods have limitations due to perspective or robustness. To counter these limitations, Markov Chains were implemented. As they disregard the players as a factor and can take in theoretically infinite amounts of data, Markov Chains seem to be a superior method to the previous ones suggested by current literature. By applying the data from Barcelona's 2020-2021 La Liga Season to generate a transition matrix, it was seen that Markov Chains were effectively able to deduce patterns in soccer gameplay. However, as it only provides a transition matrix, giving the probabilities that the ball travels from one section of the field to the next, it does not provide a direction countering strategy to an opponent but rather the basis for which a strategy can be created.

Introduction

Statistical analysis and modeling have become heavily integrated with sports in recent decades. Through the use of prediction, these analysis and models offer teams an upper hand in games. As the majority of teams have begun incorporating mathematics in their strategies, sports have been overhauled. Sabermetrics, the empirical analysis of baseball invented in 1980, for example, which gives "forecasts result by making predictions based on previous data ", has led to new rules in Major League Baseball (MLB) (Guess, 2015). Sabermetrics has led to the invention of new tactics which have been so effective that the league has attempted to altogether ban the tactics that sabermetrics encourages. For instance, the shift, when a team strategically placed position players in spots on the field where a batter is likely to hit the ball, has yielded such positive defensive returns that it will be banned from 2023 onwards (Joseph, 2022; Koch, 2021). With its proven effectiveness, other teams have adopted analysis and modeling to improve themselves. In soccer for instance, the market value of a player is solely determined through analytics, which takes into account the expected goals the player can produce among other factors (Soccerment Research, 2020). Soccer game statistics have also vastly improved. Case in point, players are now not only assessed based on basic statistics like assists and goals but also more intricate ones such as key passes and big chances. (Brownell, 2017). Yet, despite the installation of market values and introduction of new statistics, there still remain many components of soccer where mathematics can be more deeply integrated. One component that has been heavily scrutinized by analytics is predicting penalty kicks. Statisti-

cians find the element of the game captivating as many different models can be applied in hopes of predicting/forecasting where the opponent will shoot the ball based on certain parameters (i.e. foot kicked with), thus enabling them to create optimal strategies for goal keepers (Hunter et al., 2018)

In a similar fashion, predicting the movement of the ball during a game has garnered much attention, leading to implementations and creations of many new analytical models. Samriddha Sanyal, for instance, created a probabilistic framework for pass prediction using "two dependent models designed from the coordinates of the players." Based on the coordinates of any two given players. Sanyal created a model that forecasted the most probabilistic next pass. The prediction of where the ball will travel next is highly useful to coaches and players. Not only does it allow a team to analyze and thus prepare for an opponent, but it can also allow a team to better understand themselves. However, despite the numerous different models that have been produced, these approaches often have their limitations. For example, the limitations of the two dependent models that Sanyal used is that one could not demonstrate the robustness of a pass network while the other model could not be applied to numerous players at once (Sanyal, 2021).

To address the limitations many models hold, this paper implements Markov Chains as a method to predict the most probable location of the ball. Since Markov Chains are hypothetically able to take in an infinite amount of data points, as long as the used data set is robust, the generated matrix will be sufficient in predicting the next location of the ball. Moreover, as this study purely uses the location to where the ball travels, it is not limited by the number of players taken into account. Thus, given its foretold benefits, we sought to test the validity of Markov Chains in helping statisticians analyze soccer teams by implementing the model in a series of several games.

Dataset

In this study, it was crucial that a proper data set be chosen in order to ensure that there were enough data points to generate an effective matrix. As we were analyzing the effectiveness of Markov Chains as a model for soccer analysis, we chose to have our data specialize on a singular team so that the eventual matrix derived had purpose. If the data set had instead included numerous teams, the matrix would not be applicable to any of the teams as the pattern is derived from combinations of datasets from both teams. Thus, some patterns found in a union of the sets may not prevail in any single of the data sets.

