MASTER IN COMPUTER SCIENCE

Quand les tirs au but sont injustes

Analysis of a fair mechanism for penalty shoot-out in soccer competitions

Nguyen, Guillaume

Award date:
2022

Awarding institution:
University of Namur

UNIVERSITÉ DE NAMUR

FACULTÉ D'INFORMATIQUE

Quand les tirs au but sont injustes.

Analysis of a fair mechanism for penalty shoot-out in soccer competitions.

Guillaume
Nguyen

RUE GRANDGAGNAGE, 21 ● B-5000 NAMUR(BELGIUM)

Contents

Chapter 1

Introduction

Analysing human behaviour and determining which are the main factors that have influence in the decision-making process is not an easy task. Psychologists have been trying to extrapolate information from experiments for ever. Furthermore, witnessing a real-life situation and trying to predict the behaviour of a human actor in an uncontrolled environment is particularly complex. However, Apesteguia and Palacios-Huerta [2] showed that soccer and its penalty shoot-out mechanism offered a peculiar situation with few thus observable factors in the scoring probability of a shot.

The experience that is analysed is the mechanism of kicking shots from the penalty mark at the end of a competitive soccer match which ended in a tie. It goes as follows: each team get a side of a coin and the referee tosses it. Before 2003 the winner of the toss was designated as first shooter but as from 2003 the winner of the toss can decide if they want to shoot first or second. Then, for each round a shooter from the first shooting team (further referred as "A") takes a shot followed by a shooter from the second team (referred as B). Before five rounds, the only mean to win the shoot-out is to score more shots than the other team or to score so much more shots than the other team that it would be impossible for them to come back to the score by the fifth round. After five rounds, every round is decisive meaning that if one team succeeds and the other does not, we get a winner and the shoot-out is stopped. If not, it goes on until we get a winner. Since 2019 a

shooter can't shoot twice before every player on the field has shot at least once (Appendix A). This mechanism will later be referred as "ABAB".

As this experiment was deemed valuable by many researchers, there were many debates on the initial findings of Apesteguia and Palacios-Huerta [2]. Indeed, there was a rumour that the team going first in the shoot-out had a winning advantage. This will later be referred as "the first shooter advantage". Thus, Apesteguia and Palacios-Huerta [2] analysed a data set on penalty shoot-outs in order to scientifically establish whether there was a first shooter advantage or not. The initial findings were quite close to the rumour as they found a 60.5% advantage for the first shooting team. However, not everybody agreed with these findings of a first shooter advantage. So, other researchers tried to deny or statistically reinforce this initial claim.

On top of those empirical and experimental analyses, there were also studies on the possibility of a fairer mechanism. Those researches based themselves on the existence of a first shooter advantage. Researchers went looking at other sports and how they tackled a tie situation during competitions such as tennis and its tie-break. More, they also came up with mechanisms of their own and tested those with the scoring probabilities determined by empirical researchers. Interestingly enough, they used static scoring probabilities instead of a Markov chain which allows a probability to evolve through an experiment depending on previous outcomes. This is the main focus point of this work to come up with an empirically computed Markov chain to compare the fairness of those different suggested mechanisms using a simulator.

To reach our goal, we created our own data set as the ones from previous researchers were not made publicly available. We also wanted it to be easily scalable. Thus, we built up a script to collect penalty shoot-out data automatically from the Internet using "Python" programming language. The criteria for our source choice were the following: consistent HTML code, large number of football matches, sufficient match data and trustfulness of the website.

After collecting the data, we carried out a general analysis to describe our set using "R" programming language. Next, we tried to reproduce experiments from previous researchers. Then, we determined the different factors

that had influence on the scoring probability of a shot based on our own data set using particular data mining techniques. Afterwards, we carried out multiple regressions analyses with different factors. A regression analysis allows us to estimate a relation between an observed variable or outcome (the dependent variable) and influencing factors (one or more independent variables). We were able to pin point the ones with the most statistical significance when predicting the outcome of a shot.

After choosing two prediction models, we created various simulations to compare the different penalty shoot-out mechanisms with each other. The purpose is to see if the mechanism that was deemed the fairest in the literature is also the fairest based on our own empirical and experimental analysis.

Chapter 2

Literature review

In this chapter, we will review the different papers and other works that have been carried out by researchers on the fairness of the current penalty shoot-out mechanism in soccer. This will allow us to get a deeper understanding of this issue and find relevant information for our own research. We will start by focusing on studies which gathered and analysed empirical soccer competition data. Then we'll go deeper in the statistical analyses performed on the issue. There will be a summary of all the researches at the end of that section. Finally we will have a look at researches that suggested different shoot-out mechanisms to provide fairer winning rates. In general we will have:

- The null hypothesis H_0 = "Both team have a 50% chance of winning the game when reaching the penalty shoot-out after a tie match in a soccer competition".
- The alternative hypothesis H_a = "The first shooting team has a significantly higher winning rate than the second shooting one".

2.1 Initial and empirical analyses

The first paper was written by Apesteguia and Palacios-Huerta in 2008 [2] and was meant to analyse the psychological pressure in competitive environments. It will act as a chronological and analytical base as previous researches

will not be referred to here. Indeed, this paper is considered to be the starting point of the specific discussion about the penalty shoot-out mechanism and its fairness in high level soccer competitions.

In their paper Apesteguia and Palacios-Huerta [2] reviewed the impact of emotions on performance and socioeconomic outcomes. To support their theory, they decided to use soccer competitions as an example of two teams competing against each other. Furthermore, the sequential order inside each round is based on a random coin toss. The purpose of the research was to determine if the random outcome of the coin toss would significantly impact the outcome of an encounter. The psychological factors used in their research were based on multiple behavioural economics studies which incorporates psychological motives in economic models. Their main critic of such researches was the use of laboratory experiments. Indeed, the "generalizability" also known as "external validity" or the ability of an experiment results to apply on a bigger issue was debatable. Thus, the use of real-life data should be preferred to laboratory experiments. However, psychological principles at work are difficult to observe that is why they chose soccer penalty shoot-outs as an unusual clean opportunity for their study.

The real-life experiment is based on professionals performing simple tasks in a soccer tournament competition. There were many reasons why this mechanism represented a clean opportunity:

- Shooting a ball once requires no effort.
- Possible outcomes are binary (score or no score).
- All players are in the same position and location when taking the shots.
- Assuming the number of supporters are equals, the audience is the same for both teams.

This is the perfect opportunity to understand if the randomized sequence of the shoot-out has a different impact on the psychological pressure for each team. The importance of their study is based on previous psychological researches showing that the human nature could deem unfair the result of a perfectly randomized experience after it had happened (e.g., after losing.).

As introduced before, the experiment is as follows:

- The referee tosses a coin and the winning team takes the first shot (before 2003) or choses which team takes the first shot (as from 2003).
- Each shot involves a shooter and a goalkeeper with the typical time before the signal of the referee and the kick being 0.3-0.4 seconds. This is less time than required by the goalkeeper to determine the course of the ball and moving there to intercept it.
- The shot is either scored or not.

They used data from "the Union of European Football Associations (UEFA), the Rec.Sport.Soccer Statistics Foundation, the Association of Football Statisticians, and the Spanish newspaper MARCA. The dataset comprises 269 penalty shoot-outs with 2,820 penalty kicks over the period 1970-2008" [2, p. 7]. Which means that their data set comprised of all the main international competitions and some national penalty shoot-out data. The results of this statistical analysis were separated into two categories: before and after 2003 as explained why previously. In order to find statistical evidence of unfair treatment between teams they conducted multiple analysis. In the first one, they separated the set of observations between the following sources based on multiple factors such as "the nature of the crowd in the stadium", "the quality of the team", etc. [2, p. 8]: FIFA rankings, UEFA rankings, Category (national division), Position (when same category), Experience and Home team.

The result of this first analysis was not successful in rejecting the null hypothesis of 50% winning rate for both teams. In the second one and the main results of their research, they computed the effect of the average treatment of the difference between winning when shooting first and winning when shooting second. They found a significant advantage for the team shooting first in the shoot-out equal to a 60.5% winning rate. Thus, accepting the alternative hypothesis of the first shooting team having a winning rate bigger than 50%. They pushed their analysis further by using a "regression framework" which confirmed their results on the significant effect of kicking

first in winning a shoot-out. This "regression framework" consisted of three logistic regressions[1] and three probit[2] regressions based on a combination of the previously cited factors. The results of these analyses will not be presented in full as the only significant factor for all the regression models was the binary value determining if a team was shooting first.

They also watched 20 matches and conducted a survey on 240 players in order to see what a player would choose between starting to kick or being second. In the first case only one chose to kick second while in the survey 100% of the players chose to kick first explaining explicitly in 96% of the cases that they intended to put pressure on the other team. Furthermore, they analysed the data deeper in order to find more specific information on the effect of shooting first. To do so, they computed the success rates of each team for each round and the following results were all in favour of the first kicking team with a decrease in the effect after the fifth round. In about three quarters of the case a match is won within the first five rounds, with the first shooting team having a 65.9% chance of winning the shoot-out. This advantageous winning probability decreases to 52.9% when matches go for more than five rounds. They concluded that psychological pressure had a detrimental effect in their experiment. Although all the factors influencing said pressure are "too complex to clearly discern the impact of these elements on human behavior" [2, p. 16]. This randomized experiment offers a more reduced environment allowing for a clearer analysis of this effect. Their research found that psychological pressure had a critical role in the outcome of their experiment. Furthermore, they also found that individuals are aware of such role and responded rationally to it.

From the perspective of the reader, one could question the validity of such results. Indeed, their small data set could be biased and not represent

[1] Also known as logit model, it is a regression analysis thus a predictive one. It can be used to predict binary outcomes based on multiple selected variables following a logistic function. It allows us to estimate the probability of a success based on various independent variables. [20]

[2] cf. logit. The probit differs from the logit in the cumulative distribution function used. For the probit we use the normal distribution. [20]

the true nature of a shoot-out. That is why Kocher *et al.* [13] ran their own tests to contest the findings of Apesteguia and Palacios-Huerta [2]. In the past, Kocher *et al.* 2008 [14] already analysed a dataset on the German soccer competition (DFB-Pokal). While the first shooter advantage was not the main focus of their paper, they found that between 1986 to 2007 only 48.4% of the penalty shoot-outs were won by the first shooting team. This is not only far from the 60,5% found by Apesteguia and Palacios-Huerta [2] but also not significantly different[3] from 50%. Kocher *et al.* [13] also explained that at the time they were writing their paper, the work from Apesteguia and Palacios-Huerta [2] is the only one showing empirical evidence of a first shooter advantage. All other papers on different sports showed no advantage or difference in winning ratio for the first shooter.

In order to push the analysis further, they decided to extend the set used by Apesteguia and Palacios-Huerta [2]. They went from 129 shoot-outs to 540 strictly using the same tournaments as Apesteguia and Palacios-Huerta [2] did thus, approaching almost the full set of shoot-outs that had taken place in such tournaments. They found a 53,3% winning rate for the first kicking team. However, they failed to find significance in both methods they used. In the binomial test which is used to reject or not the null hypothesis of the probability of an outcome (e.g., a coin toss being 50% head and 50% tail based on 100 throws) they found a probability value (p-value) of 0,13 unable to prove a significance of the 53,3% winning rate at the 95% confidence interval (p-value < 0.05) commonly used in statistics. In the probit regression model, they used covariates [4] based on the location of the match. Either at home for one of the team or not (neutral ground). They found a p-value [5] of 0,15 which fails to show significance of a first shooter advantage. Their superset consisted of 76.2% of all the shoot-outs in the considered tournaments between 1970 and 2003.

They concluded that the results from Apesteguia and Palacios-Huerta [2]

[3] A two-sided binomial test intends to help us in determining if the outcome of an experiment is different between two distinct groups. It follows a discrete probability distribution. [9]

[4] Variables linearly associated to other variables. [9]

[5] We assume they carried out a two-sided binomial test

were due to a sampling bias and to prove it, they computed the probability of taking 129 shoot-outs out of 540 and getting a 60,5% winning ratio for the first kicking team while the rest of the set (n = 411) had a 51,1% winning ratio. The results showed that the probability of finding such a set was less than 8%. As their set also was a subset of the total of shoot-outs in selected tournaments, they ran the same calculation in order to get the probability of finding a 53,3% winning rate for the first kicking team out of 540 shoot-outs in a set of 709. The probability was close to zero. They insisted that while the 53,3% was more than the expected 50% the insignificance of the results could be mitigated by using an even bigger dataset including other competitions. Also, results could even be in favour of the second shooting team.

These findings did not please Palacios-Huerta [16] who wrote a whole book about game theory. In their book, they extensively look how soccer can help economics. To do so, they analyse more thoroughly the mechanism of penalty shoot-out previously studied by Apesteguia and Palacios-Huerta [2]. They also analysed the strategic choice of the players on the field from an economic point of view.

As the average time a shot takes from the kick to the goal is approximately 0.3 seconds which is not giving the necessary reaction time to allow the goalkeeper to jump on the path of the ball. This suggests that both the goalkeeper and the player should move at the same time. After calculating the probabilities of the different strategies and gathering data on 9017 penalty kicks from 1995 to 2012 including the targeted zone of the shot (left "L", right "R", centre "C") and the main foot of the kicker (left "L" or right "R") they found that players were mostly shooting on the side of their main foot. As the dataset mainly comprised of right-footed shooters (80%), they decided to standardize it by simply saying if a shot was made aiming at the "natural side" of the shooter or not (right handed player shooting right and left handed player shooting left). That way they managed to gather probabilities on where a kicker would shoot and where a goalkeeper would jump for each of the shots.

They found out that in about half of the shots the strategy of the goal-

keeper matched the one of the shooters and that in other cases there was a 21.6% probability of player shooting right and goalkeeper jumping left and a 21.7% probability of player shooting left and goalkeeper jumping right. Using their sample, they computed the scoring probabilities as a two-by-two matrix depending on the choices of the goalkeeper and the player (i.e., shooting/jumping left or right). This matrix is shown in Figure 2.1 with:

- k_L the kicker shooting left.
- g_L the goalkeeper jumping left.

Then, they could return the Nash equilibrium[6] and found that those numbers were matching with the empirical data as shown in Figure 2.2.

	g_L	$1-g_L$
k_L	59.11	94.10
$1-k_L$	93.10	71.22

Figure 2.1: Two-by-two matrix of empirical winning probabilities [16, p. 20]

	g_L	$1-g_L$	k_L	$1-k_L$
Nash Predicted Frequencies	40.23%	59.77%	38.47%	61.53%
Actual Frequencies	41.17%	58.83%	38.97%	61.03%

Figure 2.2: Comparison between Nash predicted and actual frequencies [16, p. 20]

Afterwards, they tested their data against two main implications:

- A test of equal scoring probabilities across strategies. The null hypothesis of the equal scoring probability test[7] was not rejected either at the aggregate level or at the kicker/keeper level.

[6] Also known as the prisoner's dilemma, it "is a strategy profile such that no player has a unilateral profitable deviation. This is the minimal stability criterion one may ask a profile to satisfy."[9, p. 50]

[7] "the Pearson product moment correlation coefficient for sample data (commonly referred to more simply as the sample correlation coefficient) is computed by dividing the sample covariance by the product of the sample standard deviation of X and the sample standard deviation of Y."[9, p. 71]. It is used to measure the statistical relationship between two variables.

- A test of serial independence to determine the choice of kicking/jumping area and its relation to previous choices. The results were similar to the test of serial independence [8]. This test was unable to reject the null hypothesis of randomness.

They considered computing a logit for each player but it was not ideal because the choice of a player could depend on the characteristic they might perceive from their opponent on the field.

During their second chapter, Palacios-Huerta [16] aimed at finding how players were establishing their strategy during a shoot-out in a controlled laboratory experiment. First, they set up a simple card and dice game with eighty pro soccer players (divided in two equal groups for kickers and goalkeepers) where they had a chance to win real money and providing them with pay off probability as shown in Table 2.1.

They compared the empirical data with their calculation of the expected minimax probabilities [9] and rejection of the null hypothesis. The game worked as such:

- Both players were separated by a cardboard when choosing a card (amongst two).
- Based on the card both players chose, the kicker (row player) would win based on 2 dices (one for units and one for tens).
- If the score was within the related range as shown in Table 2.1 (e.g., they both pick the same card, then row player would win if the dice throw resulted in 1-60)

This was helping in simulating the strategy of both players on the field (the card choice is the kicking/jumping area and the dices represent the related scoring probability). They played 15 rounds for training and then 150 for real. They compared the empirical data with the minimax calculation and found that the numbers were close (within 1-2%). However, when looking

[8] Using two different symbols for each choice (i.e., shooting to the natural side or not), the "runs test" intend to find a sequence in the succession of those symbols[16]

[9] This is an algorithm in game theory which goal is to minimize the maximum loss [15]

at the decisions made by the row players, they found that the numbers were significantly different (0.333 and 0.667 for empirical data vs 0.363 and 0.637 for minimax). This implies that players are well aware of the Nash equilibrium. In order to determine if the choices were independent, they conducted another test which "reports the relative frequencies of each combination of actions for each of the pairs in the sample." [16, p. 39]. They found that the rejection rate was well inside the forecast thus reinforcing previous findings on the statistical difference of observations and minimax.

Shooter/Keeper	**Left**	**Right**
Left	60%	90%
Right	95%	70%

Table 2.1: Two-by-two matrix of winning probabilities as given to participants

Then, they performed a Pearson test in order to analyse the distribution of play and the winning rates. The test showed that "at the individual level, the hypothesis that scoring probabilities are identical both across strategies and to the equilibrium rate cannot be rejected for most players at conventional significance levels." [16, p. 40]. And the same goes when aggregating the results. They performed a Runs test and a logit regression to see if players were taking decision randomly in a laboratory. In both test this hypothesis could not be rejected.

In their next chapter, they talk about the different lesson learned from their analysis. First, the possible bias in their dataset mainly composed of European soccer data as this is a highly competitive environment. Indeed, the statistics on playing minimax might differ in other parts of the world. The second lesson is about friendship between players or collaboration. They conducted the same experiment as previously described but they gathered 15 pairs of players from the same team. The expected rejection ratio was exceeded thus, they rejected the null hypothesis of playing at the Nash equilibrium. They explained that it could come from the need of a team to collaborate on the field. The third lesson showed them that the lab experiment should as close as possible to the situation on the field (by using

shooter and goalkeeper pairs). To demonstrate their point, they gathered 15 pairs of goalkeepers against goalkeepers and shooters against shooters and the results showed to be drastically different. Indeed, there was an absolute rejection of the null hypothesis of equal payoffs for different strategies. The last lesson is about reproduction of behaviour in a face-to-face situation. They reported the results of a Japanese research team who developed a bot able to get a 100% winning rate at Rock-Paper-Scissor in using a high-speed camera and a high-speed computer able to determine the sign chosen by a player in 1/1000 of a second and react to it. Even if the time laps to react to a shot in a human-to-human situation is too short, the information still gets to the brain. In a lab situation such information would have the time to be processed by each player, potentially affecting their strategic behaviour. They noted that in their experiment, separating both player with a cardboard piece when selecting a card allowed to control such possibility.

