Kang Min Seong

How effective is the Monte Carlo Method for predicting the outcome of the football tournament?

Table of contents

Introduction

My extended essay touches on the research question "How effective is the Monte Carlo method for predicting the outcome of the football tournament?" The power of the Monte Carlo method (MCM) is a topic of interest not only for mathematicians but also for people from other fields, because it offers a variety of implementation including prediction of results of an event basing on probabilistic data such as a probability density function. In my research I explore the usage of the method on a real-life situation, more concretely – predicting the outcome of the football tournament.

MCM is defined as a technique in which a large quantity of randomly generated numbers are studied using a probabilistic model to find an approximate solution to a numerical problem that would be difficult to solve by other methods.[1] I have learnt about the existence of the method on one of the news articles that was about the betting strategy based on the MCM, which was applied to the probabilistic model made from the betting history of people.[2] While reading about it, I was impressed by this method. Then, in one of the casual chats with my friends, we touched on the topic of football. Being fierce football fans, we argued about the result of the La Liga in the season 2017/2018. After the discussion, I recalled the article about the MCM and thought that it would be very interesting to apply the method to predict the result of the real football tournament.

In the first part of the essay I research into the MCM in general. Then I focus my work on pseudorandom numbers (PRN), which is a crucial component of the MCM. Further on I explore the Linear Congruential Generator used to produce PRN and justify the need in a large sample of PRN by the Law of Large Numbers.

In the second part of the essay I model the probability of La Liga (Spanish national football league) teams winning each other using Poisson distribution. After, I define the range of possible inputs and generate random inputs from the pseudorandom number generator stated in the first part. Then I conduct the MCM with this distribution and random inputs to get the prediction of the standings of the La Liga in season 2017/2018.

In the conclusion, I evaluate my prediction by comparing it to the actual standings of La Liga teams. Then I reflex on limitations of the method, and factors that could affect the accuracy of my prediction.

Monte Carlo Method

As it was stated before, MCM is defined as a technique in which a large quantity of randomly generated numbers are studied using a probabilistic model to find an approximate solution to a numerical problem that would be difficult to solve by other methods.[3] There are many implementations of MCM, but all of them follow these steps:

---

[1] Kroese, Dirk P. "Why the Monte Carlo Method Is so Important Today." *Wires Computational Statistics*, March 21, 2014, pp2-3. Accessed April 4, 2018.

[2] Iordanova, Tzveta. "Bet Smarter With the Monte Carlo Simulation." Investopedia. March 21, 2017. Accessed April 4, 2018. https://www.investopedia.com/articles/07/monte_carlo_intro.asp.

[3] Kroese, Dirk P. "Why the Monte Carlo Method Is so Important Today." *Wires Computational Statistics*, March 21, 2014, pp2-3. Accessed April 4, 2018.

1. Defining statistical properties of inputs[4]
2. Generate sets of random inputs under above properties
3. Perform deterministic (same input always leads to the same output) calculation with these sets

And the output of the MCM can vary from numerical values, probability distributions, or range of possible outcomes.

Monte Carlo Integration

The basic formulation of MCM is the evaluation of a definite integral. Although the implementation of the MCM used in my essay is generating draws from probability distribution, here I describe the MCM integration, because it most explicitly illustrates the usage and meaning of MCM.

To start with, let's say that we need to evaluate an integral of a function $y = h(x)$ with the limits $a, b \in \mathbb{R}$ so that

$$\boldsymbol{\theta = \int_a^b h(x)dx}$$

The idea of MCM integration lies on finding the area under curve of a function $y = h(x)$, which is θ, using random points on the graph. Let's say that $A = \{x_1, x_2, x_3, \dots\}$ is a set of random numbers in the interval $[a, b]$ that are inputted into $h(x)$. Consider the case when $x_1 \approx b$: the value of $h(x_1)\times(b-a)$ gives the rectangular area that overestimates the area under the curve. Then if $x_2 \approx a$, $h(x_2)\times(b-a)$ results in an area that underestimates the area under the curve. However, if we calculate the mean of two areas formed by x_1 and x_2 the value will be a better approximation of θ. So that is how the MCM integration works: we generate random points of a function inside the limits, then find the mean of the areas formed by the value of a function at each point multiplied by the limit to find the area under the curve, which is the value of a definite integral. Also the important assumption is that the bigger the size of the sample of random variables is, more accurate will be the approximation of the area under the curve. In fact, this assumption is proved by the law of large numbers that is explained further in the essay. Thus the formula of the MCM integration is[5]

$$\boldsymbol{\widehat{\theta} = (b-a)\frac{1}{N}\sum_{i=1}^{N} h(A_i)}$$

where $\hat{\theta}$ is the estimate of θ, N is a size of a A, and A is sample of random numbers.

Having derived the formula, I will prove that the expected value of $\hat{\theta}$ is equal to θ using the law of unconscious statistician, which states that the expected value of the function $f(R)$ with the set of random variable R is

$$\boldsymbol{E\big(f(R)\big) = \int f(R)P(R)dA}$$

where $P(R)$ is a probability density function (PDF) of the random variable.

---

[4] Paltani, Stephane. "Monte Carlo Methods." PhD diss., UNIG, 2010. Abstract. May 10, 2010. Accessed April 25, 2018. https://www.unige.ch/sciences/astro/files/2713/8971/4086/3_Paltani_MonteCarlo.pdf.

[5] "Mathematics and Physics for Computer Graphics." Scratchapixel. December 4, 2009. Accessed May 2, 2018. https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/geometry/points-vectors-and-normals.