As it would be vastly difficult to obtain the data by hand from rewatching prerecorded games and marking down the sector changes every time the ball moved, the Stats Bomb Soccer open data set was used. The Stats Bomb open data source offered statistics for Barcelona between the years 2009-2021. Moreover, since a team's starting players and strategies mostly remain constant throughout a single year, the data set chosen was the 32 games from the Stats Bomb Soccer open data source of Barcelona's 2020-2021 La Liga Season. Increasing the data set to two seasons, though would give a larger data set, may reduce the effectiveness of the pattern created as the strategies and players often change between two seasons.

Data Extraction

Though the data set was quite simple to choose, the extraction of the data into a form that could be easily utilized to code with was more difficult. In the open-source format, the data came on a JSON (an example pictured below):

	index	team	minute	type	location	pass_end_location
0	0	Barcelona	0	Starting XI	NaN	NaN
1	1	Deportivo Alavés	0	Starting XI	NaN	NaN
2	2	Barcelona	0	Half Start	NaN	NaN
3	3	Deportivo Alavés	0	Half Start	NaN	NaN
4	4	Barcelona	45	Half Start	NaN	NaN
5	5	Deportivo Alavés	45	Half Start	NaN	NaN
6	6	Deportivo Alavés	0	Pass	[61.0, 40.1]	[42.3, 42.4]
7	7	Deportivo Alavés	0	Pass	[42.3, 42.4]	[84.2, 12.8]
8	8	Deportivo Alavés	0	Pass	[84.2, 12.8]	[91.7, 20.4]
9	9	Barcelona	0	Pass	[27.5, 55.3]	[9.9, 42.4]
10	10	Barcelona	0	Pass	[10.7, 43.0]	[25.9, 15.7]
11	11	Barcelona	0	Pass	[17.2, 3.4]	[30.8, 10.2]
12	12	Barcelona	0	Pass	[31.1, 9.9]	[23.1, 10.6]
13	13	Barcelona	0	Pass	[23.1, 10.6]	[24.7, 18.8]
14	14	Barcelona	0	Pass	[24.7, 18.8]	[9.1, 19.2]
15	15	Barcelona	0	Pass	[11.3, 19.8]	[15.0, 52.6]
16	16	Barcelona	0	Pass	[41.3, 29.7]	[54.0, 32.8]
17	17	Barcelona	0	Pass	[54.0, 32.8]	[43.5, 29.1]

Figure 1. JSON of Raw Data. The table above shows the first 18 rows of the JSON file by Stats Bomb of a game between Barcelona and Deportivo Alaves

Since all the JSON files were extracted through the python library supplied by Stats Bomb, all coding except the matrix creation (mentioned later in the methods section) was done in python. All variables and co-ordinate values used will be explained in the methods sections as well. Note the first 6 rows of data present no value to this study as it simply indicates the game has started. Moreover, the rows are sorted chronologically with the lower the index number, the earlier the event happens. From row 6 onwards, each row reveals the team that made the pass, the starting location of the pass, and ending location of the pass. For example, in row 6, Deportivo Alves made a pass from (61.0, 40.1) to (42.3, 42.3).

Since the data would be much easier to manipulate if it were in an excel format, it was first necessary to convert the given data into an excel sheet. Unfortunately, since the Stats Bomb data was presented in arrays, an online JSON to XLS converter could not be used as no converters could read the data correctly. Thus, a python program was written to parse through the columns and write the data in the columns desired. The following is the small segment of code used to parse through columns:

```
for x in range (len(df['index'])):
    pass_end_locationX = events.at[row, 'pass_end_location'][1]
    pass_end_locationY = events.at[row, 'pass_end_location'][0]
    team = events.at[row, 'team']
    pass_start_locationY = events.at[row, 'location'][1]
    pass_start_locationX = events.at[row, 'location'][0]
```

Figure 2. Data Extraction Code**.** The segment of code above is the code used to parse through the Stats Bomb JSON folder and convert it into an Excel file.