Coming back to our main interest in determining the existence of a first shooter advantage, Palacios-Huerta [16] created a superset of their previous research Apesteguia and Palacios-Huerta [2]. They collected 1001 penalty shootouts gathering 10731 penalty kicks over the period 1970-2013 and ran the same tests as shown in Figure 2.3. They found that first shooting teams had a 60.6% winning probability thus slightly more than the previous research. They confirmed those results by carrying out four regression analysis as shown in Figure 2.4. Those regression models all showed a clear advantage for the first kicking team. The next thing they did was asking players and coaches from Spain what their strategy was when given the opportunity to choose between kicking first or not. Nobody answered that they would choose to go second but a maximum of 9.5% (for professional coaches) answered that it depended on the situation which is consistent with results from Apesteguia and Palacios-Huerta [2]. After presenting researches on mechanisms from different cognitive (i.e., Chess) and non-cognitive sports (i.e., soccer), they concluded that information on performance had an impact on the performance of the team. It means that on top of strategical reasons of going first or second "there are, in addition, psychological reasons why

leading or lagging may affect the performance of the competing agents." [16, p. 81]. Then, they aimed at determining what makes a good and fair sequencing order by experimenting three mechanisms with professionals from Spain's first soccer league. With "A" the team shooting first and "B" the other one, we have:

- The standard ABAB (currently used in soccer).
- The Tennis tiebreak (ABBA).
- The Prouhet-Thue-Morse (i.e., ABBABAAB were we invert the sequence of the tennis tiebreak every time).

Carrying 200 penalty shoot-outs composed of 4 shots per team they found without surprise that for the standard rule there was a 61% winning probability for the first kicking team. Interestingly, when using the tennis tiebreak mechanism this advantage lowers to 54% but with Prouhet-Thue-Morse it gets down to 51% representing a 2% winning rate advantage difference instead of the 22% from the standard mechanism. They conclude this chapter by saying that "the advantage may be not only substantial but entirely psychological." [16, p. 85].

They went deeper in their study in analysing the areas of the brain that responded to strategical dilemmas and in the economical implication of their results but we will not speak about it here. As we saw, what looked like a simple question became quickly interesting. Indeed, analysing data is not as straight forward as one would think and we should be careful when trying to provide an answer for this issue.

Table 5.1. Percentage First Team Wins in International and National Competitions 1970–2012

	Number of shoot-outs	First team wins (%)	
International Competitions			
1. National Teams			
World Cup	22	59.1%	
European Championship	15	33.3%	
Copa América	18	61.1%	
African Nations Cup	20	60.0%	
Gold Cup	10	70.0%	
Asian Cup	16	56.3%	
2. Club Teams			
European Champions League	49	63.3%	
European Cup Winners' Cup	32	62.5%	
UEFA Cup	110	55.5%	
National Competitions			
German Cups	183	49.7%	
English Cups	179	53.6%	
Spanish Cup	347	72.3%	
All International Competitions	292	57.8%	*p*-value: 0.0139
All National Competitions	709	61.0%	*p*-value: <0.0001
Total	1001	60.6%	*p*-value: <0.0001

Figure 2.3: Percentage first team wins and national competitions [16, p. 76]

Table 5.2. Determinants of Winner of Penalty Shoot-Out

	Probit	Probit	Logit	Logit
Constant	−0.267	−0.273	−0.437	−0.403
	(0.217)	(0.506)	(0.343)	(0.609)
Team kicks first	0.657***	0.633***	1.027***	1.012***
	(0.140)	(0.134)	(0.192)	(0.187)
Home field	−0.092	−0.114	−0.128	−0.165
	(0.210)	(0.244)	(0.352)	(0.340)
Neutral field	−0.052	−0.048	−0.073	−0.079
	(0.275)	(0.314)	(0.422)	(0.412)
Category	0.002	−0.007	0.011	−0.007
(1 if higher)	(0.182)	(0.170)	(0.272)	(0.228)
"Team kicks first" interacted with				
Home field	No	Yes	No	Yes
Neutral field	No	Yes	No	Yes
Category	No	Yes	No	Yes
N (teams)	2002	2002	2002	2002
Adjusted R^2	0.106	0.108	0.106	0.108

Note: Regressions in columns 2 and 4 also include fixed effects for Champions League, UEFA Cup, National Team, and National Cup competitions, as well as interactions between Home and Neutral field and Category.

Figure 2.4: Regression analyses [16, p. 77]

2.2 Deeper statistical analyses

A question still remains, until now we only looked at empirical data to determine if there was a first shooter advantage. We could also theoretically determine the impact of a factor related to psychological pressure in giving an advantage to one of the team based on the shooting position (first or second). Following this path, we could also try to find other factors that could lead to a first shooter advantage.

In their paper Vandebroek *et al.* [23] assume that psychological pressure has an impact on penalty shoot-out leading to a first shooter advantage. However, they say that there are a lot of other factors to be taken into account when computing the impact of such pressure. One of the first factor they analyse is the perception of a success from the first shooter by the second one. Indeed, the "observation of an opponent's successful performance can increase anxiety about one's ability to match that outcome, leading to worse performance for the second participant (i.e., a lagging behind effect)."[23, p. 5]. To tell the story of the first shooter advantage they refer to:

- Apesteguia and Palacios-Huerta [2] using a set of 129 shoot-outs.
- Kocher *et al.* [13] using a set of 540 shoot-outs.
- Palacios-Huerta [16] using set of 1001 shoot-outs.

In order to deepen the analysis of the Palacios-Huerta [16] – Kocher *et al.* [13] disagreement they created a mathematical model to represent the lagging-behind effect and its relationship with the first shooter advantage:

- They see each kick as a succession of binary outcomes multiplied by two (for each team). the amount of possible path is calculated as follows: 2^{2n} meaning that for n = 5 rounds we have 1024 possible outcomes.
- They keep in mind that a team led to a score by a factor more important than the remaining number of rounds (within the first five rounds) will lose before reaching the fifth round (e.g., at the end of round 3 we have 3-0 then the match stops as it would be impossible for the other team to come back to the score within 2 rounds).

- They assume that previous researches on the simplicity of the task and expected outcomes are right (75% success rate for penalty shots).
- They consider that the physical effort taken to kick the ball was not relevant.

The first proposition of their model consists on finding a relationship between the first shooter advantage and lagging behind pressure:

1. If there is no lagging behind pressure there is no first shooter advantage.
2. The first shooter advantage rises with the lagging behind effect.

For simplicity they start with a model in two phases "regulation" and "sudden death" consisting of 1 round each and constructed the tree of all outcomes in a 2-round shoot-out. They mathematically demonstrated that the first shooting team had a higher probability of leading after round 1 in the presence of a lagging-behind effect (λ). It also shows that the probability for the first team to keep the lead after taking it in the first round is equal to the probability of the second team to keep the lead after it has taken it in the first round. The same logic applies in the case of a return to the score as it does not matter who leads in terms of probability. The purpose of the model is to calculate the pressure on the lagging team and how much more probability there is for the first shooter to be in an advantageous situation at the end of the second round. It goes as follows:

- The first shot has a p probability of being scored and 1-p not to be.
- Going to the next round and if the first shot was scored, the second team has a scoring probability of 1-p+λ and so on.

The purpose of this proposition was not to show if lagging behind pressure existed but to establish that if it existed then it induced a first shooter advantage. So, using p=.75 and λ=.20 they showed that the existence of a lagging effect created a first shooter advantage of more than 28,6% between winning probabilities of both teams.

Coming back to previous researches, they criticized the rather small size of their chosen datasets. Furthermore, deploying a theoretical model of a small dataset with lagging behind effects of .05 would lead to a first shooter advantage of 9,1% representing a winning probability of 54,5% for the first shooting team. Thus, in order to push their analysis further they created a model to simulate the statistical power [10] of a dataset of size 540 in rejecting the null hypothesis of a 50% winning ratio. By configuring the simulation with p=.75, λ=.05 and a sample size of 540 for 10000 iterations, they found that in 56.5% of the cases the model would reject the null hypothesis and that might be the reason why Kocher *et al.* [13] failed to reject the null hypothesis while Apesteguia and Palacios-Huerta [2] did not. The fact that Palacios-Huerta [16] found evidence of a first shooter advantage in his second research in 2014 using a bigger dataset is consistent with this analysis.

The second proposition is that the first shooter advantage "is an increasing function of the base probability" [23, p. 15]. The base probability is the overall probability that a shot will be successful previously referred as p=.75. Here they insist on the relationship between the difficulty of a task and the first shooter advantage. Indeed, if a task is harder (i.e., with a lower success rate) then the effect would be reduced but always positive as long as $p > 0$ and $\lambda > 0$. Of course, this interpretation is tied to their specific model. A different model could show that harder tasks (with lower success rates) could decrease the pressure on shooters to perform (e.g., p = .25). However, such model could show an increase of pressure for the goalkeepers. Taking the example of hockey where the National Hockey League (NHL) gave the success rates of a shot being .33, a study from 2015 on NHL shoot-outs found no equivalent first shooter advantage. They note that in such different environments where scoring probability (difficulty) varies it may require other modelling approaches.

In their other propositions they investigated the relation between a first shooter advantage and the position in the shooting sequence of the star player of each team (players with higher scoring probability). They found that if

[10] The power of a statistical test refers to its ability to reject the null hypothesis when it is false and accept it when it is true, (cf. Power curve)[9]

lagging behind effect exists, both teams should choose their best kicker to shoot first. They showed that by using $\lambda = .20$ the first team increases its winning rate by 5.72% in placing its star player in first position instead of the fifth. They also showed that the ordering had a bigger impact on the first team as the second team only decreases the first shooter advantage by 3.14% by placing their star player first instead of fifth. They also looked into two other effects:

- the pressure of a missed shot meaning the loss of the match;
- the pressure of a successful shot meaning the win of the match.

They computed that if the scoring probability was higher than 50% then the second shooting team had 1.67 times more chances in finding itself in a disadvantageous situation with the negative effect of the fear of losing outweighing the positive effect of the prospect of winning.

Following Vandebroek *et al.* [23] study we can put the emphasis on the importance of correctly estimating the lagging behind effect. Indeed, the three first studies researched the same topic yet found different results. We saw that if psychological pressure had an effect on scoring probabilities, then there was a first shooter advantage. Furthermore, the second shooting team had a higher probability of finding itself in a disadvantageous situation in the current shoot-out mechanism. Strategically, the first shooting team should put its "star player" first.

Continuing our journey through this penalty shoot-out analysis, researchers such as Silva *et al.* [19], Arrondel *et al.* [3] and Rudi *et al.* [18] carried out their own empirical analyses. However, they apply themselves to the difference in pressure perceived by both teams. It is interesting to see the choices they made for the data set creation and analysis. It also provides us with ideas and different perspectives about this issue.

In their paper, Silva *et al.* [19] analyse the possible existence of first shooter advantage as the team going second has to play "catch-up". They find that this bias exists in soccer while it does not in tennis. So, they started comparing both sports in high level competitions in order to suggest

an alternative to the current mechanism. For their study, they admit the existence of a lagging behind effect. They start by suggesting that:

> "One possible solution is to adopt the tennis tie-break format, which follows an ABBA pattern. Team A is followed by team B, before team B goes again. Team A would then get two successive penalties, and so on until there is a winner."[19, p. 2].

Their reasoning on the fairness of tennis tie-break is based on 345 observations from the 2017 grand slam. They found that 163 first moving tennis players won their match (47.25%) without being statistically different from 50%. Following the same logic, they establish that out of 232 penalty kicks from 1970 to 2017 138 first kicking teams ended up winning the match (59.48%) being statistically different from 50%. By aggregating all the shoot-outs in a matrix as shown in Figure 2.5 with r the score of the first kicking team and s the score of the other team we have the number of matches that ended up in various score combinations. For example, we have 33 matches that ended up in 5-3 for the first shooting team. They established the null hypothesis to be: "r and s are exchangeable" or no advantage for the first shooting team. Carrying a "Hollander's test of bivariate symmetry for H_0[10], the RFPW test of symmetry for H_0[17], and Wilcoxon's signed rank test for H_0[11]"[19, p. 22]. They concluded that there was a highly significant probability of a first shooter advantage and that FIFA should adopt the tennis tie-break mechanism.

The next researchers, Arrondel *et al.* [3] look into the penalty shoot-out of the French soccer competitions by using data collected from the "Coupe de la ligue", the "Coupe de France" and "Le trophée des champions" they aim at finding the various psychological factors influencing the scoring probability of a shot. Their data set comprised of 239 shoot-outs (out of 252 over the analysed period) and 2504 penalty kicks. After analysing their data, they

[11] "The Wilcoxon signed-rank test is a non-parametric procedure for analyzing data from a matched-sample experiment. The test uses quantitative data but does not require the assumption that the differences between the paired observations are normally distributed. It requires only the assumption that the differences between the paired observations have a symmetrical distribution, and examines whether the population differences are centred on the value zero (i.e. have a mean or median equal to zero)."[15, p. 571]

Table 3. Observed quantity of goals scored by a team starting the penalty shoutout, *r*, and the scores of the other team *s*.

		r											
		0	1	2	3	4	5	6	7	8	9	10	11
	0	-	0	2	5	0	0	0	0	0	0	0	0
	1	0	-	1	14	14	0	0	0	0	0	0	0
	2	1	1	-	5	22	0	0	0	0	0	0	0
	3	0	5	11	-	14	33	0	0	0	0	0	0
	4	0	0	16	24	-	16	0	0	0	0	0	0
s	5	0	0	0	0	20	-	7	0	0	0	0	0
	6	0	0	0	0	0	8	-	3	0	0	0	0
	7	0	0	0	0	0	0	2	-	2	0	0	0
	8	0	0	0	0	0	0	0	2	-	1	0	0
	9	0	0	0	0	0	0	0	0	1	-	0	0
	10	0	0	0	0	0	0	0	0	0	1	-	0
	11	0	0	0	0	0	0	0	0	0	0	1	-

Figure 2.5: Observed quantity of goals scored by first shooter, r, and by the other team, s. [19, p. 4]

established that the team kicking first had no advantage on the other team and that the only explanatory variable they found was the difference in the level of the competing teams. However, they dug deeper into their data set in order to find other potential psychological pressure factors. By analysing the success rates of all shots for each round they did not find any significant difference between the two teams.

Furthermore, in opposition to Apesteguia and Palacios-Huerta [2] they did not find that the team shooting first had a leading advantage at the end of the different rounds. They even found the opposite for the first round. Considering three types of scenarios, they analysed the scoring probability of each one of them:

- The "break point kick" or leading the score with the same number of kicks resulted in a 70.8% scoring rate.
- The "survival kick" or the opposite situation resulted in a 70.2% rate.
- The "catch up kick" or the possibility of equalizing leveraged an 83.6% rate.

They explained that based on those rates they cannot say that a difference in scoring rate comes from psychological pressure but rather from the comfort that players might perceive when taking a shot.

In the second part of their analysis, they focus on the uncertainty of the outcome of a kick. Indeed, a successful kick does not necessarily mean that the team of the kicker will win the shoot-out and Vis Versa (a failed kick does not necessarily mean that the team will lose). Assuming the constant scoring probability of 73%, they constructed a tree with the possible outcomes. They analysed two parameters:

- The "stakes" driven by the difference between the probability of winning the shoot-out if the kick scores and the probability of winning if the kick fails.
- The "pressure" driven by the probability of losing the match before kicking.

They showed that both are "emotions" that negatively impact the scoring probability. Indeed, when maximizing "pressure" (all other things being equal) the scoring probability falls down to 66,3% while when maximizing the "stakes" it falls down to 62%. They concluded that players might choke under pressure and that it was almost objectively observable in their analysis. Furthermore, such results did not imply a first shooter advantage. They added that the pressure on the goalkeepers was overlooked but that it was significantly lower than the one on the players as it is extremely rare for them to be criticized for failing to catch a penalty kick.

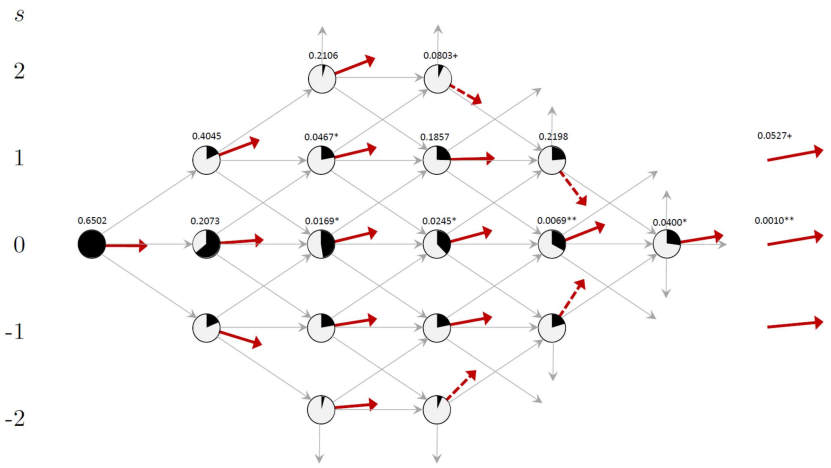
In their paper, Rudi *et al.* [18] do not assume the existence of a first shooter advantage but try to look into it. To do so, they got inspired by the three first exposed researches [2], [13] and [16]. They built a superset of the data used by Apesteguia and Palacios-Huerta [2] adding about 45 tournaments and employing undergraduate students to manually gather the missing data. Their findings were in favour of a first shooter advantage with a winning rate of 55%. This is closer to the winning rate found by Kocher *et al.* [13] while for them it is strongly significant. However, Rudi *et al.* [18]

add that those numbers are not sufficient to say if a change in the shooting order would mitigate this advantage.

To see how another sequencing order could mitigate said advantage, they constructed a tree in Figure 2.6 using different states defined as "the combination of the score difference s in favour of team A at the beginning of round t."[18, p. 5]. In that figure we can see that the lighter arrows linking the different nodes represent the possible transitions between states (score difference). When the thicker arrows point to the north it means that the first shooting team is the shoot-out winner. The pie charts indicate the proportion of matches that reached the corresponding state. The value of each one-sided t-test on the drift in advantage of the first shooting team are shown on top of each note. By performing a statistical difference test [12] on states which s=0 (tie), they found a positive drift in winning probability for the first kicking team as from round 3. Concerning the drift in situation where s≠0 (advantage for team A or B) they found the drift to be higher for team A while there were only statistically significant results for round 3. Furthermore, they found a positive drift difference of 4,32% in favour of team A with a weak yet significant statistical power (90% confidence interval) by aggregating rounds 2 to 4 with $|s| = 1$.

To understand better the advantage of the first shooter they computed the scoring proportion for both team A and B. They found that the proportion for A was always higher than for B except for the first round while the difference is not huge (75.2% vs 75.8%) the numbers are not statistically significant. The difference becomes bigger and more significant in round 3, 4 and 5 though. They concluded that estimating the value of policy change would benefit from a randomized trial. However, the time and cost of such methods make it nearly impossible to set up. The main strength of their paper is the use of a network model which allows for visual and intuitive understanding of the potential strategic analysis from soccer team manager. They added that the order sequencing would benefit of more in-depth research.

[12] In a t-test, the student t-distribution is used. This set of tests is performed for small sample size with an unknown standard deviation.[9]



t	1	2	3	4	5	6+	Total	#shots
A	75.2%	76.1%	75.5%	74.3%	73.2%	72.5%	74.7%	8854
B	75.8%	74.9%	71.3%	70.6%	69.0%	69.0%	72.2%	8348
p-value	0.6540	0.2258	0.0028**	0.0095**	0.0150*	0.0323*	0.0002***	

Table 2 Summary statistics using extended data of shot performance for first and second shooter, by round. P-values of alternative statistical tests are reported, with significance levels indicated by (+) 10%, (*) 5%, (**) 1% and (***) 0.1%.