Coming back to the problem the expected value of $\hat{\theta}$ is

$$\boldsymbol{E[\widehat{\theta}] = E[(b-a)\frac{1}{N}\sum_{i=1}^{N} h(x_i)]}$$

where N is the amount of elements in the set A. This equation can be expressed as

$$\boldsymbol{E[\widehat{\theta}] = (b-a)\frac{1}{N}E[h(x)]}$$

Applying the law of unconscious statistician yields

$$\boldsymbol{E\left[\widehat{\theta}\right] = (b-a)\frac{1}{N}E[h(x)] = (b-a)\frac{1}{N}\sum_{i=1}^{N}\int_{a}^{b} h(x)P(x)dx}$$

knowing that the distribution of the random variable is uniform in the interval $[a, b]$

$$\boldsymbol{P(A) = \frac{1}{(b-a)}}$$

$$\boldsymbol{(b-a)\frac{1}{N}\sum_{i=1}^{N}\int_{a}^{b} h(x)P(x)dx = \frac{(b-a)}{(b-a)}\frac{1}{N}\sum_{i=1}^{N}\int_{a}^{b} h(x)dx =}$$

$$\boldsymbol{= \frac{1}{N}\sum_{i=1}^{N}\int_{a}^{b} h(x)dx = \int_{a}^{b} h(x)dx = \theta}$$

The expected value of $\hat{\theta}$ is indeed equal to the θ. We can conclude that MCM actually can be used to evaluate definite integrals.

Random Numbers

Generation of random numbers is a crucial technique used in MCM. In this section I define random numbers (RN), explain why pseudorandom numbers (PRN) are preferred over truly random numbers, investigate Linear Congruential Generator, and determine the appropriate size of sample of PRN needed for the MCM.

What are random numbers?

In mathematical term, "random numbers is the set of numbers that meet two requirements – the values are uniformly distributed over a defined set, and it is impossible to predict consequent values based on past or present ones"[6]. A uniform distribution means that it has a constant probability for any element so the PDF would be

$$\boldsymbol{P(x) = \begin{cases} \frac{1}{b-a} \, for \, a \leq x \leq b \\ 0 \, for \, x > b, x < a \end{cases}}$$

[6] Rouse, Margaret. "Random Numbers." Techtarget. September 5, 2005. Accessed May 3, 2018. https://whatis.techtarget.com/definition/random-numbers.

This means that any number x of the set of random numbers in the interval $[a, b]$ has an equal probability to emerge in a set.

Probability and predictability

Even though terms probability and predictability seem equivalent in meaning, in fact, they are quite different. A set of numbers may have the uniform distribution probability but may be quite predictable. Consider two sets:

$$\boldsymbol{A} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}\}$$
$$\boldsymbol{B} = \{\mathbf{1}, \mathbf{7}, \mathbf{5}, \mathbf{3}, \mathbf{4}, \mathbf{2}, \mathbf{6}\}$$

While both of the sets A and B have the uniform distribution as the probability of each element in the interval [1,7] is $P(x) = \frac{1}{7}$, the set A doesn't look random, because there is a pattern or general rule from which we can deduce other elements basing on previous ones. In fact, it is possible to predict next elements in the B by the formula $a_{n+1} = a_n + 1$, where a_n is the previous element of the set. Contrary to the set A, the set B looks more random, because there is no evident way to estimate next elements basing on previous ones. Thus the predictability and probability are distinct concepts of RN.

Pseudorandom numbers

Pseudorandom numbers are samples that seem to be randomly drawn from some known distribution.[7] Unlike RN, PRN numbers are predictable for producer (values can be predicted from previous ones) and even not necessarily uniformly distributed (biased PRN). So it is rather paradoxical to call PRN random, but still without the knowledge about the method used to generate PRN the numbers seem indeed random.

While the RN find their application in many fields, such as scientific experiments, computer programs, cryptography, people rarely use truly random numbers for a few reasons. First of all, it is very inefficient and inconvenient to generate truly random numbers, because it needs external factors that can provide entropy and chaos as computers or machines can't offer randomness by themselves (consequent elements can be always predicted based on previous ones). For example, it is possible to use a radiation detector, dices, or waves analyser to produce truly random values[8], but it is evident that these installations aren't affordable for many situations. Secondly, random numbers are mainly employed in analytical methods such as the MCM, which should be reproducible or deterministic[9]. The deterministic means that the method must be able to be repeated and output the same results under the same inputs. So true random number generators can't be used in MCM as they yield different unpredictable values every time, which means the same input can't be made. That is the reason why not truly random numbers but pseudorandom numbers, which can be always reproduced, are used in the MCM.

---

[7] Gentle, James E. *Random Numbers and Monte Carlo Simulation*. pp 173-174. George Mason University, 2002.

[8] Stepehens, Rod. *Algorithms: A practical approach to computer algorithms.* pp 34-35. Wiley, 2013.

[9] Kroese, Dirk P. "Why the Monte Carlo Method Is so Important Today." *Wires Computational Statistics*, March 21, 2014, pp2-3. Accessed April 4, 2018.

Linear Congruential Generator

While there are many pseudorandom number generators (PRNG) in existence, one of the most famous one is the Linear Congruential Generator (LCG) that was introduced by Lehmer, a famous mathematician, in 1951.[10] Although LCG is not widely used nowadays because of security issues (random numbers are primarily used in cryptography), I employ it in my essay, because my MCM doesn't require security and LCG is very simple and efficient in comparison with other methods. The formula of the LCG is following

$$\boldsymbol{x_{n+1} = (ax_n + c)\ mod\ m}$$

where x_{n+1} is a next random number in the sequence x, x_n - previous random number, a - "multiplier", c – "increment", m – "modulus". [11] The symbol mod defines modulo operation, which finds the remainder of the division. For example, $6\ mod\ 4 = 2$, because dividing 6 by 4 leaves the remainder 2. To generate random numbers we need to define a, c, m, and x_0 - the initial number of the sequence or the seed value. However, it is important to remember the range of possible values for these constants

$$\boldsymbol{m > 0, m \in \mathbb{Z}}$$
$$\boldsymbol{0 < a < m, a \in \mathbb{Z}}$$
$$\boldsymbol{0 \leq c < m, c \in \mathbb{Z}}$$
$$\boldsymbol{0 \leq x_0 < m, x_0 \in \mathbb{Z}}$$

In fact some of the restrictions above are not really strict, for instance, even if $c > m$ it won't affect the coherence of the equation: it would still produce PRN. But it is preferable to set the values under these constraints for the clarity of the expression.