Note, however, that there were two variables each for the “pass_end_location” and “pass_start_location” that denoted the X and Y coordinate of each value since the pair of coordinates were presented in an array in the JSON file. The code segment would then open a new excel file creating and filling in the columns x, y, endX, endY, and team. The following excel segment is what was produced by the code based on the above JSON segment:

x	y	endX	endY	team
61	40.1	42.3	42.4	Deportivo Alavés
42.3	42.4	84.2	12.8	Deportivo Alavés
84.2	12.8	91.7	20.4	Deportivo Alavés
27.5	55.3	9.9	42.4	Barcelona
10.7	43	25.9	15.7	Barcelona
17.2	3.4	30.8	10.2	Barcelona
31.1	9.9	23.1	10.6	Barcelona
23.1	10.6	24.7	18.8	Barcelona
24.7	18.8	9.1	19.2	Barcelona
11.3	19.8	15	52.6	Barcelona
41.3	29.7	54	32.8	Barcelona
54	32.8	43.5	29.1	Barcelona
43.5	29.1	47.2	17.3	Barcelona
49.7	15.2	35.4	32.8	Barcelona
35.4	35.6	37.9	55.7	Barcelona
44.8	55.4	41	38	Barcelona

Figure 3. Excel of Raw Data. The table above shows the Excel version of the initial JSON file shown above after the JSON file was run through the data extraction program.

Methods

In all data sets, the field dimensions used was a 120 x 80 as pictured below:

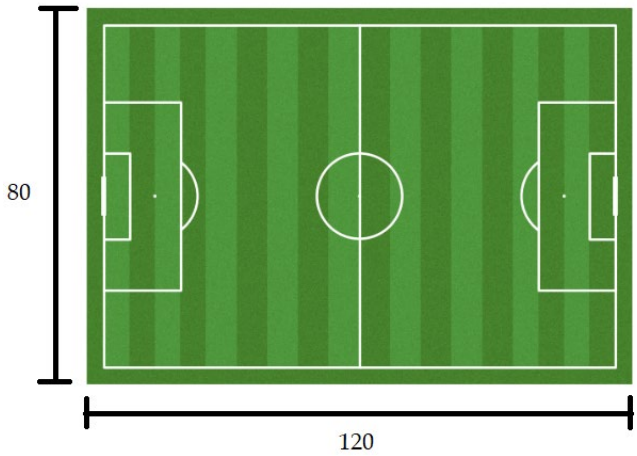

Figure 4. Field with Dimensions. The soccer field above is used throughout the entirety of the paper with the dimensions of 120x60. A coordinate of value (60, 40) would thus be at the center of the field.

Referring to the previous coordinates used in the data extraction section, the coordinate (61, 40.1) essentially marks the center of the field. However, as this is the beginning position of the ball, it was slightly off-center to represent which team starts with the ball.

The field was then split into a 4x4 with each sector being 30x20 and numbered accordingly as pictured:

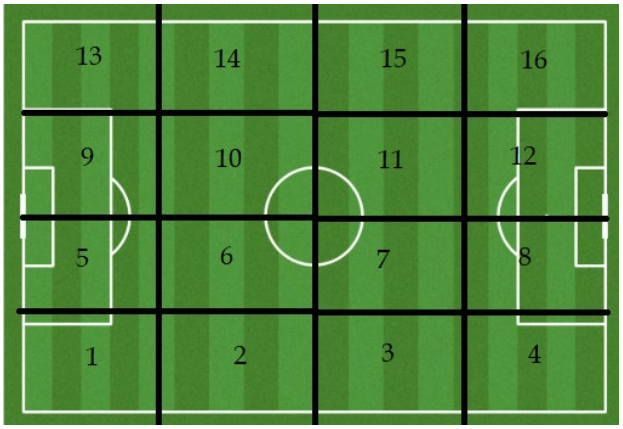

Figure 5. Field with Sector Numbering. The soccer field above is identical to that in figure 4. However, this field displays the 4x4 layout of the sector numbering used. For example, a coordinate value of (10, 10) would fall into sector 1.