Figure 2.6: Summary statistics from Rudi *et al.*[18, p. 6].

Now that we have seen the different empirical analyses and their various results, we may consider that there is a high probability of a first shooter advantage. Table 2.2 shows the different outcomes and general details of those empirical researches. The null hypothesis is: "There is a 50% chance for both teams to win the game no matter the sequencing order". The alternative hypothesis is: "There is a significant winning advantage for the first shooting team depending on the sequencing order". We can also see that the size of the data set is quite an important feature when considering statistical significance.

Authors	Year	First Shooter	Set Size	Reject H_0?	Description
Apesteguia *et al.*	2008	60.5%	129	Yes	Pre-2003 penalty shoot-outs.
Apesteguia *et al.*	2008	59.2%	269	Yes	Including post 2003 cases.
Kocher *et al.*	2012	53.3%	540	No	Superset [2], criticised the size of the data set.
Palacios-Huerta	2014	60.6%	1001	Yes	Superset [2] deeper psychological and economic analysis alongside with lab experiments.
Vandebroek *et al.*	2018	/	/	/	Introduced the link between psychological pressure and first shooter advantage.
Silva *et al.*	2018	59.48%	232	Yes	Comparison with tennis tie-break data from 2017, concluded that the FIFA should take on the tennis mechanism.
Arrondel *et al.*	2019	50%	239	No	French penalty shoot-outs and break-point, survival and catch-up kicks. Difference in comfort of the kick rather than sequencing order.
Rudi *et al.*	2019	54.86%	129	No	Higher statistical advantage. Looked for a drift between tie situation and found an advantage for the first shooter.

Table 2.2: Summary of the different empirical researches

2.3 Suggesting new mechanisms

Looking at those previous studies, it became obvious that the first shooter advantage might be real. Thus, following studies from previous researcher Brams and Ismail [5], Csató [6] and Anbarcı *et al.* [1] proposed ways of finding better mechanisms for the penalty shoot-out in soccer. They did not carry out their own empirical analyses but re-used the one previously found.

In their paper, Brams and Ismail [5] admit the existence of a first shooter advantage in the current penalty shoot-out mechanism of soccer competitions. Nonetheless, they still prove it in order to compare its performance in terms of fairness with other mechanisms. They discuss two alternative rules:

- The Catch-Up Rule (comprising of 3 smaller rules) works as such: "In the first round, one team is advantaged (by kicking first). If it is suc-

cessful (W) and the other team is not (L), rule 2 says that L becomes advantaged on the next round (whether it was advantaged or disadvantaged in the current round). If both teams on a round are either successful or unsuccessful in scoring a point, neither team becomes L or W the contest is U. Rule 3 says that the team that was advantaged in a U round becomes disadvantaged in the next round." [5, p. 6] with U standing for "unresolved". They show that this rule tends to reduce the split between the winning probabilities of both teams. More exactly with A the first kicking team with scoring probability p=3/4 and B the second one with scoring probability q=2/3 we can see that "A is favored in 2/39 = 5.1% more cases than B, whereas recall that under the Standard Rule A was favored in 59.4% more cases than B" [5, p. 12]. This represents a factor 10 bias reduction of the first shooter advantage compared to the standard ABAB rule.

- The Behind First, Alternative Order Rule works as such: "If one team is behind, it kicks first on the next round; if the score is tied, the order of kicking alternates (i.e., switches from the previous round)" [5, p. 7].

The overall analysis shows that both the Catch-Up Rule and the Behind First, Alternative Order Rule decrease the split between the probability of A winning and the probability of B winning when the probability of a tie decreases while the standard rule increases said split.

They pushed the analysis further in computing the encounter of a "good team" and a "bad team" and they showed that the standard rule was unfair as "bad team" had a higher winning probability than "good team" when taking the first kick despite being less good. However, the two other rules give higher winning probabilities to "good team" for any starting position. Concerning Sudden death and strategic manipulation they show that while the probability of A winning rises for 5 rounds of sudden death (from 60% to 63,7%), it decreases with the Catch-Up Rule (from 52,6% to 51,4%) using regular values for p and q. We can clearly see that the Catch-Up Rule reduces greatly the first shooter advantage.

Computing the winning probability for different value of p and q they

also show that the Catch-Up Rule gives closer to 50% winning rate while still advantaging A. However, as the Catch-Up Rule depends on previous results and necessitates a Chain of Markov to be modelled [13], one can speculate that team managers would try to establish a strategy in order to get more advantageous winning probabilities. To answer that, Brams and Ismail [5] show that the Catch-Up Rule is strategy proof (coaches could not decide to fail a shot in the prospect of gaining an advantageous situation) for (p-q) $\leq \frac{1}{2}$. As this condition is likely to be met in real life soccer competition, one can consider the Catch-Up Rule to be strategy proof. Short, they showed that the Catch-Up Rule was fairer than the standard rule. When considering other sports, they did not find that the Catch-Up Rule had the same equalizing power as in soccer.

They added that spectators would benefit from the Catch-Up Rule as it would enhance the suspense of the game if the first kicking team in each round depends from the previous round. Furthermore, even by using this proposed mechanism fairness would not be totally achieved. Indeed, to achieve such scenario the first phase of penalty shoot-outs (currently 5 rounds) should become an even number of rounds such that both teams would have the same chance of starting to kick in the same number of rounds. They also added that their analysis did not take into account matches played "home" or "away" but that it was a great opportunity for competitions where all teams are in the same location.

Looking at the research from Csató [6], we can see that they summed up all the interesting mechanisms that have been proposed to FIFA in order to mitigate the first shooter advantage:

- "Alternating (ABBA) Rule: the order of the first two penalties (AB) is mirrored in the next two (BA), and this sequence continues even in the possible sudden death stage of the shoot-out (the sixth round of penalties is started by team B, the seventh by team A, and so on).

[13] A Markov chain allows us to model a situation were the probability of an outcome evolves through the experiment[15].

- Catch-Up Rule [5]: the order of the penalties in a given round, including the sudden death, is the mirror image of the previous round except if the first team failed and the second scored in the previous round when the order of the teams remains unchanged.

- Adjusted Catch-Up Rule: the first five rounds of penalties, started by team A, are kicked according to the Catch-Up Rule, but team B is guaranteed to be the first kicker in the sudden death stage (sixth round)." [6, p. 184].

The third rule is a combination of the two others and is the newer design introduced by Csató. Those three mechanisms are compared between each other to find the mitigating power of each one. Based on the statistics determined by Apesteguia and Palacios-Huerta [2], they computed the average scoring probability for each of the teams to be 3/4 (p) for the first kicking team and 2/3 (q) for the second one. Those numbers are similar to the rates used by Brams and Ismail [5]. Using a two round experiment they establish that using the catch-up rule, the adjusted catch-up and the ABBA mechanism there give respectively a 51,6%, 49,5% and 51,1% probability for the first kicking team to win the game. While all three mechanisms are fairly closer to 50% than the regular ABAB mechanism, the adjusted catch-up rule seems to mitigate the first shooter advantage the most. Indeed, when extending the analysis to 8 rounds they computed that the catch-up rule and ABBA rule still provide the first shooting team with respectively 6.8% and 4.64% advantage. With the adjusted catch-up rule this number falls down to 1.92%. Furthermore, by plotting the winning rate of the first shooting team for different value of p when q evolves (with $0.5 \leq q \leq p$) in Figure 2.7, they showed that the adjusted catch-up rule was performing the best. All mechanisms were performing better for values of q closer to p.

Concerning the adoption of a new mechanism, Csató [6] define the complexity as being the number of (simple yes/no or short answers) questions a mathematician has to ask to a referee in order to tell which team should shoot first. The ABAB rule has complexity 0, ABBA has 1, Catch-up rule has 2 and the Adjusted catch-up rule has between 2 and 3. While the increased com-

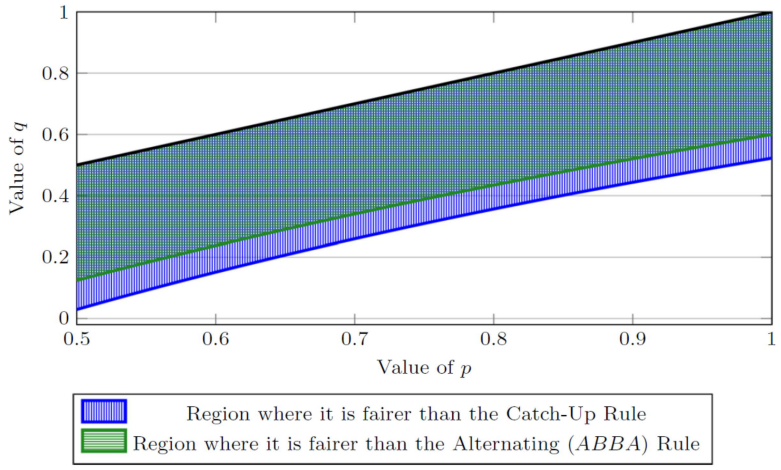


Fig. 3 The fixed scoring probabilities in sudden death which guarantee that the Adjusted Catch-Up Rule is fairer than the other penalty shootout designs

Figure 2.7: Comparison between the performances of the different mechanisms [6, p. 194]

plexity of the rule remains questionable it should be considered when it yields a considerable mitigation of the potential first shooter advantage. They conclude that tournaments are supposed to be fair and that the current shoot-out mechanism in soccer is not. They also found that the catch-up rule is not performing significantly better than the less complex ABBA mechanism. So, it would not be worth implementing it over ABBA. Nonetheless, the adjusted catch-up rule has shown to be a good candidate in mitigating the first shooter advantage. Furthermore, their complexity classification method can be used in the future to determine the applicability of any suggested (fairer) mechanism.

The last paper is from Anbarcı *et al.* [1] and is about theoretical statistics. They try to establish a method to evaluate the different mechanisms based on a set of criteria. It is purely informational as we will not try to create our own shoot-out mechanism. However, it gives an interesting view on the prospects of this long-discussed issue and the value of the present work. For the purpose of their research, they admit the existence of a first shooter advantage. They analyse the various mechanisms from different sports using

the penalty shoot-out method. They tried to find how dependent shoot-outs are compared to sequencing order. To get a clearer view of the psychological factors that may influence a player when taking a shot, they introduced a model where players not only care about the team winning but also about their own performance. Indeed, multiple studies that will not be discussed here showed that a player might rather take a safe shot instead of a riskier one that might fail. Then, they defined order independence:

> "as the requirement that equally balanced teams – in terms of their players' shoot-out abilities – have equal chance of winning any time when the score is tied at the beginning of any round, i.e., after equal numbers of attempts, under all state-symmetric equilibria of the induced shoot-out game." [1, p. 26].

They add that the mechanism which is put into place in case of a tie should be balanced even following an unfair coin toss. The main concept they use for their analysis is the state-symmetric equilibrium. This means that they will consider a mechanism to be order independent if at the beginning of a round there is a tie situation then each team will have a 50% scoring probability. For regular rounds they found that neither ABAB nor ABBA were order independent and that most intuitive mechanisms were not.

The next concept they introduce is the "exogeneous order" or an order of shooting that would be unrelated to previous scores. They found that it was only order-independent in one case: a random coin toss at the beginning of each round. They also defined an uneven score mechanism as such:

> "as long as the score is not tied at the end of a round, the probability of who kicks first in the next round is the same for Team 1 and Team 2 whenever they are in each other's shoes." [1, p. 33].

For the sudden death phase of shoot-outs, they found that the ABAB mechanism was not fair but that ABBA mechanism was. They concluded that:

> "For easy contests, we show that there is a class of order-independent mechanisms satisfying maximization of the expected number of

> attempts, which we term the behind-first mechanisms, such that the team that is behind in score after a round always kicks first in the next round, but if the score is tied after any round, then any random or fixed exogenous or endogenous order is admissible at the next round. In difficult contests, the same property is satisfied by ahead-first mechanisms." [1, p. 37].

In this last section we saw that establishing fair mechanisms is a complex task. Indeed, one need to take into account the empirical data and the potential bias related to it. Furthermore, we have seen that psychological factors related to the pressure felt by players on the field are the main drivers towards a first shooter advantage. However, we only saw analysis on observable sources of pressure (lagging-behind effect, prospects of losing or winning, etc.). Thus, it is of great importance that we clearly establish the factors influencing the outcome of a shot when testing a new mechanism and comparing it to the current ABAB one.

Chapter 3

Extraction of the data

The main objective of this work is to create a simulation tool for penalty shoot-out mechanisms. This method should provide us with results as close as possible to the real world. So, we deemed necessary to have a dataset with even more data than previous researchers. Indeed, as shown in the previous chapter, the need for powerful and significant statistical values is absolutely necessary when comparing various shoot-out mechanisms.

Furthermore, the latest extensive dataset dates back to 2019 (Rudi *et al.*) with 1635 shoot-outs comprising of European and international competitions. However, two soccer seasons have since passed and more will pass in the future. We were also limited in terms of resources and could not spend time on manually growing a dataset. Thus, we decided to use or make a tool able to gather data from publicly available sources which could help grow our data set through time. Our approach was the following:

1. First, we went looking for data from previous papers but no researchers published their dataset. More, they mention their own sources and methods such as Rudi *et al.* who hired undergraduate students to manually enlarge their dataset. Indeed, they built it based on publicly available soccer data from national league websites among other things.

2. Secondly, we looked at specialized websites on soccer data also mentioned in the previously documented papers. The following websites were investigated:

- worldfootball.net,
- "Union of European Football Associations (UEFA),
- the Rec.Sport.Soccer Statistics Foundation,
- the Association of Football Statisticians in the United Kingdom,
- the Spanish newspapers Marca and El Mundo Deportivo,
- www.weltfussball.de,
- and the archives of various soccer clubs." [2, p. 73]
- the "Football Association (FA) Cup ("Coupe de France"),
- the League Cup ("Coupe de la Ligue") and
- the Champions Trophy ("Trophée des champions")" [1, p. 26].

After reviewing the literature, we acknowledged many sources of data. However, we did not find a suitable data set.

3.1 Finding a source of data

The quest for the perfect data set or data source on the Internet is quite extensive that is why we did not consider all available sources. However, we established a list of criteria for the acceptance of a source:

- There should be a shot-by-shot description of penalty shoot-outs for us to have substance to work on.
- The match data should be sufficiently complete (playtime, goals, cards, spectators, etc.) to eventually eliminate other factors unrelated to an eventual first shooter advantage as described before.
- The source should include as many football matches from as many locations possible in the world.
- The website should be well-structured and consistent in its coding structure.

We first looked at Wikipedia for shoot-out data but even if available data was complete, the structure was not consistent and the search method was too complex. Quickly we turned to Google which also provided us with extensive match data. However, we still needed to manually search for the specific match and year in order to display it. Thus, it required another source of knowledge to make this work. After considering multiple bet platforms and football data websites our choice came down upon *Transfermarkt* (`https://www.transfermarkt.com/`) for its richness in football data and the easiness of finding penalty shoot-out results for specific leagues.

Our quest led us to a GitHub project called *worldfootballR* which is a R package able to extract various data from *FBref*, *Transfermarkt*, *Understat* and *Fotmob*. It could have been our holy grail. However, the data that could be scraped from *Transfermarkt* was limited to league season-level, team, player and club staff. Thus, not enough to perform a shot-by-shot analysis on penalty shoot-out.

After thinking about the various opportunities, we decided to develop our own tool to scrap *Transfermarkt* off what we needed. The choice of the technology was considered amongst three: R, Python and JavaScript. They were discussed following different criteria such as the simplicity of deployment, our experience, documentation on the internet and known related projects. In the end our experience with Python packages *BeautifulSoup* oriented our choice towards it.

3.2 Preparing the data

The process of extracting the links from *Transfermarkt* is depicted in Figure 3.1. We use the search engine of the website to look for the different competitions [1]. Then, we request all the matches that ended up in a penalty shoot-out. Those requests are answered with a list comprising of matches results and hyperlinks to specific match data among other things. The html results are all manually downloaded. This task has to be carried out for every

[1]The extraction took place between the 2021-12-08 and the 2022-01-25. It comprised of 3154 matches within 30 different competitions.

competitions.

Afterwards, we use a python script (cf. Appendix B) to download batches of html code from the links embedded in the previously downloaded html file. This script works by aggregating all the names of the html files stored in the working directory and extracting all the hyperlinks pointing to the matches. However, those files must all come from *Transfermarkt* and follow the same structure as the extractor only recognizes given html "class" names. Finally, all hyperlinks are fed back to the script in order to download all the source html codes from the different match pages. Every hyperlink is related to a single match and extracted to a single file.

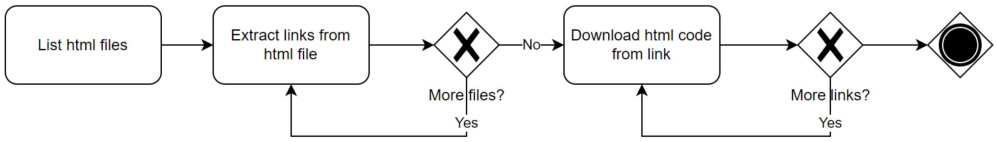


Figure 3.1: Process of downloading html code with match data

After getting the individual html file for each match, we can extract the raw data as shown in Figure 3.2. To do so, we use another python script (cf. Appendix C). The data extractor works as follows:

1. All the html files previously downloaded are now manually transferred to the working directory of this second script.

2. Afterwards, they are listed inside the program which loads each file as a "page".

3. Then one by one, each page is separated by "div" elements with class "Row" into items in a list.

4. If the "Row" has a "h2" element with text which is looked for, then the specific data from that "Row" is extracted into variables and if not, the script looks for the next row.

5. The text that is looked for is the following: "Timeline", "Goals", "Penalty shoot-out", "Substitutions", "Missed penalties" or "Cards".

6. Following the extraction of the data, after there are no “Row” left, all the variables are loaded into a json object.

7. Then, the json object is saved to a json file and the script restarts while there are more unextracted pages.

The parsing of the json objects is presented in Tables 3.2, 3.2, 3.3, 3.4 and 3.5.