The iteration of the equation will produce the sequence of PRN, but all the numbers will be integers located in the interval $[0, m)$, because the minimum possible remainder of the division is 0, while the maximum possible remainder is $m - 1$.

Generating PRN using LCG

Having learnt the formula of the LCG, now I can generate PRN using it. However, to start with, I need to define the constant values of the equation and a seed value. As mentioned in the previous section, the output of the LCG lies in the interval $[0, m)$, so it is logical to choose the value of m very large, because otherwise the same values will repeat over and over in the sequence. For example, If $m = 2$, then the possible outcomes are only 0, 1. The "multiplier" and "increment" are also a very important constant that with "modulus" determine the randomness (predictability) of the output. In fact I can choose any spontaneous numbers under defined restrictions for constant values, but it is uncertain whether they will yield the sequence of PRN suitable for the MCM that demands high randomness of the input. For this reason, I decided to use the constants that are widely used in practice, passed randomness-evaluating criteria and have been already approved in statistical tests. Thus the LCG equation that I use in my essay is[12]

---

[10] Knuth, Donald E. *The Art Of Computer Programming.* pp 20-21. Addison-Wesley, 1997.

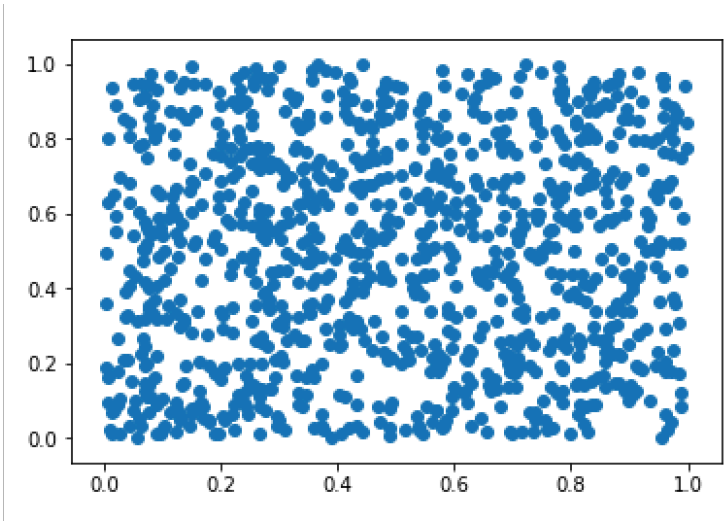
[11] Gentle, James E. *Random Numbers and Monte Carlo Simulation*. pp 175-177. George Mason University, 2002.

[12] William H. Press, Saul A. Teukolsky. *Numerical Recipes. pp 56-57.* Cambridge University Press, 2007.

$$x_{n+1} = (1664525x_n + 1013904223)\ mod\ 2^{32}$$

After defining the constants, I generated 2000 random numbers with the seed value 35 and

Diagram 1. LCG random numbers



converted them into the values in the interval of the uniform distribution $U(0,1)$ by dividing them by $m = 2^{32}$. Then I paired first 1000 numbers with next 1000 numbers to represent them in the Cartesian coordinate system. I used computer programs coded by myself (Appendix A). The result can be seen in the diagram.

The diagram illustrates that the random values are indeed distributed uniformly and quite unpredictable as there is no easily identifiable pattern. Thus pseudorandom numbers used further in the essay are generated by LCG with defined earlier constants.

Law of large numbers: amount of PRN needed in the MCM

After identifying the generator the question rises, "How many PRN are needed for the MCM?" In fact, the MCM can be done with a vary small sample of PRN, but it rarely leads to the meaningful output, because according to the law of large numbers "As the number of trials of a random process increases, the percentage difference between the expected and actual values goes to zero"[13]. This means that if the sample of PRN is too small then the variance and standard deviation of the output will be too high to deduce some meaningful numerical values.

Let's say we have a unfair coin the probability of which can be described with Bernoulli distribution:

$$P(x) = \begin{cases} 0.6\ \ for\ x = 1 \\ 0.4\ for\ x = 0 \\ 0\ for\ x \neq 1, x \neq 0 \end{cases}$$

[13] Weisstein, Eric W., and John Renze. 2010. http://mathworld.wolfram.com/LawofLargeNumbers.html (accessed 7 28, 2018).

where $x = 0$ identifies a tail, and $x = 1$ identifies a head. And the rule of the game is following – the player starts the game with the balance of 0$; the player throws the coin, and if it shows the head he earns 1$, otherwise he loses 1$.

Let's simulate the outcome of the game using the MCM. Taking $g(Y)$, where Y is a random variable drawn from the $U(0,1)$, as a function describing a revenue after each throw the game can be modelled by:

$$\boldsymbol{g(Y = y) = \begin{cases} 1\ for\ 0 \le y < 0.6 \\ -1\ for\ 0.6 \le y < 1 \end{cases}}$$

The total revenue after all throws is

$$\sum_{i=1}^{n} \boldsymbol{g(Y_i)}$$

where n is the total amount of throws made.