We now had every pass by Barcelona in the 2020-2021 La Liga season. Though this did not directly create a series that was needed for a Markov Chain, it was not difficult to create one given the beginning and ending coordinate of each pass. To understand the code utilized, let us first define a "successful" string as the longest series of passes in which Barcelona maintains 100% possession.

	x	y	endX	endY	team
2	61	40.1	60.8	42.4	Villarreal
3	62.1	42.6	50.6	28.1	Villarreal
4	48	28.7	32.1	38.9	Villarreal
5	27.6	36.1	37.6	37.3	Villarreal
6	32.7	33.4	17.8	12.8	Villarreal
7	20.2	13.9	14.9	80	Villarreal
8	105.7	0.1	98.6	4.6	Barcelona
9	96	15.7	80.7	6.5	Barcelona
10	78.1	7.3	50.1	30.4	Barcelona
11	50.7	30.2	56.4	10.3	Barcelona
12	57.4	10.1	67.3	13.9	Barcelona
13	66.6	13	56.4	8.1	Barcelona
14	50.3	9.2	35.5	25.5	Barcelona
15	33.8	28.5	29.7	51.8	Barcelona
16	29.5	54.2	39.9	72.8	Barcelona

Figure 5. Excel Table of Barcelona vs Villarreal from Index 2 to 16. The table above resembles the table in figure 3 as it is essentially the same table except with the index numbers to the left and the data coming from a game of Barcelona vs Villareal.

41	77.8	53.6	93.2	20.2	Villarreal
42	114.2	22.4	113	24.9	Villarreal
43	120	0.1	111.8	47.5	Villarreal
44	6	36	8.5	29.3	Barcelona
45	10.3	25.3	19.1	9.9	Barcelona
46	21	6.7	11.8	18.1	Barcelona
47	12.6	16.9	2.4	31.8	Barcelona
48	8.1	39	17.7	66.8	Barcelona
49	19.8	69.5	31.3	76.3	Barcelona
50	32.2	76	37.9	67.3	Barcelona
51	37.9	67.3	18.9	70.8	Barcelona
52	18.1	71	2.6	54.4	Barcelona
53	13.6	43.8	49.9	13.8	Barcelona
54	70.2	66.3	89	72.2	Villarreal
55	104.9	80	107.5	68.5	Villarreal

Figure 6. Excel Table of Barcelona vs Villarreal from Index 41 to 55. The table above comes from the same excel sheet presented in figure 5. However, the snap shot here comes from index numbers 41 to 55.

From the example in figure 5, a series from index 7-16 would not be considered successful since row 7 indicates a pass that Villareal made, meaning Villareal held possession during that pass. However, a series from index 8-16 would be considered successful since within all of those rows, Barcelona maintains possession. Moreover, from the continuation of the excel table presented in figure 6, a successful series would start from index 44 and end at index 53. A series that begins at index 45 and ends at 53 would not be considered successful however since it is not the *longest* series of passes made in which Barcelona maintains possession. Thus, in the figures 5 and 6, the two successful series would be from index 7-16 and from index 44-53. Now, let us iterate across the successful string above to define which sector each coordinate lies in. As there are (16-8+1) = 9 rows, we should expect to have 18 sector numbers as each row has both a beginning coordinate and ending coordinate.

```
def position(x, y):
    if x <= 30:
        if y <= 20:
            return "1"
        elif y <= 40:
            return "5"
        elif y <= 60:
            return "9"
        elif y <= 80:
            return "13"
    if x <= 60:
        if y <= 20:
            return "2"
        elif y <= 40:
            return "6"
        elif y <= 60:
            return "10"
        elif y <= 80:
            return "14"
    if x <= 90:
        if y <= 20:
            return "3"
        elif y <= 40:
            return "7"
        elif y <= 60:
            return "11"
        elif y <= 80:
            return "15"
    if x <= 120:
        if y <= 20:
            return "4"
        elif y <= 40:
            return "8"
        elif y <= 60:
            return "12"
        elif y <= 81:
            return "16"
```

Figure 7. Code Segment for Determining which Sector a Coordinate is in. The code to the left is a simple sorting algorithm that determines which sector a coordinate lies in based on the parameters in figure 5. For example, a coordinate of (50, 40) would fall under the "<= 60" section for the x value and "<= 40" section for the y value, leading to a sector value of 6.