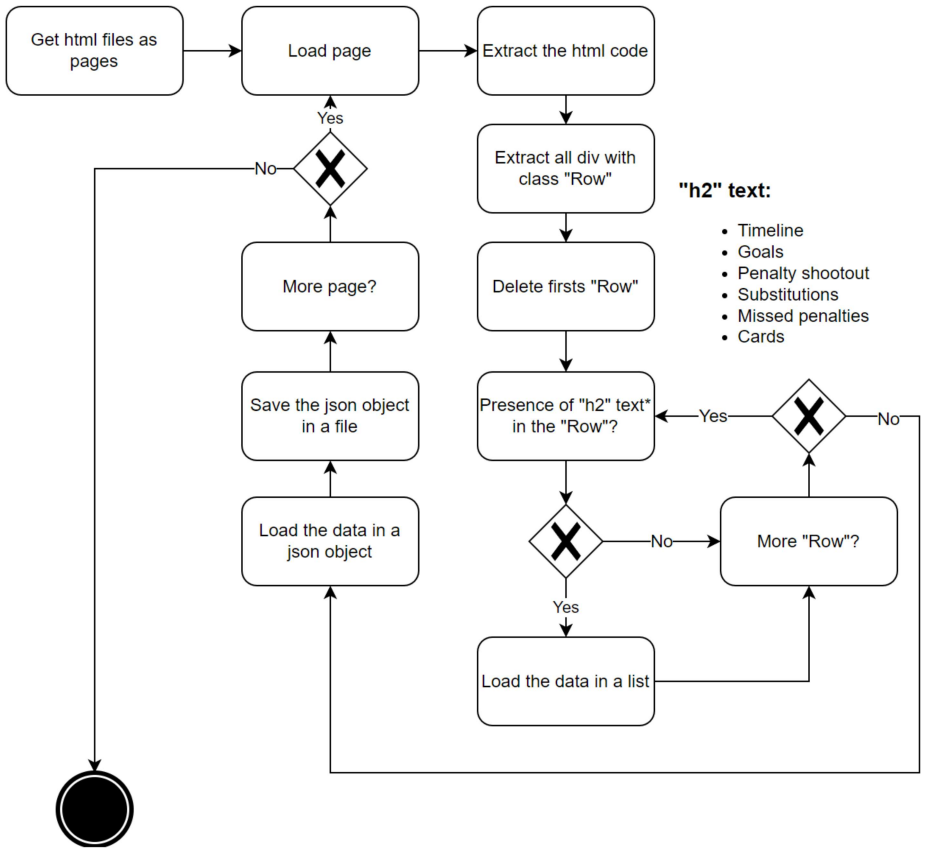


Figure 3.2: Process of downloading html code with match data

Field	**Description**
competition	The name of the competition in which the football match took place
date	The full date of the football match
year	The year of the football match
stage	The stage of football match in the competition
winner	The name of the team which won the football match
home_team	The name of the team which was playing at home (if relevant)
stadium	The name of the stadium where the match took place
attendees	The estimated number of people attending the match
referee	The name of the referee
team_A	The name of the team which took the first shot during the penalty shootout
team_B	The name of the team which took the second shot during the penalty shootout
full_time	The score at the end of the football match, before the penalty shootout
shoutout	The score at the end of the football match
return_match	The score of the first match between the two teams (if any)

Table 3.1: Data collected for each match.

Field	Description
round	The number of the round
player_A	The name of the player taking the first shot of the round
playtime_player_A	The amount of time the first player was in the game (not counting overtime)
yellowcard_player_A	The number of yellow cards the first player received during the game
goals_player_A	The number of goals the first player scored during the game
missed_penalties_A	The number of penalties the first player missed during the game
scored_A	1 if the first player scored his/her shot during this round of the shootout, 0 either.
player_B	The name of the player taking the second shot of the round
playtime_player_B	The amount of time the second player was in the game (not counting overtime)
yellowcard_player_B	The number of yellow cards the second player received during the game
goals_player_B	The number of goals the second player scored during the game
missed_penalties_B	The number of penalties the second player missed during the game
scored_B	1 if the second player scored his/her shot during this round of the shootout, 0 either.

Table 3.2: Data collected for each round of the penalty shoot-out.

Field	Description
time	The time at which a substitution took place
player_in	The name of the player going on the field
player_in_info	Additional information on the player going on the field
player_out	The name of the player going off the field
player_out_info	Additional information on the player going off the field

Table 3.3: Data collected for each substitution of the match.

Field	Description
time	The time at which the card was given
player	The name of the player who received the card
player_info	Additional information on the reason the card was given

Table 3.4: Data collected for each yellow card of the match.

Field	**Description**
time	The time at which a goal took place
player	The name of the player who scored
info	Available complementary information on the goal (penalty, etc.)
assist	The name of the player who gave an assist (can be none)

Table 3.5: Data collected for each goal.

3.3 Building the dataframes in R

Finally, we can exploit the raw data and create a data set to perform our analysis. This process is depicted in Figure 3.3. To create the dataset, we used the programming language "R" and made another script to build it (cf. Appendix E). In "R", a dataframe is the fundamental data structure. It can be compared to a list of same sized vectors or as a matrix. The script works as follows:

1. First, it initialises a dataframe for the shots and another one for the general matches data.
2. After it looked through the working directory where we manually put all our json files and got their respective paths, it loads the content from the first one and extract the general data about the match.
3. The same extraction process is executed for each shot.
4. Then, they are loaded in the "shots" dataframe.
5. Finally, the general match data are loaded in the "match" dataframe.

Then if there are more paths, the same process is executed while if there are not, the process is terminated. The various fields of both dataframes are listed in Table 3.6 and Table 3.7.

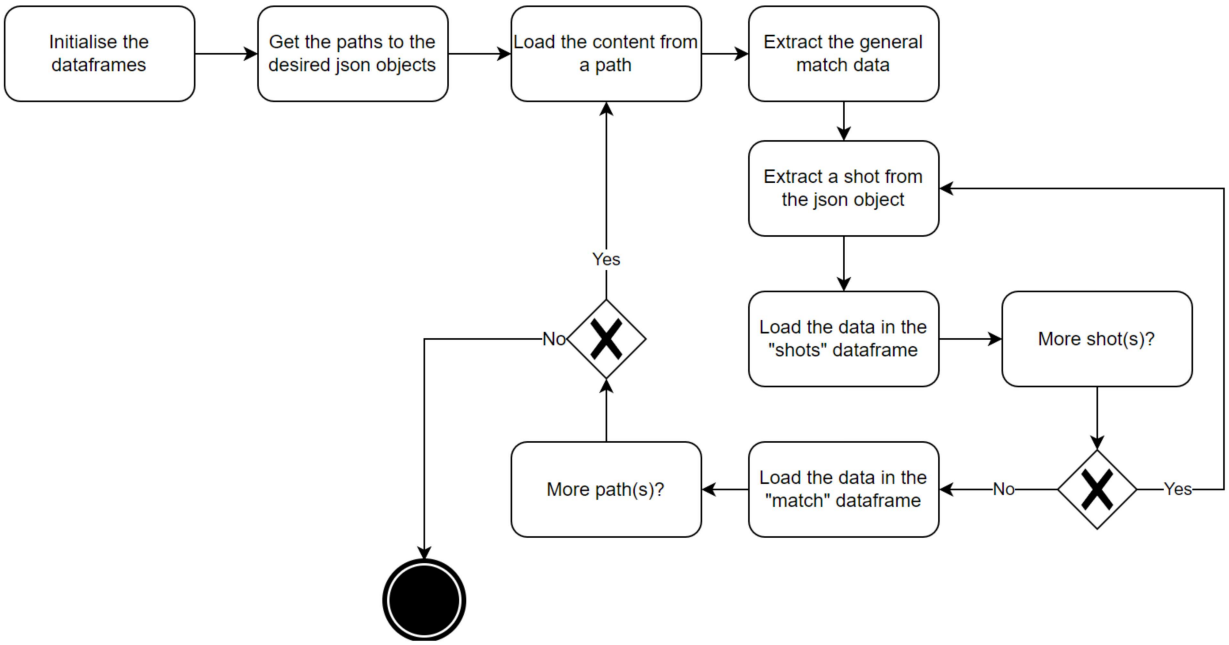


Figure 3.3: Process of downloading html code with match data

Field	Description
id	The identification of the match starting from 1
competition	The name of the competition in which the football match took place
date	The full date of the football match
year	The year of the football match
stage	The stage of football match in the competition (Final, last 16, etc.)
attendees	The estimated number of people attending the match
referee	The name of the referee
stadium	The name of the stadium where the match took place
team_A	The name of the team which took the first shot during the penalty shoot-out
team_B	The name of the team which took the second shot during the penalty shoot-out
winner	The name of the team who won the match
home_team	The team (A or B) which was playing at home (if relevant)
rounds	The number of rounds undertaken for the shoot-outs
shots	The number of shots taken during the shoot-outs
missed	The number of shots missed during the shoot-outs
rm_A	The score of the first leg match for team A (if relevant)
rm_B	The score of the first leg match for team B (if relevant)
ft_A	The score of the match before the shoot-outs for team A
ft_B	The score of the match before the shoot-outs for team B
so_A	The score of the shoot-outs for team A
so_B	The score of the shoot-outs for team B

Table 3.6: Structure of the data frame with match data.

Field	**Description**
match_id	The identification of the match in the matches dataset
round	The number of the round
so_A	The score of the shoot-outs for team A before the shot
so_B	The score of the shoot-outs for team B before the shot
team	The team taking the shot (A or B)
player	The name of the player taking the shot
scored	Either 1 if the shot was successful or 0
playtime	The amount of time the player was on the field during the match
goals	The number of goals the player scored during the match
yellows	The number of yellow cards the player received during the match (max 1)
pen	The number of penalties the player scored during the match
missed_pen	The number of penalties the player missed during the match
decisive	-1 if a missed shot concedes the victory to the other team, 1 if a successful shot concedes the victory of the shooting team, 0 else
advantage	True if the shooting team has a higher score than the other team
take	True if a successful shot would lead to taking an advantageous position

Table 3.7: Structure of the data frame with shots data.

Chapter 4

Analysis of the data set

Now that we found data on soccer matches, the next step consists in analysing said data. Indeed, as our main goal is to create a comparative simulator between shoot-out mechanisms, we need to corroborate, debunk or come up with the main drivers influencing the success of a shot. In this chapter, we present the dataset from a high level view and carry out multiple general analyses. Then, we replicate and compare experiments from previous researchers with our data set. Finally we present our predictive models.

4.1 General presentation of the dataset

After running the two scripts and extracting our data as described in the previous chapter, we ran simple analyses of the dataset in order to describe it.

A cleaning had to take place as we noticed inconsistencies in some of the competitions and some of the early years matches were incomplete. To perform that cleaning, we established a set of rules to take out matches and related shots without enough data or aberration in the logic.

1. First, we deleted the matches without any shots available in the extracted data (we only have general match data about the shoot-out).

2. Then, we deleted the ones were the balance in the number of shots was off such that the second kicking team had more shots than the first one

or the first kicking team had more than one shot ahead of the second one.

We managed to gather data on:

- 30 national and international competitions between clubs or nations.
- 3154 shoot-outs comprising of 21405 shots before cleaning.
- 2001 shoot-outs comprising of 20860 shots after cleaning.

As you can clearly see after the cleaning, the number of matches decreased drastically (-36.5%) while the number of shots decreased lightly (-2.5%). Table 4.1 shows the different competitions and their related numbers with:

- Type, either club or nation depending on the type of competition.
- Continent, the continent in which the competition took place (AFR for Africa, ASIA, EU for Europe, SAM for South-America and WORLD for international competitions).
- Competition, the name of the competition.
- Start, the earliest year we have data for a specific competition.
- End, the latest year we have data for a specific competition.
- n_0, the total number of matches before cleaning.
- n, the total number of matches after cleaning.
- ≤ 5, the total number of matches that ended up in 5 rounds or less.
- > 5, the total number of matches that ended up after 5 rounds.
- p, the success rate of a shot for the first shooting team.
- q, the success rate of a shot for the second shooting team.

- FSA, the first shooter advantage in terms winning rate for the first kicking team with associated p-value from a chi-squared test [1].
- FSA_0, the same calculation when taking into account the whole set of data before the cleaning (n_0)

[1]The Chi-squared test is an hypothesis test on a population variance based on the chi distribution. This test approximates the population variance using the sample variance[9]. It is used here to determine whether the variance of the sample fits the 50% distribution or is bigger.

Type	Continent	Competition	Start	End	n_0	n	Shots	≤ 5	>5	p	q	FSA	p-value	FSA_0	$p-value_0$
club	AFR	CAF-Champions League	1999	2021	50	20	209	15	5	0.731	0.762	0.45	0.588	0.5	0.5
club	ASIA	AFC Champions League	2007	2021	19	18	184	13	5	0.758	0.809	0.5	0.5	0.53	0.5
club	EU	European Champion Clubs' Cup	1970	1991	20	20	187	16	4	0.794	0.622	0.75	0.022**	0.75	0.022**
club	EU	UEFA Champions League	1996	2016	16	16	166	12	4	0.713	0.684	0.5	0.5	0.5	0.5
club	EU	Europa League	1972	2021	103	95	973	71	24	0.735	0.72	0.48	0.581	0.48	0.653
club	EU	UEFA Champions League Qualifying	1998	2021	30	28	315	17	11	0.714	0.636	0.61	0.172 .	0.6	0.181
club	EU	League Cup	1983	2016	164	136	1448	98	38	0.74	0.734	0.51	0.466	0.52	0.292
club	EU	EFL Cup	2016	2021	102	102	1096	76	26	0.734	0.792	0.44	0.862	0.44	0.862
club	EU	Copa del Rey	1975	2021	234	67	701	48	19	0.754	0.718	0.52	0.403	0.5	0.5
club	EU	Türkiye Kupasi	1990	2021	341	200	1888	152	48	0.857	0.854	0.5	0.472	0.56	0.02**
club	EU	KNVB Beker	1970	2021	258	117	1247	85	32	0.765	0.713	0.63	0.003***	0.55	0.06*
club	EU	Taça de Portugal Placard	2008	2021	286	73	787	51	22	0.734	0.717	0.55	0.241	0.49	0.616
club	EU	Italy Cup	1971	2021	200	188	1991	132	56	0.742	0.723	0.52	0.305	0.52	0.31
club	EU	Beker van België	1985	2021	112	73	799	47	26	0.74	0.721	0.55	0.241	0.5	0.5
club	EU	Amstel Cup	1998	2005	37	8	80	5	3	0.683	0.692	0.38	0.638	0.43	0.745
club	EU	Carlsberg Cup	2008	2008	10	1	10	1	0	0.6	0.4	1	0.5	0.5	0.5
club	EU	Allianz Cup	2009	2020	47	30	318	20	10	0.714	0.72	0.43	0.708	0.34	0.979
club	EU	Taça CTT	2017	2018	4	4	41	3	1	0.762	0.65	0.75	0.309	0.75	0.309
club	EU	UEFA Super Cup	2013	2021	3	3	34	2	1	0.941	0.765	1	0.124 .	1	0.124 .
club	EU	Coupe de France	1982	2022	383	167	1784	116	51	0.739	0.758	0.46	0.861	0.49	0.695
club	EU	Coupe de la Ligue	1995	2020	202	144	1526	100	44	0.733	0.746	0.47	0.773	0.45	0.93
club	EU	Trophée des Champions	1996	2012	7	6	82	1	5	0.683	0.732	0.33	0.658	0.29	0.775
club	EU	DFB-Pokal	1970	2022	253	242	2535	170	72	0.747	0.73	0.52	0.24	0.51	0.353
club	EU	DFL-Supercup	2011	2017	3	3	31	2	1	0.75	0.867	0.33	0.5	0.33	0.5
nation	EU	EURO	1976	2021	22	22	232	16	6	0.775	0.759	0.55	0.416	0.55	0.416
club	SAM	Copa Libertadores	1977	2021	93	74	733	61	13	0.749	0.692	0.61	0.041**	0.53	0.339
club	SAM	Copa Sudamericana	2002	2021	91	80	839	53	27	0.709	0.677	0.54	0.288	0.52	0.417
nation	SAM	Copa América	1993	2021	29	29	280	23	6	0.781	0.746	0.52	0.5	0.52	0.5
club	WORLD	Intercontinental Cup	1985	2004	5	5	65	3	2	0.697	0.688	0.6	0.5	0.6	0.5
nation	WORLD	World Cup	1982	2018	30	30	279	28	2	0.705	0.699	0.5	0.5	0.5	0.5
TOTAL			1970	2022	3154	2001	20860	1437	564	0.751	0.739	0.52	0.064	0.51	0.233

Table 4.1: Summary of the data set. Signification code: $< 0{,}1\%$ (****), $< 1\%$ (***), $< 5\%$ (**), $< 10\%$ (*), $< 20\%$ (.)

The aggregated calculation on an eventual first shooter advantage as described by Apesteguia and Palacios-Huerta [2] and others gives different results compared to our dataset. Indeed, we calculated a 52% winning rate for the first shooting team giving a 4% advantage compared to the second shooting one which is considerably lower than the 60% (20% advantage) described by Apesteguia and Palacios-Huerta [2], Palacios-Huerta [16], Silva *et al.* [19]. Our test results with null hypothesis of winning rates being equal to 50% is closer to the 53.3% found by Kocher *et al.* [13]. Although, ours is statistically significant at the 10% level. This is similar to Rudi *et al.* [18] which found a significant winning rate of 55% at the 5% level.

4.1.1 Impact of cleaning

Concerning the results related to our data before the cleaning (n_0) we can see four changes in significance of the first shooter advantage at the 5% and 10% levels respectively "Türkiye Kupasi" and "KNVB Beker" (that was previously significant at 5%) while the "Copa Libertadores" and the "UEFA Champions League Qualifying" are no longer significant.

4.1.2 Competitions

When looking more closely into the different competitions, we can see that only three out of thirty ("European Champion Clubs' Cup", "KNVB Beker", "Copa Libertadores") are shown to give a significant advantage to the first shooting team at the 5% level. For the 10% and 20% levels we have respectively zero and two significant numbers. Furthermore, when looking at winning rates for the first shooting team we noticed that for nine competitions it looks like the other team has a higher winning rate.

4.1.3 Evolution of winning rates through years

By computing the evolution of winning rates for A through the years in Figure 4.1, we can intuitively see that this rate is mostly above 50% before the years 2000 while it looks to be oscillating around it afterwards. However,

this variation gets closer to 50% as the number of matches increases. We will look into a possible reason as to why our results differ in relation to time.

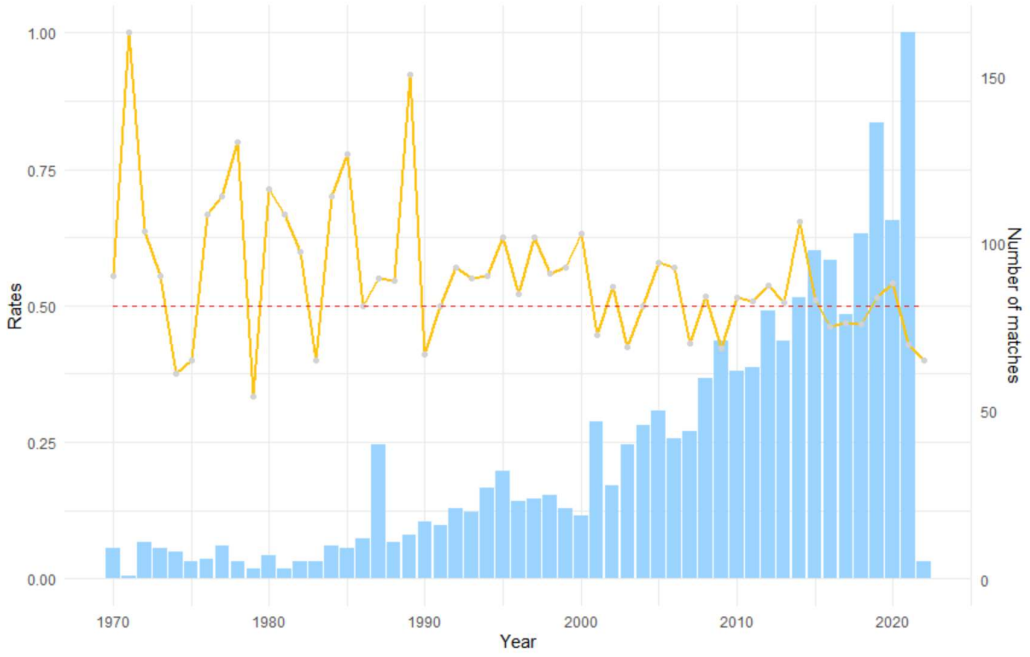


Figure 4.1: Winning rates for the first shooting team through the different years

4.1.4 Scoring rates

Looking at analyses from Apesteguia and Palacios-Huerta [2] and Brams *et al.* [5] on the success rates of shots for the first and second shooting team (noted respectively p and q) we can see that p (75,1%) is close to 3/4. However, q (73,9%) is quite far from 2/3. This is not consistent with their results. By visualizing the different values of p and q for each round in Figure 4.2, we cannot intuitively say that one of those probabilities is different or bigger than the other. The barchart indicates the number of shots available for the different rounds.