Now, let's make Monte Carlo Simulation with $n = 100$ 5 times with different PRN. I implemented the program stated in the Appendix B

Table 1. Total revenues with 100 PRN

Trial	n	The total revenue
1	100	-6
2	100	38
3	100	12
4	100	20
5	100	30

Calculating the mean, variance, and standard deviation of the output correct to 1 decimal place gives

$$\boldsymbol{Mean: \bar{x} = \frac{\sum x}{n}}$$

$$\boldsymbol{\bar{x} = 18.8}$$

$$\boldsymbol{Variance: \sigma^2 = \frac{\sum (x - \bar{x})^2}{n}}$$

$$\boldsymbol{\sigma^2 \approx 231.4}$$

$$\boldsymbol{Standrad\ deviation: \sigma \approx 15.2}$$

The standard deviation is as large as $\frac{15.21}{18.8} \times 100\% \approx 81\%$ of the range. This makes the data sample from the MCM inapplicable to any further analysis, because the values are too widespread to detect any trend.

However, let's take the $n = 100000$ and conduct MCS with different sets of PRN.

Table 2. Total revenues with 100000 PRN

Trial	n	The total revenue
1	100000	19958
2	100000	19420
3	100000	20506
4	100000	20028
5	100000	20160

Calculating the mean, variance, and standard deviation of the output correct to nearest integer gives

$$\boldsymbol{Mean}: \boldsymbol{\overline{x} = 20014}$$
$$\boldsymbol{Variance}: \boldsymbol{\sigma^2 \approx 123904}$$
$$\boldsymbol{Standrad\ deviation}: \boldsymbol{\sigma \approx 352}$$

The standard deviation is only $\frac{352}{20014.4} \times 100\% \approx 1.8\%$ of the mean value. The low standard deviation means that there is a certain trend in the behaviour of the group, which can be studied further. Evidently the second sample of data is of more interest to statisticians than the previous one.

So from this we can draw a conclusion that in my application of the MCM it is preferable to use a large sample of PRN counting as much as 100000.

Probability model of the La Liga

In this section of the essay, I construct the probability model of the La Liga needed for the MCM. In the process I use the data of the La Liga matches from the previous season 2016/2017.

La Liga

To start with, La Liga is a football tournament, where 20 Spanish football teams compete with each other. La Liga uses double round-robin format, which means that each team plays with other clubs twice. Using the permutation equation, we can calculate the total amount of matches

$$\boldsymbol{nPk} = \frac{\boldsymbol{n!}}{\boldsymbol{(n-k)!}}$$
$$\boldsymbol{38P2 = 380}$$

A win in any match is awarded with 3 points, draw – with 1 point, loss – 0 point.

Expected goal model

It is evident that in reality a result of a match and tournament is largely dependent on many different factors - players' condition, rest time, players' relative powers and hundreds of other aspects. However, the data about these things are not commonly available. So in my model I will consider only two most relevant aspects – relative goal-scoring power of a team and whether a match is held at home or away – to assess the probability of each team winning or drawing against other. The data used in the essay is taken from this source.[14] The sample of the data is given in the Appendix C

Power indexes

First of all we need to find the La Liga teams' relative attacking (AP) and defensive (DP) power indexes, which can be understood as a measurement of average goals scored in a match by a concrete team $\boldsymbol{\overline{a}}$ divided by average goals scored in a match by all teams $\boldsymbol{\overline{b}}$ and average goals

[14] Football-data. "Historical Data." *Football-data.* 2018. http://www.football-data.co.uk/ (accessed 7 18, 2019).

conceded in a match by a concrete team $\overline{\boldsymbol{c}}$ divided by average goals conceded in a match by all teams $\overline{\boldsymbol{d}}$.[15] The formula for AP and DP would be

$$\boldsymbol{AP} = \frac{\overline{\boldsymbol{a}}}{\overline{\boldsymbol{b}}}, \boldsymbol{DP} = \frac{\overline{\boldsymbol{c}}}{\overline{\boldsymbol{d}}}$$

However, as said before, I also consider the factor of whether the game is played at home or away. This means that teams have two types of AP and DP: AP at home (APH), AP away (APA), DP at home (DPH), and DP away (DPA):

$$\boldsymbol{APH} = \frac{\overline{\boldsymbol{a_h}}}{\overline{\boldsymbol{b_h}}}, \boldsymbol{APA} = \frac{\overline{\boldsymbol{a_w}}}{\overline{\boldsymbol{b_w}}}, \boldsymbol{DPH} = \frac{\overline{\boldsymbol{c_h}}}{\overline{\boldsymbol{b_w}}}, \boldsymbol{DPA} = \frac{\overline{\boldsymbol{c_w}}}{\overline{\boldsymbol{b_h}}}$$

Let's calculate these parameters for the football club Barcelona. First, the formula for the mean is

$$\overline{\boldsymbol{x}} = \frac{\sum \boldsymbol{x}}{\boldsymbol{n}}$$

Using it and data of the La Liga 2016/2017, I calculated mean goals scored by a home team. The calculations are made correct to 9 decimal places

$$\overline{\boldsymbol{b_h}} = \mathbf{1.663157895}$$

The mean goals scored by an away team is

$$\overline{\boldsymbol{b_w}} = \mathbf{1.278947368}$$

The mean values for Barcelona are

$$\overline{\boldsymbol{a_h}} = \mathbf{3.368421052}$$
$$\overline{\boldsymbol{a_w}} = \mathbf{2.736842105}$$
$$\overline{\boldsymbol{c_h}} = \mathbf{0.894736842}$$
$$\overline{\boldsymbol{c_w}} = \mathbf{1.052631578}$$