By running the successful string in the code shown in figure 5, the sector series 4, 4, 4, 3, 3, 6, 6, 2, 2, 3, 3, 2, 2, 6, 6, 9, 9, 14 is generated. To obtain all the sector series for Barcelona within an excel file, we have to parse through the entirety of the excel file, determine all the successful strings that lie within the excel file, and convert the successful strings into sector series. Once we have found all the sector series however, we cannot simply connect them together to form one large string as the ball does not travel from the end of one sector series to the beginning of the sector series. Rather, in between each sector series, Barcelona loses possession of the ball. Thus, we will denote the separation between series with a written statement "lost possession".

There is one more factor that must be considered when determining the sector series given the successful string. The field presented in figures 4 and 5 assume that Barcelona attacks to the right (the left goal is their own). However, Barcelona does not always play with the left goal being theirs. It is important to take this into account as a pass from sector 3 to 4, for example, should be a pass from 13 to 14 if this problem was attributed for. Thus, when running the code in figure 7, it is crucial to adjust the sector numbering based on which side Barcelona starts from. Moreover, teams switch sides at half time. As JSON file from Stats Bomb

also provided when a new half began, this was straightforward to account for since the sector numbering was simply flipped during half time.

```
for x in range (len(df['x'])-1):
    if (df['team'][x] == "Barcelona") and not (df['team'][x+1] == "Barcelona"
        print (position(df['x'][x], df['y'][x]))
        print (position(df['endX'][x], df['endY'][x]))
        print ("lost possession")
    elif (df['team'][x] == "Barcelona"):
        print (position(df['x'][x], df['y'][x]))
        print (position(df['endX'][x], df['endY'][x]))
    else:
        pass
```

Figure 8. Code Segment for Determining the Sector Series. The code above takes in the converted excel file for a certain game and converts the successful strings into sector series that have accounted for the problems aforementioned. The code also places a "lost possession" line segment between each sector series to indicate where Barcelona lost possession.

The code in figure 8 accomplished everything in the aforementioned couple paragraphs and generated a column of data along the lines of:

```
14
14
14
13
13
9
9
2
lost possession
1
1
1
2
lost possession
9
9
9
```

Figure 9. Sector Series Printed from code in Figure 8. The column of data to the left is an example of what the code segment from figure 8 would print. The first 8 rows of number (14, 14, 14, 13, 13, 9, 9, 2) is one sector series where the next four rows after the "lost possession" is another sector series.

By running the total of 38 La Liga games, now converted into excel sheets, through the above series of code segments, we obtained a large list of roughly 22,000 total rows (a combination of both sector numbers and "lost possessions"). The data was then saved into a text file. A 17x17 matrix as there are 16 different sectors and one row and column to account for the axis labeling was then created to count the instances the ball left from one sector to another.

A sorting algorithm was created in IntelliJ and implemented to create the 17x17 matrix. By parsing through each row of the text file, the algorithm would count the instances a ball left on sector number and entered a number, tallying the total and inputting it into a 17x17 matrix.

```
public class SoccerPassAnalysis
{
    private static List<int[][]> passDistributions = new ArrayList<>();

    private static void printPassDistribution(int idx, int[][] passDistribution)
    {
        System.out.println("Pass distribution " + idx);
        System.out.println("T\\F\t1\t2\t3\t4\t5\t6\t7\t8\t9\t10\t11\t12\t13\t14\t15\t16");
        for (int toPos = 0; toPos < 16; ++toPos)
        {
            System.out.print(toPos+1);
            System.out.print("\t");
            for (int fromPos = 0; fromPos < 16; ++fromPos)
            {
                System.out.print(passDistribution[fromPos][toPos]);
                System.out.print("\t");
            }
            System.out.print("\n");
        }
    }
```

Figure 10. Java code for Creating the Matrix. The code above was done in IntelliJ and was used to create/print the actual matrix by creating a 17x17 table (one extra row and column for the axis labeling).