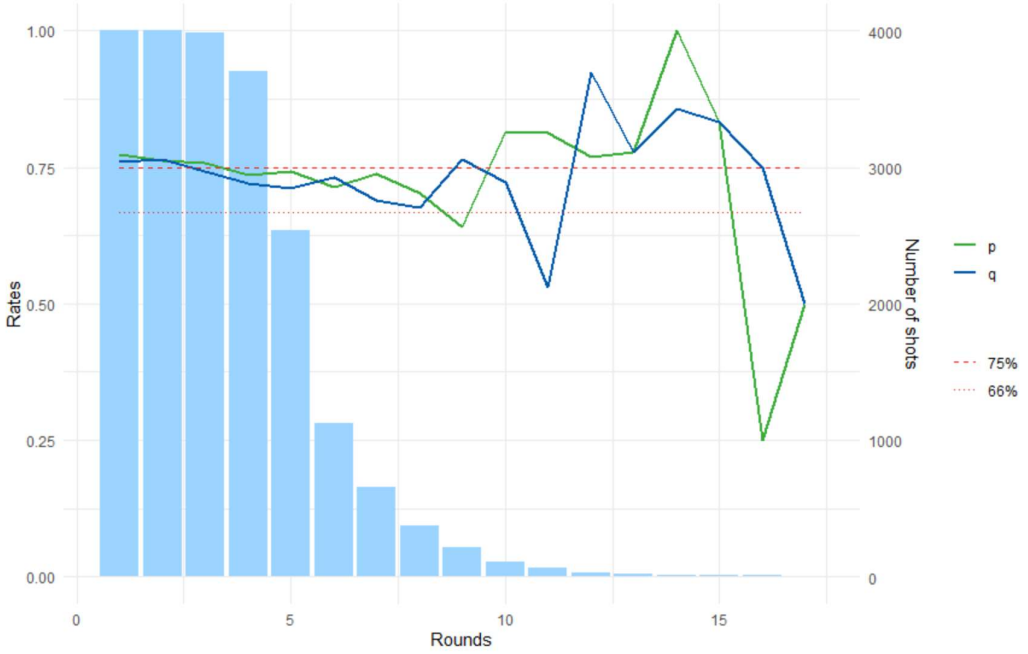


Figure 4.2: Evolution of scoring probabilities for both team across rounds

4.1.5 Data per type, continent and stage

Investigating more on the data per type and continent as per Table 4.2, we found only two significant first shooter advantage for the aggregated club results (52% rate with p-value < 0.069) and for the clubs in South America (57% rate with p-value < 0.041).

In Table 4.3 we aggregated the data per tournament stage and only kept the ones with more than thirty matches before cleaning. However, we did not find any first shooter advantage at a lower level than 20% for "Quarter-Finals", "Final" and "First-Round".

Type	**Continent**	**Start**	**End**	n_0	n	**Shots**	≤ 5	>5	p	q	FSA	$p-value$	FSA_0	$p-value_0$
nation		1976	2021	81	81	791	67	14	0.752	0.734	0.52	0.412	0.52	0.412
club		1970	2022	3073	1920	20069	1370	550	0.751	0.739	0.52	0.069*	0.51	0.247
	WORLD	1982	2018	35	35	344	31	4	0.704	0.697	0.51	0.5	0.51	0.5
	SAM	1977	2021	213	183	1852	137	46	0.736	0.693	0.56	0.052	0.52	0.292
	EU	1970	2022	2837	1745	18271	1241	504	0.754	0.744	0.51	0.135	0.51	0.287
	ASIA	2007	2021	19	18	184	13	5	0.758	0.809	0.5	0.5	0.53	0.5
	AFR	1999	2021	50	20	209	15	5	0.731	0.762	0.45	0.588	0.5	0.5
nation	WORLD	1982	2018	30	30	279	28	2	0.705	0.699	0.5	0.5	0.5	0.5
club	WORLD	1985	2004	5	5	65	3	2	0.697	0.688	0.6	0.5	0.6	0.5
nation	SAM	1993	2021	29	29	280	23	6	0.781	0.746	0.52	0.5	0.52	0.5
club	SAM	1977	2021	184	154	1572	114	40	0.728	0.684	0.57	0.045**	0.52	0.303
nation	EU	1976	2021	22	22	232	16	6	0.775	0.759	0.55	0.416	0.55	0.416
club	EU	1970	2022	2815	1723	18039	1225	498	0.754	0.743	0.51	0.145	0.51	0.299
club	ASIA	2007	2021	19	18	184	13	5	0.758	0.809	0.5	0.5	0.53	0.5
club	AFR	1999	2021	50	20	209	15	5	0.731	0.762	0.45	0.588	0.5	0.5

Table 4.2: Data set analysis per type and continent. Signification code: $<$ 0,1% (****), $<$ 1% (***), $<$ 5% (**), $<$ 10% (*), $<$ 20% (.)

Stage	**Start**	**End**	n_0	n	**Shots**	≤ 5	>5	p	q	FSA	$p-value$	FSA_0	$p-value_0$
Semi-Finals	1972	2021	120	93	1007	62	31	0.74	0.727	0.51	0.5	0.51	0.464
Quarter-Finals	1972	2021	195	168	1748	121	47	0.749	0.726	0.54	0.158 .	0.53	0.195 .
Round of 16	1972	2022	243	181	1897	132	49	0.726	0.732	0.5	0.5	0.49	0.551
Final	1971	2021	92	87	965	56	31	0.749	0.717	0.55	0.196 .	0.52	0.377
last 16	1970	2021	104	84	861	63	21	0.749	0.701	0.54	0.293	0.48	0.616
First Round	1970	2021	859	517	5394	371	146	0.759	0.75	0.52	0.146 .	0.51	0.247
Second Round	1970	2021	732	412	4240	298	114	0.769	0.743	0.52	0.201	0.51	0.241
Third Round	1970	2021	343	197	2041	148	49	0.743	0.771	0.48	0.665	0.5	0.5
Fifth Round	1990	2021	36	15	157	9	6	0.812	0.714	0.67	1	0.56	0.309
Fourth Round	1991	2022	118	72	744	49	23	0.731	0.745	0.49	0.547	0.48	0.609
Second Round Replay	1985	2014	40	11	117	7	4	0.767	0.632	0.64	1	0.48	0.563
First Round Replay	1970	2013	43	7	64	6	1	0.719	0.781	0.29	1	0.42	0.82
Sixth Round	1992	2019	52	29	341	16	13	0.711	0.738	0.48	1	0.46	0.661

Table 4.3: Data set analysis per stage. Signification code: $<$ 0,1% (****), $<$ 1% (***), $<$ 5% (**), $<$ 10% (*), $<$ 20% (.)

4.1.6 Median and Mean

Having a deeper look at the median and mean for the shots taken during a penalty shoot-out in Table 4.4 we can see that the median number of shots for our dataset is 10 while the mean is at 10.42. This could seem weird when knowing that 28% of the matches from our dataset reached the second phase of the penalty shoot-out. Thus, shooting more than ten times. However, one should consider that some penalty shoot-outs ended up during phase 1 (within the first five rounds) before shooting ten times or go over 10 shots in phase 2 (after five rounds). Based on those numbers we can assume that phase 2 goers are well above 10 shots. Those results seem to be consistent throughout all the competition as we can see in Figure 4.3 with the Box and Whisker plots[2] of all the respective shots taken.

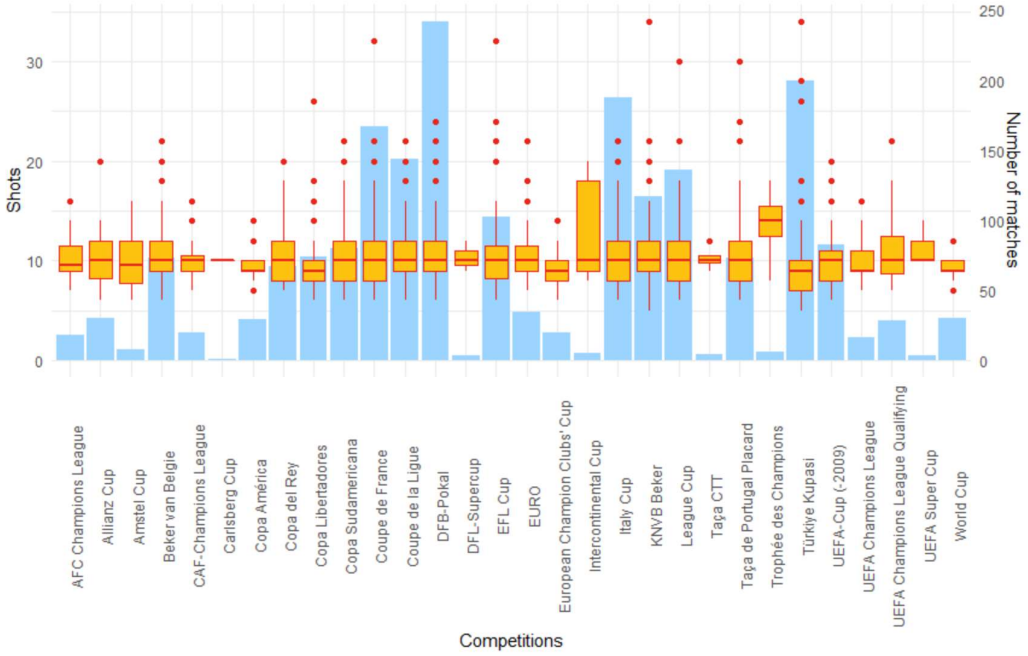


Figure 4.3: Box and Whisker plot of shots taken for each competition

With Tables 4.5 and 4.6 we computed those same number when separat-

[2]The Box and Whisker plots allows us to visualise the spread of a dataset. The borders indicate the first and third quartiles while the central line indicates the median. The lines indicate the minimum and the maximum while the dots represent single observations outside the calculation (outliers) [9].

ing the analysis between matches that reached the second phase of penalty shoot-outs and the ones that did not. We can see that the scored rate for phase 2 goers is higher than the other ones (75,21% > 73.77%) implying that the successfulness of shots for shoot-outs consisting of more rounds (ergo shots) is higher. This is consistent with the fact that successful shooters are more likely to make the game last. However, such a small difference (1.44%) deserves to be analysed deeper. Carrying a student t-test on both means, we found that the difference between phase 1 and phase 2 goers is significant at the 5% level (p-value = 0.015). This means that the scoring rate is significantly lower for the matches that end up before five rounds.

General	**Shots**	**Scored**	**Missed**
Median	10.00	7.00	3.00
Mean	10.42	7.77	2.65

Table 4.4: Mean and Median analysis for all shots taken whether they were scored or missed

Phase 1	**Shots**	**Scored**	**Missed**
Median	9.00	6.00	2.00
Mean	8.76	6.47	2.29

Table 4.5: Mean and Median analysis for all shots taken whether they were scored or missed in Phase 1.

Phase 2	**Shots**	**Scored**	**Missed**
Median	14.00	11.00	3.00
Mean	14.65	11.07	3.58

Table 4.6: Mean and Median analysis for all shots taken whether they were scored or missed in Phase 2.

4.2 Comparison with previous researchers

As discussed previously, our results quite differ from Apesteguia and Palacios-Huerta [2], Palacios-Huerta [16] and Silva *et al.* [19] who found respectively 60.5%, 60.6% and 59,48% winning rate for the first shooting team.

4.2.1 Before and after 2003

In order to run a deeper analysis to get another view on such rates, we separated our dataset between pre and post 2003 matches known as the IFAB switch between 2002 and 2003 determining that:

> "The referee tosses a coin and the team whose captain wins the toss decides whether to take the first or the second kick."[12, p. 52].

Previously, the toss winning team was deemed to kick first. As shown in table 4.7, we can see that numbers for pre-2003 matches are significantly indicating a first shooter advantage of 57% at the 1% level. However, we can notice that this advantage decreases by 5% as well as its significance (10% level) when taking into account the whole set of data before cleaning.

Interestingly, the results for post 2003 matches have no significance in showing that the winning rate for the first kicking team is bigger than 50%. However, they are showing a 50/50 chance of winning based on empirical data (although not statistically significant). It is closer to the interpretation of Silva *et al.* [19] who found a 53.3% advantage with a p-value of 0.13.

	Start	**End**	n	**Shots**	**≤5**	**>5**	**FSA**	**p-value**
n	1970	2002	502	5173	370	132	0.57	0.001****
n	2003	2022	1499	15687	1067	432	0.5	0.5
n_0	1970	2002	956	5360	823	133	0.52	0.07**
n_0	2003	2022	2198	16045	1757	441	0.5	0.53

Table 4.7: Comparison of a first shooter advantage before and after 2003

While this behaviour has not been explained in the reviewed literature, one can assume that coaches have been made aware of a potential first shooter advantage and its psychological roots thus, preparing their players accordingly. Indeed, the studies presented in the first chapter were trying to find out the existence of an unfair advantage and suggesting fairer mechanisms. Thus, readers could take on their deep statistical analyses as well as the psychological and strategical ones. However, those soccer specific papers were only published later than 2003 (starting from Apesteguia and Palacios-Huerta [2]

in 2008). So, we can only assume that coaches started to prepare the players psychologically when getting acquainted to this new rule. Indeed as a choice needs to be made there could possibly be a better one to make. Furthermore, as there is a better choice to make it is important to mitigate the negative impact of playing in the worst condition (going second). But those are only assumptions at attempting to explain this gap in significance in the data set and it deserves to be deeper analysed in another research.

4.2.2 The various shooting scenarios

Concerning studies focused on the different kick scenarios and their related scoring probabilities, Arrondel *et al.* [3] did not find a first shooter advantage in the analysed French competition. This is quite consistent with our results in Table 4.1:

- The "Coupe de France", FSA: 46%
- The "Coupe de la Ligue", FSA: 47%
- The "Trophée des champions", FSA: 33%

They all did not show any significant advantage for the first shooting team. It was even quite the opposite with winning rates for the first shooting team all lower than 50%. However, like Rudi *et al.* [18] they both analysed the probability of scoring a kick depending on the relative score position between both teams. Arrondel *et al.* [3] established three different kicks respectively:

- The kicks taken when leading the score with the same number of shots (70.8% scoring rate).
- The kicks taken in the opposite situation (70.2% scoring rate).
- The kicks allowing to equalize the score (83.6% scoring rate).

Looking at the technique deployed by Rudi *et al.* [18], we can see that they even broadened this view by defining the different situations as score differences between -2 and 2. As shown in Table 4.8, we applied this classification

to the first phase of the shoot-out on our dataset and showed the scoring rate in those different situations. We can see that compared to Arrondel *et al.* [3] our numbers differ as well from their study as catch-up kicks don't seem to benefit from a higher scoring probability.

Team/s	**-2**	**-1**	**0**	**1**	**2**
A	0.754	0.736	0.772	0.719	0.744
B	0.722	0.759	0.726	0.714	0.765
Total	0.727	0.753	0.758	0.718	0.745

Table 4.8: Computing the results from the analysis of Rudi *et al.* [18] using our data set. Scoring rate of shots for both teams in different situations.

However, when looking at the total in Table 4.8 we can see that the aggregated rates in disadvantageous situation are closer to B while the advantageous ones are closer to A. The p-value are computed using a z-test[3]. Realising a proportion test as shown in Table 4.9 we can see that indeed the first shooting team is significantly more often in an advantageous situation. This means that if factors such as pressure and stake as described by Arrondel *et al.* [3] do indeed exist then, the second shooting team is significantly more submitted to it. As demonstrated by Vandebroek *et al.* [23], it could be the proof of the existence of a first shooter advantage.

s	**-2**	**-1**	**0**	**1**	**2**
A	0.149	0.243	0.7	0.785	0.921
p-value (z-test)	$\approx 0^{****}$	$\approx 0^{****}$	$\approx 0^{****}$	$\approx 0^{****}$	$\approx 0^{****}$

Table 4.9: Computing the proportion for all situations described by Rudi *et al.*

When replicating Figure 2.6 from Rudi *et al.* [18] in Table 4.10 we can see that while our scoring rates are actually close. We only have significance in the advantage at the 5% level for the first shooting team at round five using a student t-test[4] (p-value of 0.0381 for an advantage of 3.1%). This is quite far from the significance of the results of Rudi *et al.* [18] as from round 3.

[3]A z-test is similar to a t-test. However, the z-test uses the normal distribution and not the student one [9].

[4]In a t-test, the student t-distribution is used. This set of tests is performed for small sample size with an unknown standard deviation.[9]

Round	**1**	**2**	**3**	**4**	**5**	**6+**	**Shots**
A	0.775	0.762	0.758	0.736	0.743	0.721	10689
B	0.761	0.765	0.742	0.72	0.712	0.713	10171
p-value	0.1474	0.5882	0.123	0.1359	0.0381**	0.3307	

Table 4.10: Replicating Figure 2.6 from Rudi *et al.*[18]. Scoring rates within different rounds for each team. Signification code: $< 0{,}1\%$ (****), $< 1\%$ (***), $< 5\%$ (**), $< 10\%$ (*)

4.3 Influencing factors analysis

To determine better which factors influence the scoring probability of a kick, we built multiple regression models. As shown in Table 4.11, on top of the extracted data we computed new indicators to determine empirically what influenced the scoring probability of a shot during a penalty shoot-out:

- so_A and so_B: The score of each team before a shot,
- advantage: True or False if the shooting team is leading to the score before shooting,
- take: 1 or 0 if the shooting team can take the advantage with a successful shot,
- diff: the difference in score between the shooting team and the other one before taking the shot.

As we can see in Table 4.11, we tested many different factors:

- The variables from the data set,
- The ground (home, away or neutral),
- The type (club or nation),
- The competition,
- The stage,
- The continent.