So the AP and DP for Barcelona correct to 3 decimal places would be

$$\boldsymbol{APH} = \frac{\overline{\boldsymbol{a_h}}}{\overline{\boldsymbol{b_h}}} = \frac{\mathbf{3.368421052}}{\mathbf{1.663157895}} \approx \mathbf{2.025}$$
$$\boldsymbol{APA} = \frac{\overline{\boldsymbol{a_w}}}{\overline{\boldsymbol{b_w}}} = \frac{\mathbf{2.736842105}}{\mathbf{1.278947368}} \approx \mathbf{2.140}$$
$$\boldsymbol{DPH} = \frac{\overline{\boldsymbol{c_h}}}{\overline{\boldsymbol{b_w}}} = \frac{\mathbf{0.894736842}}{\mathbf{1.278947368}} \approx \mathbf{0.700}$$
$$\boldsymbol{DPA} = \frac{\overline{\boldsymbol{c_w}}}{\overline{\boldsymbol{b_h}}} = \frac{\mathbf{1.052631578}}{\mathbf{1.663157895}} \approx \mathbf{0.633}$$

These numbers mean that Barcelona scores at home and away twice often than an average team, while concedes at home and away less often than an average team.
Calculating the power indexes for all 20 teams yielded the table below:

[15] Freymiller, Adam. *Monte Carlo Simulation Football.* 6 8 2017. https://medium.com/@adamfreymiller/a-monte-carlo-simulation-of-the-2017-18-premier-league-season-3b7bbe8b8a13 (accessed 7 18, 2019).

Table 3. Power indexes of La Liga teams

Team	APH	APA	DPH	DPA
Barcelona	2.025	2.140	0.700	0.633
Real Madrid	1.519	2.387	0.823	0.665
Sevilla	1.234	1.235	0.658	1.044
Atletico Madrid	1.266	1.235	0.576	0.411
Athletic Bilbao	1.139	0.700	0.741	0.791
Villarreal	1.108	0.864	0.741	0.475
Real Sociedad	0.949	1.193	0.988	0.918
Eibar	0.918	1.111	0.864	0.949
Las Palmas	1.044	0.823	1.029	1.551
Malaga	1.013	0.700	1.029	0.949
Espanyol	0.886	0.864	0.988	0.823
Celta Vigo	0.601	0.905	0.864	0.696
Deportivo Alaves	0.949	0.947	1.317	1.171
Valencia	1.013	0.988	1.317	1.044
Deportivo La Coruna	0.854	0.658	0.947	1.203
Real Betis	0.696	0.782	0.988	1.266
Leganes	0.696	0.576	0.947	1.013
Sporting Gijon	0.823	0.658	1.564	1.076
Granada	0.538	0.535	1.317	1.582
Osasuna	0.728	0.700	1.605	1.741

Expected Goals Scored

Using the power indexes it is possible to calculate expected goals (ExpG) scored by teams in any match. ExpG of a home team is calculated with the formula[16]

$$ExpG_h = \frac{APH \times DPA}{\overline{b_h}}$$

where APH is the attacking power index of a home team and DPA – defensive power index of an away team. ExpG of an away team is

$$ExpG_w = \frac{APW \times DPH}{\overline{b_w}}$$

where APW is the attacking power index of an away team and DPH – defensive power index of a home team.

For example, if Barcelona plays against Real Madrid at home, its ExpG would be

$$ExpG_h = \frac{2.025 \times 0.665}{1.663} = 0.810$$

Real Madrid's ExpG would be

[16] Freymiller, Adam. *Monte Carlo Simulation Football.* 6 8 2017. https://medium.com/@adamfreymiller/a-monte-carlo-simulation-of-the-2017-18-premier-league-season-3b7bbe8b8a13 (accessed 7 18, 2019).

$$ExpG_w = \frac{2.387 \times 0.700}{1.279} = 1.306$$

Probability of final scores: Poisson distribution

To find the probability of each outcome of a match I will use the Poisson distribution to calculate the probability of all possible score lines. I chose Poisson distribution, because the expected goal measure can be understood as the number of goals to be scored in a period of a time (one match), which is basically the rate of goals scored per match. Knowing the rate of the event I can easily implement the Poisson distribution. The Poisson distribution's probability mass function is defined as[17]

$$P(X = x) = \frac{e^{-m} m^x}{x!}$$

where m is the rate of success – in our case rate of goals scored per match - and x is the number of events occurred – in our case number of goals scored in a match. The range of possible values of the variable x is an interval $[0, 7]$ as 7 is the maximum amount of goals scored in one match in the La Liga season 2016/2017. Applying the Poisson distribution again to the match Barcelona and Real Madrid results in following equations:

$$P(X = x) = \frac{e^{-0.810} \times 0.810^x}{x!} \; for\ Barcelona$$

$$P(Y = y) = \frac{e^{-1.306} \times 1.306^y}{y!} \; for\ Real\ Madrid$$

calculating for each values of x gives

Table 4. Poisson distribution of goals scored in match Barcelona-Real Madrid

Team\Goals(x and y)	0	1	2	3	4	5	6	7
Barcelona	0.445	0.360	0.146	0.039	0.008	0.001	<0.001	<0.001
Real Madrid	0.271	0.354	0.231	0.101	0.033	0.009	0.002	<0.001

Each team has 8 possible events so the amount of all possible outcomes is

$$8P2 + 8 = 64$$

The probability of each outcome is calculated by the formula of the compound independent events

$$P(A \cap B) = P(A) * P(B)$$

For example, the probability that the match Barcelona and Real Madrid ends with the score 0-0 is

$$P(0 \cap 0) = P_{Barcelona}(X = 0) * P_{RM}(Y = 0) = 0.445 * 0.271 \approx 0.121$$

Analogically we can calculate the probability of all the 64 possible score lines, which can be divided into three groups: Barcelona's win, draw, and Real Madrid's win.

Table 5. Probability of each score lines in match Barcelona-Real Madrid

Score	Probability	Outcome	Outcome probability
1-0	0.098	Barcelona wins	0.23
2-0	0.040		

[17] Paul Fannon, Vesna Kadelburg. *Mathematics HIgher Level for the IB Diploma. pp 670-673* Cambridge University Press, 2012.