The code above created a 17x17 matrix where the columns indicated the sector number where the ball originates and where the row number indicated the sector number where the ball next traveled. For example, if there were 20 instances where the ball traveled from sector number 4 to sector number 5 immediately, then the data cell of the 4th column and 5th row would have value 20.

```
int LineNumber = 0;
int totalPass = 0;
try
{
    List<String> allLines = Files.readAllLines(Paths.get(filePath));
    int[][] passDistribution = new int[16][16];
    for (int fromPos = 0; fromPos < 16; ++fromPos)
    {
        for (int toPos = 0; toPos < 16; ++toPos)
        {
            passDistribution[fromPos][toPos] = 0;
        }
    }
    int from = -1;
    int to;
    for (String line : allLines)
    {
        ++LineNumber;
        if (line.trim().isEmpty())
        {
            continue;
        }
```

```
if (line.compareToIgnoreCase( str: "lost possession") == 0)
{
    from = -1;
}
else
{
    if (from == -1)
    {
        from = Integer.parseInt(line);
    }
    else
    {
        to = Integer.parseInt(line);
        ++passDistribution[from-1][to-1];
        from = to;
        ++totalPass;
    }
}
}
```

Figure 11. Java Code for Counting Instances Ball Travels Between Sectors. The code to the left read through all rows in the text file given, counted the number of instances a ball left a sector and entered another, and inputted the value into the matrix created in figure 12.

The code parsed each row of the text file of the sector series and added one to the data cell that corresponded to the column sector of the preceding sector number and the row sector of the succeeding sector number. For example if the first 3 rows of the text file were 1, 3, and 5, the code in figure 11 would first add one to the cell corresponding to column 1 and row 3 and then add one the cell corresponding to column 5 and row 3. As "lost possessions" should not be counted into the matrix, whenever the code parsed to a row containing the two words, it would set the corresponding row/column to -1. As this was out of bounds, the code essentially would just skip over this line. For example, if there were 5 rows that followed 1, 2, lost possession, 4, 5, the code would add one to the data cell corresponding to column 1 and row 2, attempt to add one to the data cell corresponding to column 2 and row -1, attempt to add one to the data cell corresponding to column -1 and row 4, and add one to the data column corresponding to the column 4 and row 5. Running this code for the entirety of the text file produced a 17x17 matrix.

Results

T/F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	526	104	2	0	187	26	1	0	44	1	0	0	3	0	0	0
2	203	1579	197	2	154	428	31	6	41	85	8	1	4	7	0	0
3	14	380	3617	336	24	244	731	59	2	32	85	10	0	3	11	2
4	0	25	423	1603	5	11	286	318	1	8	46	43	0	0	7	8
5	186	104	4	0	935	92	1	0	304	45	5	0	24	11	1	0
6	65	464	132	3	255	1816	130	4	100	556	60	1	6	59	18	0

7	9	105	839	112	22	333	2958	160	14	141	698	48	0	6	67	4
8	0	9	153	521	2	12	315	1403	1	4	142	198	0	3	15	63
9	28	11	1	0	294	50	2	1	906	75	3	0	160	96	3	0
10	13	81	29	0	100	544	68	3	241	1783	124	4	42	449	122	1
11	1	9	76	7	12	126	642	46	19	329	2631	130	13	76	674	134
12	0	0	16	71	0	11	111	193	5	12	275	977	2	6	105	390
13	1	0	0	0	40	2	0	0	133	11	0	1	459	74	4	1
14	3	13	5	0	45	85	3	1	123	388	22	4	143	1339	203	2
15	0	4	11	1	3	34	100	11	18	237	572	27	7	343	2937	286
16	0	0	12	14	1	4	55	44	1	12	204	246	3	13	405	1303