The various competition did not seem to have a specific impact on the scoring probability. However, we do have a significant impact for three competitions: "Carlsberg Cup", "Copa Sudamericana" and "UEFA Champions League Qualifying". The different stages also did not show a significant impact although "Third Place Play-Off" did show significance at the 10% level. However, for both competition and stage we cannot say that there is an impact on the scoring probability. Interestingly enough the ground of the shoot-out (home, away or neutral) did not appear to have a significant impact on the scoring probability. Consistently with Arrondel *et al.*[3] we can see that a decisive shot has an impact on the outcome of the match. It needs to be separated between both scenarios: decisive shot leading to the team winning or losing. The following factors did not seem to have an influence on the scoring probability:

- The playtime,
- The number of goals a team scored during the match,
- The number of yellow cards they received,
- The number of penalty scored or missed during the game,
- The respective scores of both teams.

x	**Logit**	**Logit**	**Logit**	**Probit**	**Probit**	**Probit**
(Intercept)	1,61E+00***	1,19E+00****	1,26E+00****	9,71E-01***	7,28E-01****	7,70E-01****
round	-4,18E-02	-2,80E-02***	-3,35E-02****	-2,35E-02	-1,68E-02***	-1,99E-02****
teamB	-9,12E-02*	-5,15E-02	-7,21E-02**	-5,48E-02*	-3,06E-02	-4,26E-02**
so_A	3,69E-02			2,24E-02		
so_B	-2,25E-02			-1,34E-02		
playtime	-2,22E-02			-1,28E-02		
goals	8,32E-02			4,92E-02		
yellows	4,90E-02			2,95E-02		
pen	-1,54E-01			-9,23E-02		
missed_pen	-1,82E-01			-1,10E-01		
decisive	9,21E-02*			5,60E-02*		
decisiveLOSE		-1,00E-01	-1,14E-01**		-6,00E-02	-6,79E-02**
decisiveWIN		5,35E-02			3,17E-02	
advantageTRUE	-3,65E-01**	-1,57E-01**	-1,85E-01****	-2,18E-01**	-9,34E-02**	-1,09E-01****
takeTRUE	-1,13E-01			-6,79E-02		
attendees	-6,48E-06	-1,56E-06*		-3,96E-07*	-9,25E-07	

diff	8,17E-02	4,88E-02
groundhome	-8,60E-03	-5,01E-03
groundneutral	-4,09E-01	-2,38E-01
typenation	NA	NA
competitionAllianz Cup	-4,03E-01	-2,36E-01
competitionAmstel Cup	-4,97E-01	-2,93E-01
competitionBeker van Belgie	-2,32E-01	-1,33E-01
competitionCAF-Champions League	-4,61E-01	-2,71E-01
competitionCarlsberg Cup	-1,31E+00*	-7,93E-01*
competitionCopa América	2,47E-01	1,46E-01
competitionCopa del Rey	-2,72E-01	-1,57E-01
competitionCopa Libertadores	-3,98E-01	-2,32E-01
competitionCopa Sudamericana	-4,35E-01*	-2,55E-01*
competitionCoupe de France	-2,52E-01	-1,45E-01
competitionCoupe de la Ligue	-2,49E-01	-1,43E-01
competitionDFB-Pokal	-2,55E-01	-1,46E-01
competitionDFL-Supercup	7,85E-02	4,97E-02
competitionEFL Cup	-1,52E-01	-8,78E-02
competitionEURO	2,38E-01	1,43E-01
competitionEuropa League	-3,21E-01	-1,86E-01
competitionEuropean Champion Clubs' Cup	-4,02E-01	-2,35E-01
competitionIntercontinental Cup	-4,76E-01	-2,81E-01
competitionItaly Cup	-3,10E-01	-1,80E-01
competitionKNVB Beker	-2,33E-01	-1,34E-01
competitionLeague Cup	-2,55E-01	-1,47E-01
competitionTaça CTT	-4,56E-01	-2,65E-01
competitionTaça de Portugal Placard	-2,92E-01	-1,69E-01
competitionTrophée des Champions	-4,34E-01	-2,54E-01
competitionTürkiye Kupasi	-4,35E-02	-2,44E-02

competitionUEFA Champions League	-4,57E-01	-2,68E-01
competitionUEFA Champions League Qualifying	-5,83E-01**	-3,45E-01**
competitionUEFA Super Cup	3,91E-01	2,26E-01
competitionWorld Cup	NA	NA
stage3rd round	-1,66E-01	-9,82E-02
stage4th round	2,55E-02	2,21E-02
stage4th round deciders	-2,15E-01	-1,26E-01
stage5th round	-5,91E-01	-3,54E-01
stageFifth Round	-1,75E-01	-9,68E-02
stagefinal	-2,14E-01	-1,27E-01
stageFinal	3,54E-02	2,46E-02
stagefinal decider	1,26E+00	7,15E-01
stageFirst Preliminary Round	-2,88E-01	-1,66E-01
stageFirst Round	2,01E-02	1,56E-02
stageFirst Round Replay	-8,93E-02	-4,88E-02
stageFourth Round	-3,05E-02	-1,43E-02
stageGroup 2	3,53E-01	2,08E-01
stageGroup 5	-6,65E-01	-4,01E-01
stageintermediate stage	8,49E-02	5,59E-02
stagelast 16	-5,38E-04	4,71E-03
stageQualifying Round	1,65E-01	1,04E-01
stageQuarter-Finals	-3,05E-02	-1,49E-02
stageRound of 16	-6,96E-02	-3,84E-02
stageRound of 16 Replay	-5,76E-01	-3,45E-01
stageSecond Round	-1,96E-02	-7,82E-03
stageSecond Round Replay	-3,31E-01	-1,96E-01
stageSemi-Finals	-2,63E-02	-1,19E-02
stagesemi-finals deciders	5,23E-02	3,44E-02
stageSixth Round	-1,80E-01	-1,03E-01
stageThird Place Play-Off	1,43E+00*	7,70E-01*
stageThird Round	-1,13E-01	-6,50E-02
stageThird Round Replay	4,13E-02	2,68E-02
continentASIA	NA	NA
continentEU	NA	NA
continentSAM	NA	NA
continentWORLD	NA	NA

Table 4.11: Scored shots prediction models. Signification code: < 0,1% (****), < 1% (***), < 5% (**), < 10% (*)

After testing different combinations of factors, we came up with two models that seemed to be the most consistent with our previous analyses. We kept four different factors to create those models:

- round, gives the round number. Here we can see that it has a negative impact on the scoring probability which is consistent with previous results from Rudi *et al.*[17].
- teamb, is equal to one when the team shooting is B (thus second in the round). It shows us that for all models there is a negative impact on the scoring probability.
- decisivelose, indicates if a failed shot would concede the victory to the opponent. Here we can see that significance rises when taking away less significant variables.
- advantagetrue, indicates if a team has a higher score than its opponent before shooting and we can see that it has a negative impact in both models.

For the logit model we have:

$$L_i = 1.26 - 0.028round - 0.0721teamB - 0.114decisiveLOSE - 0.185advantageTRUE$$

For the probit model we have:

$$P_i = 0.77 - 0.0199round - 0.0426teamB - 0.0679decisiveLOSE - 0.109advantageTRUE$$

For both models and various factor combination we have the following success probability p for each shot:

$$p = \frac{1}{1+e^{-(model)}}$$

As we can see in Table 4.12, the proportion of B in a "decisiveLOSE" position if 62% bigger than A. Besides the fact that going second is disadvantageous in itself, there are other reasons as to why we noticed a first shooter advantage. Indeed, if the second shooting team finds itself more often in situations with a negative impact on the scoring probability then it is also a source for a potential disadvantage.

Factor	**A**	**B**
Shots	51%	49%
DecisiveWIN	34%	66%
DecisiveLOSE	19%	81%
Advantage	80%	20%

Table 4.12: Proportions of both team for each factor

Chapter 5

Simulation

Now that we have all the information we need to reach our goal, we can finally create a simulator (Figure 5.1) to compare the following shoot-out mechanisms:

- Current rule or ABAB (Figure 5.2);
- Tennis tie-break or ABBA (Figure 5.3);
- The Catch-Up Rule from Brams and Ismail [5] or CU (Figure 5.4);
- The Adjusted Catch-Up from Csató [6] or ACU (Figure 5.5).

After selecting our two models, we created a simulator using R programming language (cf. Appendix E). In total, we ran eight simulations consisting of 10000 matches each (4 mechanisms and 2 prediction models). The simulation works as such:

1. First, we create a football match which receives as input the shot prediction model and the shoot-out mechanism we chose.
2. Then, we create a round to keep track of the round number and have each team shooting exactly once every time.
3. Finally, we generate a shot. This includes the whole context of the shot (advantage, decisive lose and of course the team). However, here

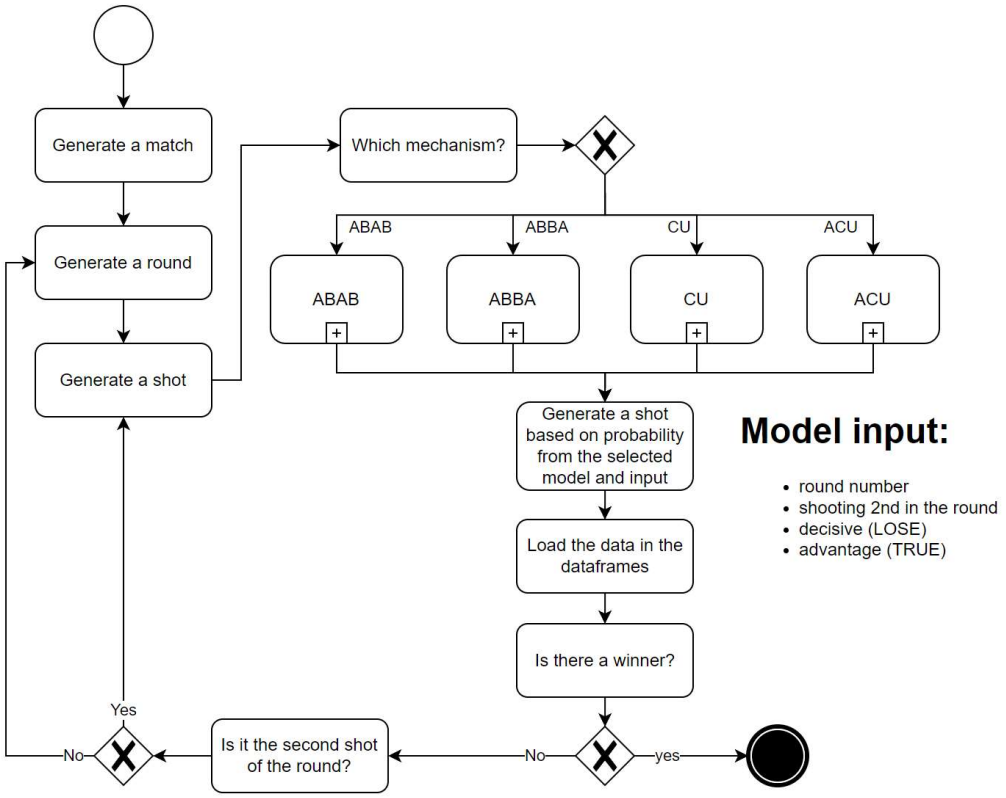


Figure 5.1: The simulator

we consider that the "team" factor gives the position of the shooting team inside the round (first or second). So, when team A is shooting second in a round we generate the probability of scoring a shot as if it was team B shooting. This has a negative impact on the scoring probability as shown with the models in the previous chapter.

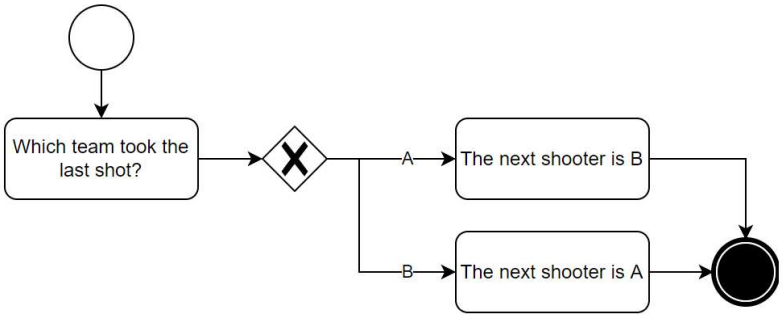


Figure 5.2: The current (ABAB) mechanism

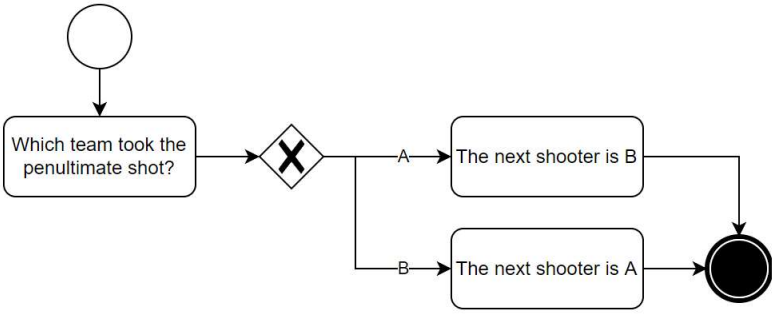


Figure 5.3: The tennis tie-break (ABBA) mechanism

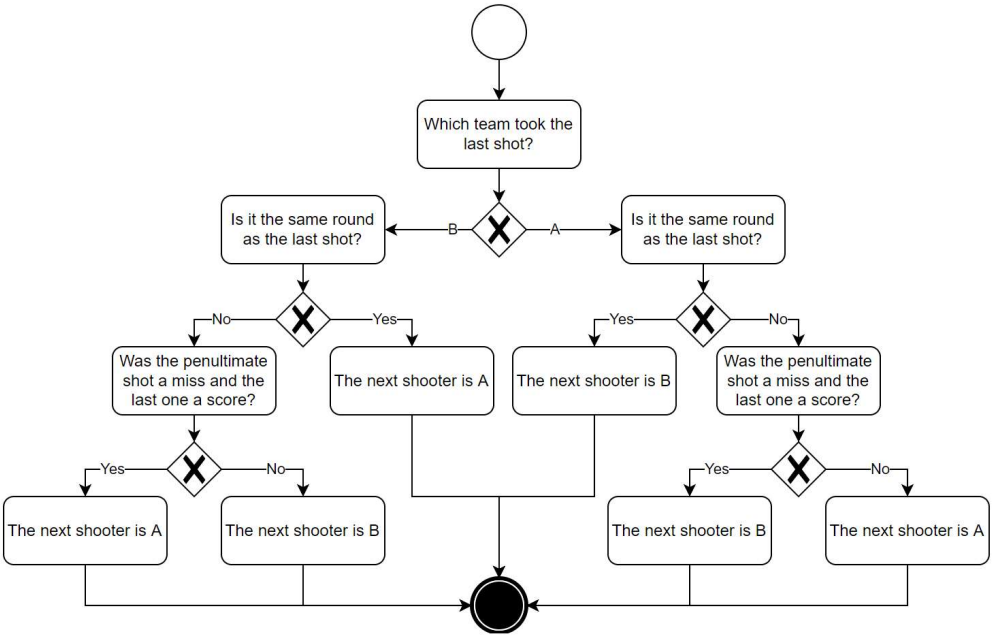


Figure 5.4: The Catch-Up (CU) mechanism

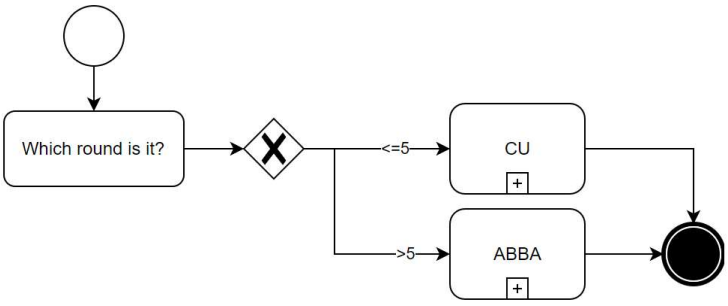


Figure 5.5: The Adjusted Catch-Up (ACU) mechanism

In Table 5.1, we can see the results from the simulations. Interestingly the numbers for the ABAB mechanism gives similar results to our empirical data. All three other mechanisms give fairer match results than the current one. By computing the mean of the winning rate of the first shooting team for ABBA, CU and ACU considering both models, we find respectively 50.43%, 50.47% and 50.49%. Thus, even if the difference seems small, ABBA appears to offer the fairest mechanism.

We can also have a look at the proportions of the first shooting team finding itself in situations having a negative influence. For ABAB, A was mostly in the "advantage" position but not in the "decisive LOSE" one. As we can see in the previous chapter, the models show a bigger negative impact for "decisive LOSE". Thus, advantaging A in the scoring probability ($p > q$). This difference seems to get mitigated with the other mechanisms while still giving an slight advantage for A.

Consistently with previous researcher and our dataset we found that 29% of the matches went for over than 5 rounds in Phase 2. Furthermore, when looking at the winning rates of A when separating the simulations between phase 1 and 2, we can say that all mechanisms are fairer than the current ABAB. However, the CU in the logit and the ACU in the probit are a bit further from 50% than the others (ABAB excluded). Tables 5.2 and 5.3 show the different proportion of each factor respectively for Phase 1 and Phase 2 of the shoot-outs. We can see that while A still shoots first in more round than B in Phase 1, the opposite happens in Phase 2. Consistently with our research and the models, p and q are lower during Phase 2.

We can conclude that mechanisms proposed by other researchers are indeed fairer when using our Markov chain. However, unlike Csató, we do not find the ACU to be specifically fairer than the ABBA or the CU. Nonetheless, the ACU seems to give fairer proportions for the "decisive LOSE" situation. Furthermore, following studies from Csató and Anbarcı *et al.* the order of complexity for the different mechanisms is: ABAB, ABBA, CU and ACU. So, it might not be worth to look into more complex mechanisms when ABBA seems to be giving fairer results.

Mechanism	**Model**	**A**	**p**	**q**	**A decisive - Lose**	**A advantage**	**A first**	**Phase 1**	**Phase 2**	**Phase 1 - A**	**Phase 2 - A**
ABAB	logit	52,92%	0,752	0,739	0,215	0,815	1	7148	2852	52,1%	55%
ABBA	logit	49,97%	0,743	0,743	0,554	0,544	0,557	7041	2959	50,6%	48,5%
CU	logit	50,83%	0,749	0,744	0,471	0,512	0,554	7001	2999	51,4%	49,4%
ACU	logit	49,77%	0,744	0,754	0,506	0,504	0,544	7096	2904	49,8%	49,7%
ABAB	probit	51,95%	0,752	0,741	0,211	0,819	1	6991	3009	51,6%	52,8%
ABBA	probit	50,89%	0,749	0,744	0,549	0,556	0,557	7005	2995	52,2%	47,8%
CU	probit	50,11%	0,745	0,745	0,479	0,508	0,556	6992	3008	50%	50,3%
ACU	probit	51,22%	0,751	0,744	0,501	0,529	0,544	7095	2905	52,19%	48,8%

Table 5.1: Summary of factors for the different simulations.

Mechanism	**Model**	**A**	**p**	**q**	**A decisive - Lose**	**A advantage**	**A first**
ABAB	logit	52,08%	75,38%	74,51%	34,76%	81,45%	100,00%
ABBA	logit	50,60%	74,84%	74,66%	50,22%	54,38%	58,51%
CU	logit	51,45%	75,36%	74,77%	41,72%	51,24%	56,96%
ACU	logit	49,79%	74,82%	74,93%	42,25%	50,37%	56,98%
ABAB	probit	51,59%	75,45%	74,64%	34,53%	81,88%	100,00%
ABBA	probit	52,19%	75,42%	74,58%	49,29%	55,62%	58,57%
CU	probit	50,01%	74,85%	74,88%	43,32%	50,79%	57,07%
ACU	probit	52,19%	75,68%	74,70%	41,13%	52,87%	56,90%

Table 5.2: Summary of factors for the different simulations in Phase 1.

Mechanism	**Model**	**A**	**p**	**q**	**A decisive - Lose**	**A advantage**	**A first**
ABAB	logit	55,01%	73,84%	69,65%	NA	NA	100,00%
ABBA	logit	48,46%	70,83%	72,13%	63,59%	NA	44,42%
CU	logit	49,38%	71,70%	72,20%	55,03%	NA	48,04%
ACU	logit	49,72%	71,72%	71,95%	62,85%	NA	44,36%
ABAB	probit	52,78%	73,38%	71,07%	NA	NA	100,00%
ABBA	probit	47,85%	71,24%	73,00%	63,49%	NA	44,38%
CU	probit	50,33%	72,00%	71,73%	54,44%	NA	48,05%
ACU	probit	48,85%	71,61%	72,54%	62,82%	NA	44,62%

Table 5.3: Summary of factors for the different simulations in Phase 2.