2-1	0.052		
…(other scores when Barcelona wins)	…		
0-0	0.121	Draw	0.29
1-1	0.127		
2-2	0.034		
…(other scores when draw)	…		
0-1	0.158	Real Madrid wins	0.48
0-2	0.103		
0-3	0.045		
…(other scores when Real Madrid wins)	…		

The sums of probabilities of outcomes under each section show the probability of one of three possible results (win, draw, or loss). So the PDF for the outcome of the match Barcelona – Real Madrid is

$$\boldsymbol{P_{Barcelona-RM}(x) = \begin{cases} 0.23 \ \ for\ x = 0 \\ 0.29\ for\ x = 1 \\ 0.48\ for\ x = 2 \end{cases}}$$

where $x = 0$ is Barcelona's win, $x = 1$ is draw, and $x = 2$ is Real Madrid's win.

Monte Carlo Simulation of matches

Having defined the probability distribution for the outcome of a match, we can perform the MCS on it to predict the actual outcome of a match. First, from the probability distribution we can derive functions $y = h(x)$ and $y = g(x)$ that show the points earned by a home team and an away team respectively. In the case of Barcelona – Real Madrid

$$\boldsymbol{h(x) = \begin{cases} 3 \ \ for\ 0 < x \leq 0.23 \\ 1\ for\ 0.23 < x \leq 0.52 \\ 0\ for\ 0.52 < x < 1 \end{cases}}$$

$$\boldsymbol{g(x) = \begin{cases} 0 \ \ for\ 0 \leq x \leq 0.23 \\ 1\ for\ 0.23 < x \leq 0.52 \\ 3\ for\ 0.52 < x < 1 \end{cases}}$$

$$\boldsymbol{0 < x \leq 0.23, 0.23 - 0 = 0.23 = P_{Barcelona-RM}(0)}$$
$$\boldsymbol{0.23 < x \leq 0.52, 0.52 - 0.23 = 0.29 = P_{Barcelona-RM}(1)}$$
$$\boldsymbol{0.52 < x \leq 1, 1 - 0.52 = 0.48 = P_{Barcelona-RM}(2)}$$

Conducting the Monte Carlo Simulation with the set of 100000 random numbers X in $U(0,1)$ and finding the average yields the predicted points earned by the teams:

$$\boldsymbol{predicted\ points\ earned\ by\ Barcelona = \frac{\sum_{i=1}^{n=100000} h(x_i)}{100000}}$$

$$\textbf{predicted points earned by Real Madrid} = \frac{\sum_{i=1}^{n=100000} g(x_i)}{100000}$$

Resultant table

Implementing the program in the Appendix D and calculating the predicted points earned in all 380 matches yields the table below

Table 6. Predicted standing of the La Liga teams in the season 2017/2018

Standings	Team	Points
1	Barcelona	79.8
2	Real Madrid	75.2
3	Atletico Madrid	68.1
4	Sevilla	59.7
5	Villarreal	58.3
6	Real Sociedad	53.2
7	Eibar	52.8
8	Athletic Bilbao	52.4
9	Espanyol	48.8
10	Celta Vigo	48.5
11	Valencia	46.6
12	Malaga	46.0
13	Deportivo Alaves	44.1
14	Las Palmas	42.9
15	Deportivo La Coruna	42.0
16	Real Betis	40.9
17	Leganes	40.3
18	Sporting Gijon	37.1
19	Osasuna	31.1
20	Granda	29.7

Conclusion

Coming back to the research question "How effective is MCM for predicting the football league outcome?", I will examine the accuracy of predictions and evaluate the limitations of the experiment.

Errors of prediction

Before comparing the prediction with the actual result I introduce three measurements of errors of predictions to quantify the effectiveness of my prediction.

The first one D measures the mean difference between predicted results and actual. If A is the set of actual results, and P – predicted results, the formula of D would be

$$\boldsymbol{D} = \frac{\sum_{i=1}^{n} |\boldsymbol{A_i} - \boldsymbol{B_i}|}{\boldsymbol{n}}$$

The next one S measures the mean deviation of the difference between predicted results and actual:

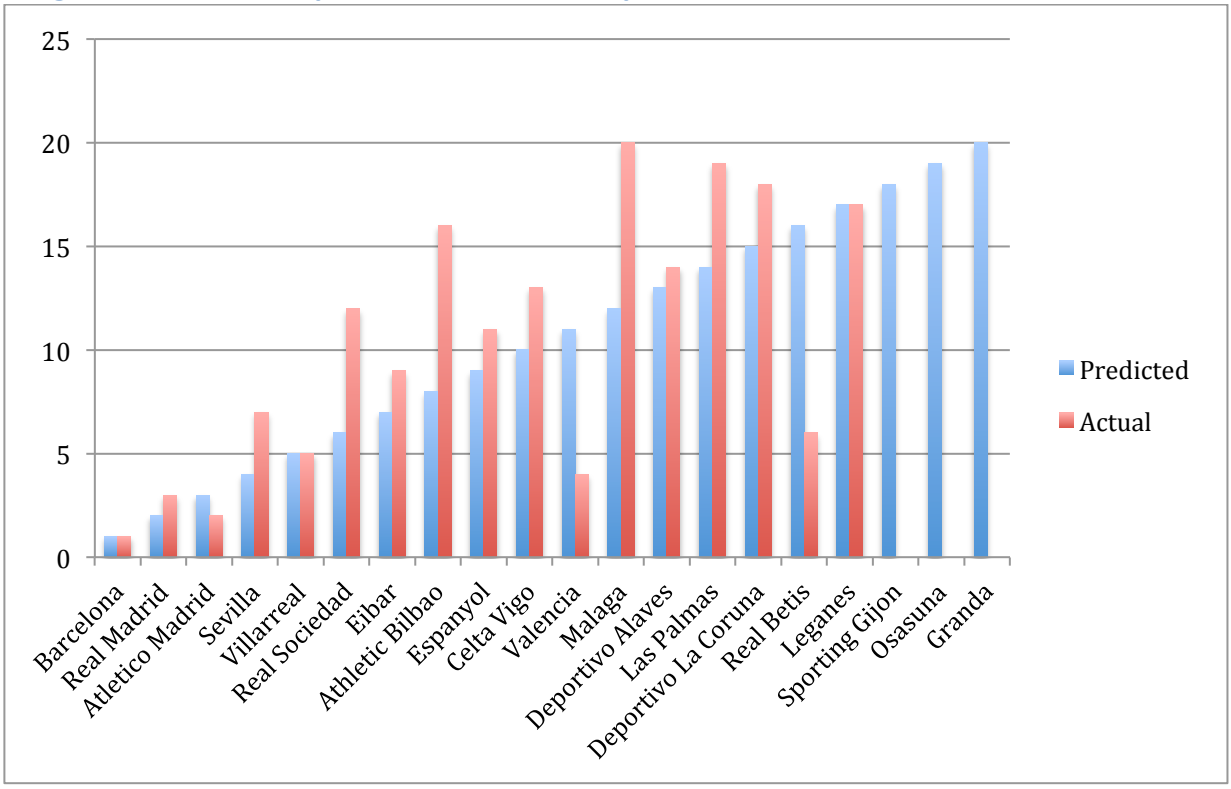
$$S = \frac{\sum_{i=1}^{n} ||A_i - B_i| - D|}{n}$$