Figure 13. Matrix of Occurrences

T/F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.5014	0.0360	0.0004	0	0.09	0.0068	0.0002	0	0.0225	0.0003	0	0	0.0035	0	0	0
2	0.1935	0.5467	0.0357	0.0007	0.0741	0.1121	0.0057	0.0027	0.021	0.0229	0.0016	0.0006	0.0046	0.0028	0	0
3	0.0133	0.1316	0.6556	0.1258	0.0115	0.0639	0.1345	0.0262	0.0010	0.0086	0.0174	0.0059	0	0.0012	0.0024	0.0009
4	0	0.0087	0.0767	0.6004	0.0024	0.0029	0.0526	0.1414	0.0005	0.0022	0.0094	0.0254	0	0	0.0015	0.0036
5	0.1773	0.0360	0.0007	0	0.4497	0.0241	0.0002	0	0.1557	0.0121	0.0010	0	0.0277	0.0044	0.0002	0
6	0.062	0.1607	0.0239	0.0011	0.1227	0.4756	0.0239	0.0018	0.0512	0.1495	0.0123	0.0006	0.0069	0.0237	0.0039	0
7	0.0086	0.0364	0.1521	0.0419	0.0106	0.0872	0.5444	0.0711	0.0072	0.0379	0.1432	0.0284	0	0.0024	0.0147	0.0018
8	0	0.0031	0.0277	0.1951	0.001	0.0031	0.058	0.6238	0.0005	0.0011	0.0291	0.1172	0	0.0012	0.0033	0.0287
9	0.0267	0.0038	0.0002	0	0.1414	0.0131	0.0004	0.0004	0.4639	0.0202	0.0006	0	0.1848	0.0386	0.0007	0
10	0.0124	0.0280	0.0053	0	0.0481	0.1425	0.0125	0.0013	0.1234	0.4794	0.0254	0.0024	0.0485	0.1807	0.0267	0.0005
11	0.001	0.0031	0.0138	0.0026	0.0058	0.0330	0.1181	0.0205	0.0097	0.0885	0.5397	0.0769	0.0150	0.0306	0.1474	0.0611

12	0	0	0.0029	0.0266	0	0.0029	0.0204	0.0858	0.0026	0.0032	0.0564	0.5781	0.0023	0.0024	0.023	0.1778
13	0.0001	0	0	0	0.0192	0.0005	0	0	0.0681	0.003	0	0.0006	0.5300	0.0298	0.0009	0.0005
14	0.0029	0.0045	0.0009	0	0.0216	0.0223	0.0006	0.0004	0.063	0.1043	0.0045	0.0024	0.1651	0.5388	0.0444	0.0009
15	0	0.0014	0.002	0.0004	0.0014	0.0089	0.0184	0.0049	0.0092	0.0637	0.1173	0.016	0.0081	0.1380	0.6424	0.1304
16	0	0	0.0022	0.0052	0.0005	0.0010	0.0101	0.0196	0.0005	0.0032	0.0418	0.1456	0.0035	0.0052	0.0886	0.5939

Figure 14. Markov Chain Produced Rounded to the Nearest Thousandths

Discussion

The main purpose of this paper was to attempt to create a method or a more intricate manner for analyzing soccer game play. The ability for Markov Chains to take in a mass amount of data evades the issue regarding robustness and limited perspective (limited to 2 or 3 player analysis) that most other methods held. The 16x16 matrix produced through this study's process affirms the hypothesized power that Markov chains hold in soccer game analysis as not only is the matrix robust in that it gives individualistic breakdowns for each cell, which can even be increased if the field were divided into more cells, but it also expands the perspective by eliminating players altogether, thus circumventing this issue. Analysis through a broadened perspective is also much more useful as it is able to negate issues regarding player biases and habits and analyze the team altogether.

The transition matrix that this method yields can help break down the strategies that teams use and how to best counter them. It may seem that by observing simply where the probabilities spike in the chart, one can conclude where the other teams like to play the ball the most and thus be more prepared. For example, in this case, Barcelona tends to play the ball from zone 14 to zone 13 (evidenced by the cell having the second highest value within the column), which may hint that Barcelona likes to control the midfield. This postulate is further evidenced by the cell symbolizing passes from zone 6 to 7 having the second highest value within the column. However, it is also important to note that sections bordering one another hold high probabilistic values. Logically, this is because passes are often short, leading them to leave and enter bordering sections. Thus, analysis of the other team's strategy requires more than viewing the spikes as those may simply be derived from having split the field into a low number of sections.