Chapter 6

Conclusion and further research

The literature review showed us many approaches and reasons as to why football shoot-outs offered such an extraordinary opportunity of analysing human behaviour. However, as various researchers looked into the matter what was supposed to be an economical and psychological study became highly statistical. Indeed, many researchers looked at the initial findings of Apesteguia and Palacios-Huerta [2] and saw a biased study that required fixing. Others saw in the various studies a deeply unfair mechanism that needed to be corrected. Those researchers suggested alternative mechanisms to the "Fédération internationale de football association" (FIFA). However, the FIFA already tried the tennis tie-break mechanism (ABBA) and deemed it too complex in 2017. Nonetheless, researchers still came up with fairer, more complex mechanisms and ways of evaluating their complexity. In order for us to make up a mind of our own, we built a tool that had for purpose to collect massive amount of penalty shoot-out data. We deemed that solution to be the most efficient considering the fact that we were limited in time and resources. In the end, we created a dataset of over 2000 matches which was bigger than any of the previous researchers. Our tool allows us to gather more data in the future should anybody think that it was necessary.

During the analysis of the dataset, we showed that we also found a first shooter advantage of 52%. However, it was smaller than the one found by previous researchers. When looking at this advantage, we saw that it was

getting closer to 50% as the year passed. However, it was correlated with the size of our data set as we had more matches in the recent years. While we did find a difference in scoring probability for both team, it was not as big as previously found by other researchers (3/4 and 2/3 vs 0.75 and 0.74 in our results). As our results quite differed from previous researchers we looked into the history of shoot-out rules. As there was a change in 2003, we separated our dataset before and after that year. Surprisingly, with 57% we found a similar first shooter advantage as Apesteguia and Palacios-Huerta [2], Palacios-Huerta [16] and Silva *et al.* [19]. However, we did not research the reason why there was such a big difference between those two time periods. We only assumed that football coaches decided to prepare psychologically their players in order to limit any psychological effect related to a specific situation. More, pre-2003 matches accounted for only 25% of our data set. Based on the potential psychological effect of a specific score difference as discussed by Arrondel *et al.* [3] and Rudi *et al.* [18] we looked at the proportion of both teams finding themselves in such situation. We found that mostly, the first shooting team found itself in advantageous situation while it was the opposite for the other team.

After analysing the data set, we tried to establish the various factors that had an influence on the scoring probability. We selected two models that only consisted of factors having a negative impact on said probability. Using those models we ran multiple simulations to compare the different values of the alternative mechanisms based on a chain of Markov. While not being drastically obvious, we found that the tennis tie-break mechanism offered the fairest alternative. Interestingly, this was the less complex alternative mechanism.

In order to pursue this work further, we could grow our data set even more. Indeed the more the data the more we can expect significant results. Furthermore, we had to clean the dataset using automated means. So, there is a possibility that the data is not completely correct compared to the events that really happened. There is a need for a qualitative analysis of the data extracted from *Transfermarkt*. However, we did not take into account factors such as the strategy of the shooter and the goalkeeper as introduced by

Palacios-Huerta [16]. It would be interesting to see how various factors could influence the strategy of players based on real-life data. Indeed, in their experiment Palacios-Huerta [16] showed that players were playing at the Nash equilibrium. But we could try to see if it is still the same under greater pressure or in different situations. In conclusion, we can say that the quality of data and probabilistic values is greatly important when trying to come up with an alternative mechanism. Indeed, we showed that the scoring rates were not as different as assumed by Csató [6]. Indeed, we showed that their analysis of the various mechanism was biased. Furthermore, we also showed that the scoring probability was influenced by the outcome of previous shots. Thus, their evaluation methods was not sufficient in showing the value of their proposed more complex mechanism (Adjusted Catch-Up).

Bibliography

[1] N. Anbarcı, C.-J. Sun, and M. U. Ünver, "Designing practical and fair sequential team contests: The case of penalty shootouts," en, *Games Econ. Behav.*, vol. 130, pp. 25–43, Nov. 2021.

[2] J. Apesteguia and I. Palacios-Huerta, "Psychological pressure in competitive environments: Evidence from a randomized natural experiment," en, *SSRN Electron. J.*, 2008.

[3] L. Arrondel, R. Duhautois, and J.-F. Laslier, "Decision under psychological pressure: The shooter's anxiety at the penalty kick," en, *J. Econ. Psychol.*, vol. 70, pp. 22–35, Jan. 2019.

[4] "Beautifulsoup." (Apr. 6, 2022), [Online]. Available: `https://beautiful-soup-4.readthedocs.io/en/latest/#`.

[5] S. J. Brams and M. S. Ismail, "Making the rules of sports fairer," en, *SSRN Electron. J.*, 2016.

[6] L. Csató, "A comparison of penalty shootout designs in soccer," en, *4OR*, Apr. 2020.

[7] "Fbref." (), [Online]. Available: `https://fbref.com/en/`.

[8] "Fotmob." (), [Online]. Available: `https://www.fotmob.com/`.

[9] J. Freeman, E. Shoesmith, D. Sweeney, D. Anderson, and T. A. Williams, *Statistics for Business and Economics*, 3rd ed. Andover, England: Cengage Learning EMEA, Feb. 2014.

[10] M. Hollander, "A nonparametric test for bivariate symmetry," *Biometrika*, vol. 58, no. 1, pp. 203–212, 1971.

[11] "Ifab rule 10 evolution." (Apr. 6, 2022), [Online]. Available: https://nachspielzeiten.de/law-10-determining-the-outcome-of-a-match-en/.

[12] "Ifab rules 2003-2004." (Apr. 6, 2022), [Online]. Available: https://downloads.theifab.com/downloads/laws-of-the-game-2003-04?l=en.

[13] M. G. Kocher, M. V. Lenz, and M. Sutter, "Psychological pressure in competitive environments: New evidence from randomized natural experiments," *Manage. Sci.*, vol. 58, no. 8, pp. 1585–1591, Aug. 2012.

[14] M. G. Kocher, M. V. Lenz, and M. Sutter, "Performance under pressure – the case of penalty shootouts in football," 2008.

[15] R. Laraki, J. Renault, and S. Sorin, "Introduction to repeated games," in *Universitext*, Cham: Springer International Publishing, 2019, pp. 151–181.

[16] I. Palacios-Huerta, *Beautiful game theory*, en. Princeton, NJ: Princeton University Press, May 2014.

[17] R. H. Randles, M. A. Fligner, G. E. Policello II, and D. A. Wolfe, "An asymptotically distribution-free test for symmetry versus asymmetry," en, *J. Am. Stat. Assoc.*, vol. 75, no. 369, pp. 168–172, Mar. 1980.

[18] N. Rudi, M. Olivares, and A. Shetty, "Ordering sequential competitions to reduce order relevance: Soccer penalty shootouts," en, *SSRN Electron. J.*, 2019.

[19] S. D. Silva, D. Mioranza, and R. Matsushita, "FIFA is right: The penalty shootout should adopt the tennis tiebreak format," *OAlib*, vol. 05, no. 03, pp. 1–23, 2018.

[20] E. L. Spitznagel Jr, "6 logistic regression," in *Handbook of Statistics*, ser. Handbook of statistics, Elsevier, 2007, pp. 187–209.

[21] "Transfermarkt." (Jan. 25, 2022), [Online]. Available: https://www.transfermarkt.com/.

[22] "Understat." (), [Online]. Available: https://understat.com/.

[23] T. P. Vandebroek, B. T. McCann, and G. Vroom, "Modeling the effects of psychological pressure on first-mover advantage in competitive interactions," en, *J. Sports Econom.*, vol. 19, no. 5, pp. 725–754, Jun. 2018.

[24] "Worldfootballr." (Apr. 6, 2022), [Online]. Available: `https://jaseziv.github.io/worldfootballR`.

Appendices

Appendix A

Law of the game evolution

Year (Source)	**new or changed laws (proposing club, if known)**
1863 (FA)	A goal is scored when the ball crosses the goal line between the goalposts from the field.
1866 (FA)	A goal is scored when the ball crosses the goal line between the goalposts and below the goal rope (as a height limit).
1867 (Sheffield FA)	A goal is scored when the ball crosses the goal line between the goalposts and below the goal band (as a height limit).
1872 (FA)	Touching the goalposts is not a goal (Wanderers FC). It is important if the ball jumps from there behind the goal line or not.
1875 (FA)	Touching the rope or the crossbar as well as the corner flags is not a goal (Queen's Park). It is important if the ball jumps from there behind the goal line or not.
1938 (IFAB)	The team scoring the greater number of goals during a game shall be the winner; if no goals, or an equal number of goals are scored the game shall be termed a 'draw'. If an outside agent wants to prevent a goal, but he failed and the ball enters the goal, the goal must be allowed.
1939 (IFAB)	Addition that the ball may not be carried into the goal (FA).
1969 (IFAB)	Intervention by an outside agent: Addition: In this case the game is restarted by a dropped ball at the place where the contact or interference occurred (SFA).
1970 (IFAB)	Introduction of the Kicks from penalty mark, but not as a part of Law 10.
1985 (IFAB)	Intervention by an outside agent: It is added that – when play was stopped and the ball was in the goal area – it is dropped on that part of the goal area line which runs parallel to the goal-line, at the point nearest to where the ball was when play (IFA).
1997 (IFAB)	Supplement that, depending on the competition rules, the extra time follows a draw after 90 minutes.
2012 (IFAB)	GLT is permitted to assist the referee in deciding whether a goal has been scored (FIFA).
2016 (IFAB)	Introduction of the Kicks from penalty mark in the Laws of the Game (previously included in the Appendix). It is added that if a referee signals a goal before the ball has passed wholly over the goal, a goal kick is awarded.
2019 (IFAB)	Kicks from penalty mark: It is added that each kick is taken by a different player, and all eligible players must take a kick before any player can take a second kick.
2020 (IFAB)	A player who has been sent off during the match is not permitted to take part; warnings and cautions issued during the match are not carried forward into kicks from the penalty 10 mark.

Table A.1: Source: [11]

Appendix B

Html files extractor - Python3

```
from bs4 import BeautifulSoup
from ftfy import fix_encoding
import json
import os.path
from datetime import datetime
import requests

#PRE:
#-There is at least one .html file manually downloaded from
                                transfermarkt.com containing a
                                 list of URL pointing to
                                football matches
#POST:
#-Created separated html files downloaded from the links
                                comprised in the manually
                                downloaded .html file(s)

def extract(html):
    #returns the list of URL pointing to football matches
                                    from a specific html code
    #PRE:
    #-the input html code comes from transfermarkt.com and
                                    contains the right divs,
                                    ids and names
    #POST:
    #-a list of valid URLs extracted from the code
```

```
    soup = BeautifulSoup(html, 'lxml')
    links=[]
    adivs = soup.find('div',{'class':'responsive-table'}).
                                    find('tbody').find_all('tr
                                    ')
    for a in adivs:
        if a.find('td',{'class':'zentriert hauptlink'}):
            links.append('https://www.transfermarkt.com'+ a.
                                            find('td',{'class'
                                            :'zentriert
                                            hauptlink'}).find(
                                            'a').get('href'))
    return links

hlinks=[]
pages=[]
for file in os.listdir():
    #gets all the .html files names and appends it to a list
    if file.endswith(".html"): pages.append(file)

for page in pages:
    #appends all the html links inside each page
    with open(page, encoding="utf8") as html:
        content = html.read()
        print(page)
        hlinks = hlinks + extract(content)

for iteration,link in enumerate(hlinks):
    #Downloads all the html code from all the links extracted
                                    from the main html codes
    if iteration < 10:
        outname = 'extract/00'+str(iteration)+'.html'
    elif iteration < 100:
        outname = 'extract/0'+str(iteration)+'.html'
    else :
        outname = 'extract/'+str(iteration)+'.html'
    with open(outname,'wb') as outfile:
        r = requests.get(link, headers={'user-agent': 'custom
                                        '})
```

```
        outfile.write(r.content)
    print(outname)
```