The third one P measures the percent deviation:

$$P = \frac{S}{D}$$

Comparison of the predicted standings with the actual standings of the La Liga 2017/2018[18]

Diagram 2. Predicted placements vs actual placements



(Last 3 teams don't have actual results, because according to the rule of the La Liga, they are relegated, and other teams replace them[19])

The diagram shows that the prediction for some teams (Barcelona, Villarreal, Leganes) is accurate, but for other ones the difference with actual standings is quite big (Atheletic Bilbao, Malaga, etc).

Calculating the error measurements to 2 correct decimal places gives:

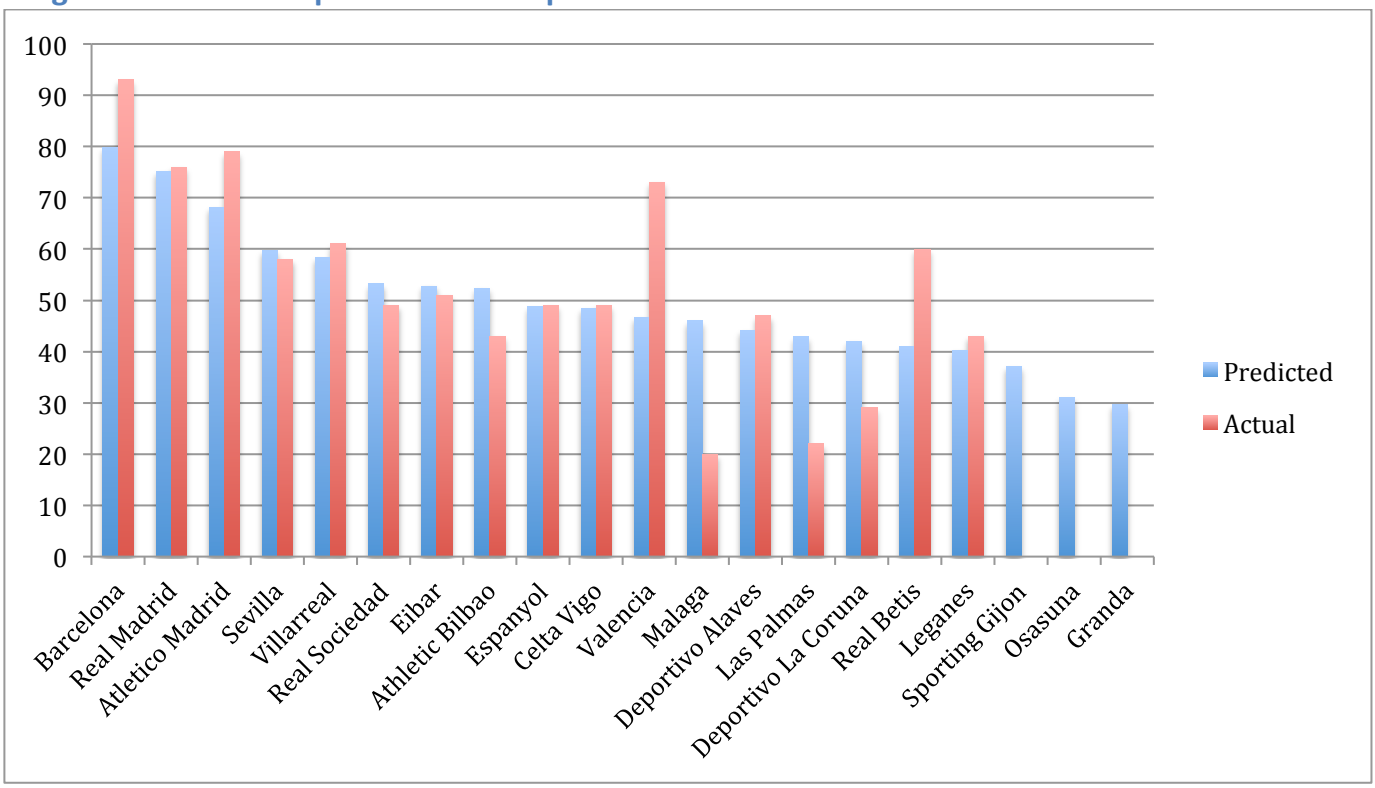
$$D = 3.53$$
$$S = 2.69$$
$$P = 76.20\%$$

[18] whoscored.com. *La Liga Table 2017/2018.* 2018. https://www.whoscored.com/Regions/206/Tournaments/4/Seasons/6960/Spain-La-Liga (accessed 7 23, 2018).