Seen in the previous trend, the cells we examined often held the second highest value within the column. This is because the diagonals in the transition matrix hold the highest probabilities. This is reasonable since the field was only split into 16 total sections, meaning whenever the ball moved (whether that be from dribbling or from passes) they occurred within the same section due to large areas each section held. If the field were to be split into more sections, the issue regarding the diagonals having exceptionally high probabilities would be avoided as the section area would be decreased and passes would be bound to enter a new section when exiting another one.

Though this model does not yield a direct way in concluding and countering the opponent's strategy, implementing Markov Chains serves as a system to gauge where the opponent often likes to play the ball, which in turn allows one to formulate a strategy to "outplay" the opponent, knowing where they will likely play the ball given its current location.

Limitations and Future Studies

Despite the matrix generated in this paper displaying data that could be widely implemented in strategies to counter Barcelona, there are a few limitations that do appear. Some have been addressed within the conclusions

section regarding some spikes being "false" spikes since the field was only split into 16 sections. Thus, we will not go over those limitations again.

The most notable limitation comes from the actual implementation of Markov Chains. Each year, new players and possibly new coaches are brought onto a team which mildly or completely alters the tactics that the team implements. Thus, unless the main starting squad and coach is kept, previous year data is often useless. Consequently, each year a new Markov Chain needs to be generated if an effective transition matrix is to be generated. Though this limits implementation of the Markov Chains early on within the year, after a few games, which generates thousands of data points, the transition matrix slowly becomes more applicable.

As this study was to test whether Markov Chains were a viable method to generate only the rudimentary of this process have been tested. Mentioned in the conclusion portion, splitting the field into more sections would provide more applicable data as it would logically make the transition matrix much more intricate. Thus, future studies could divide the field into more sections to see whether it truly does improve the data. Moreover, as established, the Markov Chain only provides a base for analysis on the opposing team. Consequently, the data is not useful until an algorithm or method is created to analyze and create a counter strategy for a team based on opposing team data.

Acknowledgments

I would like to thank Susan Duderstadt and Ramin Ahmadoghlu for introducing and guiding me through the research process. I would also like to thank Mrs. Duzyol who has helped me along my entire mathematics journey such that I could write this paper. Without her, I would only have a whimsical mind without a guide. Finally, I would like to thank the many great teachers who have shaped me into the curious, passionate person I am today. I'd like to give a special shoutout to Kelley Rodgers, my social studies teacher who taught me to question everything and derive my own understanding, and Christopher Craig, my literature teacher who always kept my mind active and delivered a good laugh occasionally.

References

Brownell, P. (2017, October 3). *The most important new advanced soccer statistics and why they matter.* Bleacher Report.

Guess, A. R. (2015, November 4). *Infographic: How Sabermetrics has changed baseball.* DATAVERSITY.

Hunter, A. H., Murphy, S. C., Angilletta, M. J., & Wilson, R. S. (2018, July 29). *Anticipating the direction of soccer penalty shots depends on the speed and technique of the kick.* Sports (Basel, Switzerland). https://doi.org/10.3390/sports6030073

Joseph, A. (2022, March 7). *Is MLB banning the defensive shift really going to make a difference in baseball?* USA Today.

Koch, T. (2021, March 13). *MLB: How to strip sabermetrics of some of its power.* Call to the Pen.

Sanyal, S. (2021). Who will receive the ball? predicting pass recipient in soccer videos. *Journal of Visual Communication and Image Representation*, *78*. https://doi.org/10.1016/j.jvcir.2021.103190

Soccerment Research. (2020, June 2). *The growing importance of football analytics.* Soccerment.

Toda, K., Teranishi, M., Kushiro, K., & Fujii, K. (2022). Evaluation of soccer team defense based on prediction models of Ball Recovery and being attacked: A pilot study. *PLOS ONE*, *17*(1). https://doi.org/10.1371/journal.pone.0263051