[19] whoscored.com. *La Liga Table 2017/2018.* 2018. https://www.whoscored.com/Regions/206/Tournaments/4/Seasons/6960/Spain-La-Liga (accessed 7 23, 2018).

Comparison of the predicted points with actual points

Diagram 3. Predicted points vs actual points



Calculating the error measurements to 2 correct decimal places gives:

$$\boldsymbol{D = 9.20}$$
$$\boldsymbol{S = 7.68}$$
$$\boldsymbol{P = 83.48\%}$$

Assessing the effectiveness of MCM prediction

The errors of predictions show that my predictions were quite accurate. In the case of placements of teams, the mean difference between the actual and predicted results is 3.53, which is

$$\frac{\boldsymbol{3.53}}{\boldsymbol{19}} = \boldsymbol{18.6\%}$$

percent of the maximum difference. However, the percent deviation 76.2% is quite high, actually more than 50%, which means that data is widespread. This indicates that there is no general consistency between predictions of each team. In the case of points earned by teams, predictions were also quite accurate, but the same problem of large deviation 83.48% can be observed. Overall my prediction was effective rather than not as it definitely shows some correspondence with actual results. The high deviation might be explained by some exceptional cases when underperforming teams suddenly improve to a very large extent (e.g. Real Betis) that are quite hard to predict by statistical prediction.

Limitations of the reasearch

While any prediction is not meant to be absolutely accurate, I should acknowledge some limitations of my probability model that possibly lead to less accurate predictions. First of all my model did not account some important factors that could affect the rate of goals scored in a match

- Players leaving the team. Some of the players left their teams during the summer, so these teams lost competitiveness and relative power.
- Structural changes. Some teams suffered not only loss of players but also bigger changes inside the administration. These teams are likely to perform worse than others.
- Injuries. Many players are injured during the season, which is a crucial factor affecting the outcome of a match.

Not only that, there were limitations in my method of data collection

- I used data only from the previous season of the La Liga. The bigger the sample of data more accurate is the probability model, according to the law of large numbers, so using the data from other seasons could strengthen my model.
- I used data only from the La Liga, while teams play other tournaments as well. Including the results from other tournaments to assess the relative power of teams could produce more accurate probability model.

And most importantly my model assumes that the goals scored in a match are independent events, which is, evidently untrue in reality. To construct a better model I need to define the dependency of between goals scored in a single match.

Further investigation

In my essay I explored the MCM to predict the outcome of the football league. Although there were some major limitations in my experiment, it is evident that not a single mathematical model can account all the factors that affect the probability of real-life events, because there would be too many variables. And the comparison with the actual outcome showed that my prediction were quite accurate.

However my investigation is far from finished and is open to further investigation as there are many questions left unanswered. First of all, in my essay I explored only a single method of generating PRN, which is recognised as not the best generator. In my opinion it would be valuable to research other PRNGs and compare the performance of the MCM with different PRNGs. Not only that it would be interesting to draw PRNG from not uniform but other distributions.

Furthermore, to assess MCM method of prediction I would like to compare it with other methods. So the investigation could be extended to other methods of prediction. In addition, the methods can be applied to other events than football tournament.

Finally constructing more complex probability model would strengthen the investigation significantly. I could take into account more then two variables (goals scored and home-away factor) such as player ratings or values. Also it would be better to use more data from previous seasons.

Hence, I would conclude that the essay left further research opportunities that I would gladly take.

Word count: 3886

Bibliography

1. Football-data. "Historical Data." *Football-data.* 2018. http://www.football-data.co.uk/ (accessed 7 18, 2018).
2. Freymiller, Adam. *Monte Carlo Simulation Football.* 6 8 2017. https://medium.com/@adamfreymiller/a-monte-carlo-simulation-of-the-2017-18-premier-league-season-3b7bbe8b8a13 (accessed 7 18, 2018).
3. Gentle, Donal E. *Random Numbers and Monte Carlo Simulation.* George Mason University, 2002.
4. Iordanova, Tzveta. *Bet Smarter With the Monte Carlo Simulation.* 2017. https://www.investopedia.com/articles/07/monte_carlo_intro.asp (accessed 4 4, 2018).
5. Knuth, Donald E. *The Art Of Computer Programming.* Addison-Wesley, 1997.
6. Kroese, Dirk P. "Why the Monte Carlo method is so important today." *Wires Computational Statistics*, 2014.
7. Paltani, Stephane. "Monte Carlo Methods." *UNIG.* 2010-11. https://www.unige.ch/sciences/astro/files/2713/8971/4086/3_Paltani_MonteCarlo.pdf (accessed 4 25, 2018).
8. Paul Fannon, Vesna Kadelburg. *Mathematics HIgher Level for the IB Diploma.* Cambridge University Press, 2012.
9. Rouse, Margaret. *Random Numbers.* 5 9 2005. https://whatis.techtarget.com/definition/random-numbers (accessed 05 3, 2018).
10. Scratchapixel. *Mathematics and Physics for Computer Graphics.* 2009. (accessed 6 22, 2018).
11. Sheldon, Ross. *Introduction to Probability Models.* 2010.
12. Stepehens, Rod. *Algorithms: A practical approach to computer algorithms.* Wiley, 2013.
13. Weisstein, Eric W., and John Renze. 2010. http://mathworld.wolfram.com/LawofLargeNumbers.html (accessed 7 28, 2018).
14. whoscored.com. *La Liga Table 2017/2018.* 2018. https://www.whoscored.com/Regions/206/Tournaments/4/Seasons/6960/Spain-La-Liga (accessed 7 23, 2018).
15. William H. Press, Saul A. Teukolsky. *Numerical Recipes.* Cambridge University Press, 2007.