Appendix C

Data extractor - Python3

```
from bs4 import BeautifulSoup
from ftfy import fix_encoding
import json
from urllib.request import Request, urlopen
import os.path
from datetime import datetime

def extract(html):
    #returns parsed match data from a html code
    #PRE: The structure of the code is conform with
                                    transfermarkt.com as used
                                    in January 2022 and
                                    contains the data about a
                                    football match with
                                    penalty shootout
    #POST: A structured json object with the correct values
                                    of the extracted match
    soup = BeautifulSoup(html, 'lxml')
    data = {}
    #Scraping the html code for general match data
    date = soup.find('div',{'class':'box-content'}).find('div
                                    ',{'class':'sb-spieldaten'
                                    }).find('a').text.strip().
                                    split(" ")[1]
    print(date)
    year = date.split("/")[2]
```

```
    if len(year)==2:
        if int(year) < 30 :
            year = '20'+year
        else :
            year = '19'+year
    competition = soup.find('div',{'class':'spielername-
                                    profil'}).find('h2').find(
                                    'span').find('a').text
    competition_stage = soup.find('div',{'class':'box-content
                                    '}).find('div',{'class':'
                                    sb-spieldaten'}).find('p')
                                    .text.strip().split("\n")[
                                    0].split('|')[0].strip()
    if ("1st Leg" in competition_stage) or ("1st leg" in
                                    competition_stage):
        competition_stage = competition_stage[:-8]
        return_match = "1st leg"
    elif ("2nd Leg" in competition_stage) or ("2nd leg" in
                                    competition_stage):
        competition_stage = competition_stage[:-8]
        if soup.find('div',{'class':'box-content'}).find('div
                                        ',{'class':'sb-
                                        spieldaten'}).find('a'
                                        ,{'title':'Match
                                        report'}) :
            return_match = soup.find('div',{'class':'box-
                                            content'}).find('
                                            div',{'class':'sb-
                                            spieldaten'}).find
                                            ('a',{'title':'
                                            Match report'}).
                                            text
        else :
            return_match = soup.find('div',{'class':'box-
                                            content'}).find('
                                            div',{'class':'sb-
                                            spieldaten'}).find
                                            ('a',{'title':'
                                            Match preview'}).
```

```
                                        text
    return_match = return_match.replace(':',' - ')
else :
    return_match = 'NA'
team_left = fix_encoding(soup.find('div',{'class':'sb-
                                team sb-heim'}).text.strip
                                ()).replace('/','-')
if soup.find('div',{'class':'box-content'}).find('div',{'
                                class':'sb-spieldaten'}).
                                find('p',{'class':'sb-
                                zusatzinfos'}).find('a') :
    stadium = soup.find('div',{'class':'box-content'}).
                                    find('div',{'class':'
                                    sb-spieldaten'}).find(
                                    'p',{'class':'sb-
                                    zusatzinfos'}).find('a
                                    ').text
else :
    stadium = 'TBC'
attendees = soup.find('div',{'class':'box-content'}).find
                                ('div',{'class':'sb-
                                spieldaten'}).find('p',{'
                                class':'sb-zusatzinfos'}).
                                find('strong').text
attendees = attendees.replace('.','').replace('Attendance
                                : ','') if '.' in
                                attendees else attendees.
                                replace('Attendance: ','')
if len(soup.find('div',{'class':'box-content'}).find('div
                                ',{'class':'sb-spieldaten'
                                }).find('p',{'class':'sb-
                                zusatzinfos'}).find_all('a
                                '))>1:
    referee = soup.find('div',{'class':'box-content'}).
                                    find('div',{'class':'
                                    sb-spieldaten'}).find(
                                    'p',{'class':'sb-
                                    zusatzinfos'}).
                                    find_all('a')[1].text
```

```
    else :
        referee ='TBC'
    score_FTP_left = soup.find('div',{'class':'sb-endstand'})
                                    .text.strip().split(':')[0
                                    ]
    team_right = fix_encoding(soup.find('div',{'class':'sb-
                                    team sb-gast'}).text.strip
                                    ()).replace('/','-')
    score_FTP_right = soup.find('div',{'class':'sb-endstand'}
                                    ).text.strip().split(':')[
                                    1][:-7]
    rows = soup.find_all('div',{'class':'row'})
    first_row = 0

    #Deleting first lines
    for index, row in enumerate(rows):
        if row.find('h2') :
            first_row = index
            break
    del rows[:index]

    #fonction pour trouver le temps via la position des
                                    horloges (png)
    def get_Time(coord):
    #Returns the time of the action based on a png clock
                                    image
    #PRE: coordinates within range (0-90-120 minutes)
    #POST: a translated integer based on the displayed clock
        cx = int(coord.split(' ')[1].replace('-','').replace(
                                        'px',''))
        cy = int(coord.split(' ')[2].replace('-','').replace(
                                        'px;',''))
        cx = int(cx/36)
        cy = int(cy/36)
        if cy == 12 :
            return 'NA'
        else:
            return cy * 10 + cx + 1
```

```
#Calcul du temps de jeu
def playtime(player, subs, time):
#Returns the playtime of a player
#PRE: a player name, the list of substitution correctly
                                formatted and the duration
                                 of the match
#POST: an integer representing the playtime of a specific
                                 player or the duration of
                                 the match if the player
                                wasn't substituted during
                                the match
    if player in subs[0]:
        if subs[1][subs[0].index(player)] == "NA":
            return "NA"
        else :
            return time - int(subs[1][subs[0].index(
                                            player)])
    else :
        return time

def who_won(tl, ls, tr, rs):
#Returns the winner team based on results
#PRE: two teams name as strings and scores as integer
#POST: the name of the team with the highest score
    if ls > rs :
        return tl
    elif rs > ls :
        return tr
    else:
        return 'NA'

#Initiating the lists
shots = []
substitutions = []
yellows = []
reds = []
goals = []
missed_penalties = []
first_kicker = ''
```

```
    score_FT_left = '0'
    score_FT_right = '0'
    full_time = 120

    #Extracting the specific data
    for row in rows:

        #Not reading lines without title "h2"
        if not row.find('h2') :
            continue

        if row.find('h2').text == "Timeline":
            full_time = 120 if '120' in row.find('div',{'
                                        class':'sb-
                                        zeitleiste-
                                        ereignisse'}).get(
                                        'class')[1] else
                                        90

        #extracting goals
        if row.find('h2').text == "Goals":
            for goal in row.find_all('div',{'class':'sb-
                                        aktion'}):
                goal_player = fix_encoding(goal.find('div',{'
                                            class':'sb-
                                            aktion-aktion'
                                            }).find('a').
                                            text)
                goal_info = fix_encoding(goal.find('div',{'
                                            class':'sb-
                                            aktion-aktion'
                                            }).text.split(
                                            ',')[1])
                goal_assist = fix_encoding(goal.find('div',{'
                                            class':'sb-
                                            aktion-aktion'
                                            }).find_all('a
                                            ')[1].text) if
                                             len(goal.find
```

```
                                        ('div',{'class
                                        ':'sb-aktion-
                                        aktion'}).
                                        find_all('a'))
                                         > 1 else 'NA'
            goal_time = get_Time(goal.find('div',{'class'
                                        :'sb-aktion-
                                        uhr'}).find('
                                        span',{'class'
                                        :'sb-sprite-
                                        uhr-klein'}).
                                        get('style'))
            goals.append([goal_player, goal_info,
                                        goal_assist,
                                        goal_time])
            if goal == row.find_all('div',{'class':'sb-
                                        aktion'})[-1]:
                score_FT_left = goal.find('div',{'class':
                                            'sb-aktion
                                            -
                                            spielstand
                                            '}).text.
                                            split(':')
                                            [0]
                score_FT_right = goal.find('div',{'class'
                                            :'sb-
                                            aktion-
                                            spielstand
                                            '}).text.
                                            split(':')
                                            [1]

    #extracting shootouts
    elif row.find('h2').text == "Penalty shoot-out":
        all_shots=row.find('div',{'class':'sb-ereignisse'
                                        }).find_all('li')

        #Looking for the first kicker
```

```
            if row.find('div',{'class':'sb-ereignisse'}).
                                        find_all('li')[0].
                                        get('class')[0] ==
                                         'sb-aktion-heim':
                first_kicker = team_left
            else:
                first_kicker = team_right

            #Extracting shots and kickers
            i = 0
            left_shots=[]
            right_shots=[]
            for shot in all_shots:
                if shot.get('class')[0] == 'sb-aktion-heim':
                    shooter = fix_encoding(shot.find('div',{'
                                                class':'sb
                                                -aktion-
                                                aktion'}).
                                                text.split
                                                (',')[0])
                    goal = 1 if fix_encoding(shot.find('div',
                                                {'class':'
                                                sb-aktion-
                                                aktion'}).
                                                text.split
                                                (',')[1].
                                                strip()) =
                                                = "Scored"
                                                 else 0
                    left_shots.append([shooter,goal])
                else :
                    shooter = fix_encoding(shot.find('div',{'
                                                class':'sb
                                                -aktion-
                                                aktion'}).
                                                text.split
                                                (',')[0])
                    goal = 1 if fix_encoding(shot.find('div',
                                                {'class':'
```

```
                                                sb-aktion-
                                                aktion'}).
                                                text.split
                                                (',')[1].
                                                strip()) =
                                                = "Scored"
                                                 else 0
            right_shots.append([shooter,goal])
    if first_kicker == team_left:
        for iteration, shot in enumerate(left_shots):
            if iteration < len(right_shots) :
                shots.append([left_shots[iteration][0
                                                    ],
                                                    left_shots
                                                    [
                                                    iteration
                                                    ][1],
                                                    right_shots
                                                    [
                                                    iteration
                                                    ][0],
                                                    right_shots
                                                    [
                                                    iteration
                                                    ][1]])
            else:
                shots.append([left_shots[iteration][0
                                                    ],
                                                    left_shots
                                                    [
                                                    iteration
                                                    ][1],'
                                                    NA','
                                                    NA'])
    else:
        for iteration, shot in enumerate(right_shots)
                                                :
            if iteration < len(left_shots) :
```

```
                    shots.append([right_shots[iteration][
                                                    0],
                                                    right_shots
                                                    [
                                                    iteration
                                                    ][1],
                                                    left_shots
                                                    [
                                                    iteration
                                                    ][0],
                                                    left_shots
                                                    [
                                                    iteration
                                                    ][1]])
                else:
                    shots.append([right_shots[iteration][
                                                    0],
                                                    right_shots
                                                    [
                                                    iteration
                                                    ][1],'
                                                    NA','
                                                    NA'])
        #Extracting substitutions
        elif row.find('h2').text == "Substitutions":
            for sub in row.find_all('div',{'class':'sb-aktion
                                           '}):
                if sub.find('div',{'class':'sb-aktion-aktion'
                                                }).find('span'
                                                ,{'class':'sb-
                                                aktion-wechsel
                                                -ein'}).find('
                                                a') :
                    player_in = fix_encoding(sub.find('div',{
                                                    'class':'
                                                    sb-aktion-
                                                    aktion'}).
                                                    find('span
                                                    ',{'class'
```

```
                                        :'sb-
                                        aktion-
                                        wechsel-
                                        ein'}).
                                        find('a').
                                        text)
            else:
                player_in = 'NA'
            player_in_info = "NA"
            player_out = fix_encoding(sub.find('div',{'
                                        class':'sb-
                                        aktion-aktion'
                                        }).find('span'
                                        ,{'class':'sb-
                                        aktion-wechsel
                                        -aus'}).find('
                                        a').text)
            player_out_info = fix_encoding(sub.find('div'
                                        ,{'class':'sb-
                                        aktion-aktion'
                                        }).find('span'
                                        ,{'class':'sb-
                                        aktion-wechsel
                                        -aus'}).find('
                                        span').text.
                                        strip().
                                        replace(', ','
                                        '))
            substitution_time = get_Time(sub.find('div',{
                                        'class':'sb-
                                        aktion-uhr'}).
                                        find('span',{'
                                        class':'sb-
                                        sprite-uhr-
                                        klein'}).get('
                                        style'))
            substitutions.append([player_in,
                                        player_in_info
                                        ,player_out,
```

```
                                        player_out_info
                                        ,
                                        substitution_time
                                        ])

        #Extracting missed penalties
        elif row.find('h2').text == "missed penalties":
            missed_penalties.append(fix_encoding(row.find('
                                        span',{'class':'sb
                                        -aktion-wechsel-
                                        ein'}).find('a').
                                        text))

        #Extracting cards
        elif row.find('h2').text == "Cards":
            for sub in row.find_all('div',{'class':'sb-aktion
                                        '}):
                if "Yellow card" in sub.find('div',{'class':'
                                            sb-aktion-
                                            aktion'}).text
                                            :
                    yellow_player = fix_encoding(sub.find('
                                                div',{'
                                                class':'sb
                                                -aktion-
                                                aktion'}).
                                                find('a').
                                                text)
                    yellow_info = fix_encoding(sub.find('div'
                                                ,{'class':
                                                'sb-aktion
                                                -aktion'})
                                                .text.
                                                split(',')
                                                [1].strip
                                                ()) if ','
                                                 in sub.
                                                find('div'
                                                ,{'class':
```

```
                                        'sb-aktion
                                        -aktion'})
                                        .text else
                                         'NA'
            yellow_time = get_Time(sub.find('div',{'
                                        class':'sb
                                        -aktion-
                                        uhr'}).
                                        find('span
                                        ',{'class'
                                        :'sb-
                                        sprite-uhr
                                        -klein'}).
                                        get('style
                                        '))
            yellows.append([yellow_player,
                                        yellow_info
                                        ,
                                        yellow_time
                                        ])
        elif "Red card" in sub.find('div',{'class':'
                                    sb-aktion-
                                    aktion'}).text
                                    :
            red_player = fix_encoding(sub.find('div',
                                        {'class':'
                                        sb-aktion-
                                        aktion'}).
                                        find('a').
                                        text)
            red_info = fix_encoding(sub.find('div',{'
                                        class':'sb
                                        -aktion-
                                        aktion'}).
                                        text.split
                                        (',')[1].
                                        strip())
                                        if ',' in
                                        sub.find('
```

```
                                                    div',{'
                                                    class':'sb
                                                    -aktion-
                                                    aktion'}).
                                                    text else
                                                    'NA'
                    red_time = get_Time(sub.find('div',{'
                                                    class':'sb
                                                    -aktion-
                                                    uhr'}).
                                                    find('span
                                                    ',{'class'
                                                    :'sb-
                                                    sprite-uhr
                                                    -klein'}).
                                                    get('style
                                                    '))
                    reds.append([red_player, red_info,
                                                    red_time])
cards = []
for card in yellows:
    cards.append(card[0])

#Computing the number of goals for each player
players_goals = [[],[]]
for goal in goals:
    if goal[0] in players_goals[0]:
        players_goals[1][players_goals[0].index(goal[0])]
                                            += 1
    else:
        players_goals[0].append(goal[0])
        players_goals[1].append(1)

#Creating a subset of substitutions for playtime
                                    calculation
subsubstitutions = [[],[]]
for sub in substitutions:
    subsubstitutions[0].append(sub[0])
    subsubstitutions[1].append(sub[4])
```

```
#Loading the json object
data = ({
    'competition' : competition,
    'date' : date,
    'year' : year,
    'stage' : competition_stage,
    'winner' : who_won(team_left,score_FTP_left,
                                    team_right,
                                    score_FTP_right),
    'first_kicker' : first_kicker,
    'home_team' : team_left if (competition_stage != "
                                    Final") and (
                                    competition_stage != "
                                    final") else "NA",
    'stadium' : stadium,
    'attendees' : attendees,
    'referee' : referee
    })
if (team_left == first_kicker):
    data.update({
        'team_A': team_left,
        'team_B': team_right,
        'full_time': (score_FT_left + ' - ' +
                                        score_FT_right),
        'shoutout': (score_FTP_left + ' - ' +
                                        score_FTP_right),
        'return_match': return_match
        })
else:
    data.update({
        'team_A': team_right,
        'team_B': team_left,
        'full_time': (score_FT_right + ' - ' +
                                        score_FT_left),
        'shoutout': (score_FTP_right + ' - ' +
                                        score_FTP_left),
        'return_match': (return_match.split(' - ')[1] + '
                                         - ' +
```

```
                                        return_match.split
                                        (' - ')[0]) if ' -
                                         ' in return_match
                                         else return_match
            })
    data['shoutout_shots'] = []
    for iteration,shot in enumerate(shots):
        data['shoutout_shots'].append({
            'round': str(iteration + 1),
            'player_A': shot[0],
            'playtime_player_A': playtime(shot[0],
                                        subsubstitutions,
                                        full_time),
            'yellowcard_player_A': 1 if (shot[0] in cards)
                                        else 0,
            'goals_player_A': players_goals[1][players_goals[
                                        0].index(shot[0])]
                                         if (shot[0] in
                                        players_goals[0])
                                        else 0,
            'missed_penalties_A' : missed_penalties.count(
                                        shot[0]),
            'scored_A':shot[1],
            'shooter_B': shot[2],
            'playtime_player_B': playtime(shot[2],
                                        subsubstitutions,
                                        full_time),
            'yellowcard_player_B': 1 if (shot[2] in cards)
                                        else 0,
            'goals_player_B': players_goals[1][players_goals[
                                        0].index(shot[2])]
                                         if (shot[2] in
                                        players_goals[0])
                                        else 0,
            'missed_penalties_B' : missed_penalties.count(
                                        shot[2]),
            'scored_B': shot[3]
            })
    data['substitutions'] = []
```

```
    for sub in (substitutions):
        data['substitutions'].append({
            'time': sub[4],
            'player_in' : sub[0],
            'player_in_info' : sub[1],
            'player_out' : sub[2],
            'player_out_info' : sub[3]
            })
    data['yellow_cards'] = []
    for yel in (yellows):
        data['yellow_cards'].append({
            'time': yel[2],
            'player' : yel[0],
            'player_info' : yel[1]
            })
    data['red_cards'] = []
    for red in (reds):
        data['red_cards'].append({
            'time': red[2],
            'player' : red[0],
            'player_info' : red[1]
            })
    data['goals'] = []
    for goal in (goals):
        data['goals'].append({
            'time' : goal[3],
            'player': goal[0],
            'info': goal[1],
            'assist': goal[2]
            })
    return data


pages=[]
#Listing all matches (html pages) in the working directory
for file in os.listdir():
    if file.endswith(".html"): pages.append(file)

#Extract data from each march (html page)
```

```
for page in pages:
    print(page)
    with open(page, encoding="utf8") as html:
        content = html.read()
        data = extract(content)
        outname = data['year'] + '_' + data['competition'] +
                                       '_' + data['stage'] +
                                       '_' + data['team_A'] +
                                        '_' + data['team_B']
                                       +'.json'
        #Not creating a new file when a match has already
                                       been extracted
        if not os.path.isfile(outname):
            with open(outname,'w', encoding="utf8") as
                                           outfile:
                json.dump(data, outfile, indent=4,
                                               ensure_ascii=
                                               False)
            with open('logs.txt','a', encoding="utf8") as
                                           logs:
                logs.write(str(datetime.now())+' '+ outname+'
                                               \n')
            print(outname)
```

Appendix D

JSON files extractor - Python3

Appendix E

Data extractor, analyser and simulator - R

```
---
title: "Master thesis"
author: "Guillaume Nguyen"
date: "23/02/2022"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Libraries

```{r}
library("rjson")
library("dplyr")
library("janitor")
library("ggplot2")
library("MASS")
library("tidyverse")
library("caret")
library("aod")
```

Source path
Every file being loaded into the dataset is a specifically parsed .json
```{r}
path_to_json <- "Data/*/*.json"
paths <- Sys.glob(paths=path_to_json)
```
```

Functions
This is a list of all the function used later in the code, they are all centralized for management purposes.
1) get_competition : returns the name of the competition in a standardized way (e.g., "World Cup 1986" into "World Cup")
2) get_type : returns the type of the competition "nation" or "club"
3) get_continent : returns the continent of the competition or "World"
4) get_decisive : returns -1 if a missed shot will grant the victory to the other team, 1 if a successful shot will grant the victory to the shooting team and 0 if none of those case is true
5) get_pen : returns the number of penalty scored by a player during the game (if available)
6) normalize : returns a column with values from 0 to 1 based on the maximum and the minimum of the fed column
7) col_extract : returns a column from the match dataset based on a match ID

```{r}
get_competition <- function(full_name="NA"){
  for (comp in competition_list$name){
    if (grepl(comp,full_name,ignore.case = TRUE)){
      return(comp)
    }
  }
  return('Europa League')
}
get_type <- function(full_name="NA"){
  i <- 0
  for (comp in competition_list$name){
    i <- i+1
    if (grepl(comp,full_name,ignore.case = TRUE)){
      return(competition_list$type[i])
    }
  }
  return(full_name)
}
get_continent <- function(full_name="NA"){
  i <- 0
  for (comp in competition_list$name){
    i <- i+1
    if (grepl(comp,full_name,ignore.case = TRUE)){
      return(competition_list$continent[i])
    }
  }
  return(full_name)
}
get_decisive <- function(left,right,round,from){
  if (from == 'A') {
    if (round <= 5){
```

```
      if (left + 5 - round < right){return(-1)}
      if (left + 1 > right + 5 - round + 1){return(1)}
      return(0)
    } else {
      return(0)
    }
  } else {
    if (round <= 5){
      if (right + 5 - round < left){return(-1)}
      if (right + 1 > left + 5 - round){return(1)}
      return(0)
    } else {
      if (right + 1 > left){return(1)}
      if (right < left){return(-1)}
    }
  }
}
get_pen <- function(name){
  if (name %in% goal_list$player){
    return(goal_list[goal_list$player %in% name,]$pen)
  } else {return(0)}
}
normalize <- function(column){
  clean <- as.integer(column)
  mean_clean <- mean(clean, na.rm = TRUE)
  clean[is.na(clean)] <- mean_clean
  big <- max(clean)
  out <- c()
  for (row in clean){
    out <- c(out,row/big)
  }
  return(out)
}

col_extract <- function(in_id, out_id, var){
  out_col <- c()
  for (i in in_id){
    out_col <- rbind(out_col, var[match(i, out_id)])
  }
  return(out_col)
}
```

```
##
    ----------------------------------Extraction-------------------------------

##
    --------------------------------------------------------------------------------

## Initiating the dataframes
```

```
1) match_db : dataframe with all the data of the matches
2) shots_db : dataframe with all the shots taken during the matches
3) metadata : dataframe with the description of the dataset divided by each
    competition
4) round_pq :
5) competition_list : dataset comprising of all the tournaments, their type
    (nation/club) and their continent

```{r, echo=FALSE}
match_db <<- data.frame(id = character(),
  competition = character(),
  type = character(),
  continent = character(),
  date = character(),
  year = integer(),
  stage = character(),
  attendees = integer(),
  referee = character(),
  stadium = character(),
  team_A = character(),
  team_B = character(),
  winner = character(),
  home_team = character(),
  rounds = integer(),
  shots = integer(),
  missed = integer(),
  rm_A = character(),
  rm_B = character(),
  ft_A = integer(),
  ft_B = integer(),
  so_A = integer(),
  so_B = integer())
shots_db <<- data.frame(
  match_id = character(),
  round = integer(),
  so_A = integer(),
  so_B = integer(),
  team = character(),
  player = character(),
  scored = integer(),
  playtime = integer(),
  goals = integer(),
  yellows = integer(),
  pen = integer(),
  missed_pen = integer(),
  decisive = integer(),
  advantage = logical(),
  take = logical()
)
```
```

```
metadata <<- data.frame(
  competition = character(),
  type = character(),
  continent = character(),
  first_year = integer(),
  last_year = integer(),
  number_matches_1 = integer(),
  number_matches = integer(),
  number_shots = integer(),
  phase_1 = integer(),
  phase_2 = integer(),
  p = integer(),
  q = integer(),
  winning_A = numeric(),
  p_05 = numeric(),
  winning_A_1 = numeric(),
  p_05_1 = numeric()
)

metadata_stages <<- data.frame(
  stage = character(),
  first_year = integer(),
  last_year = integer(),
  number_matches_1 = integer(),
  number_matches = integer(),
  number_shots = integer(),
  phase_1 = integer(),
  phase_2 = integer(),
  p = integer(),
  q = integer(),
  winning_A = numeric(),
  p_05 = numeric(),
  winning_A_1 = numeric(),
  p_05_1 = numeric()
)

metadata_sub <<- data.frame(
  first_year = integer(),
  last_year = integer(),
  number_matches = integer(),
  number_shots = integer(),
  phase_1 = integer(),
  phase_2 = integer(),
  winning_A = numeric(),
  p_05 = numeric()
)



year_winning <<- data.frame(
```

```
  year = integer(),
  number = integer(),
  rate = numeric())

round_pq <<- data.frame(
  round = integer(),
  p = numeric(),
  q = numeric(),
  number = integer()
)

competition_list <<-
  data.frame(
    name = c(
      'World Cup',
      'Europa League',
      'European Champion Clubs\' Cup',
      'KNVB Beker',
      'EURO',
      'Copa America',
      'UEFA Champions League Qualifying',
      'Copa Libertadores',
      'AFC Champions League',
      'UEFA Champions League',
      'League Cup',
      'EFL Cup',
      'Copa del Rey',
      'Turkiye Kupasi',
      'Taca de Portugal Placard',
      'Italy Cup',
      'Beker van Belgie',
      'Amstel Cup',
      'CAF-Champions League',
      'Copa Sudamericana',
      'Carlsberg Cup',
      'Intercontinental Cup',
      'Allianz Cup',
      'Taca CTT',
      'UEFA Super Cup',
      'Coupe de France',
      'Coupe de la Ligue',
      'Trophee des Champions',
      'DFB-Pokal',
      'DFL-Supercup'
    ),
    type = c(
      'nation',
      'club',
      'club',
      'club',
```

```
    'nation',
    'nation',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club',
    'club'
  ),
  continent = c(
    'WORLD',
    'EU',
    'EU',
    'EU',
    'EU',
    'SAM',
    'EU',
    'SAM',
    'ASIA',
    'EU',
    'EU',
    'EU',
    'EU',
    'EU',
    'EU',
    'EU',
    'EU',
    'EU',
    'EU',
    'AFR',
    'SAM',
    'EU',
    'WORLD',
```

```
      'EU',
      'EU',
      'EU',
      'EU',
      'EU',
      'EU',
      'EU',
      'EU'
    )
  )

```

```

## Populating the dataframes

```{r, echo=FALSE}
id = 0
status = 0
prev_status = 0
palen = length(paths)
for (path in paths) {
  status <- status + 1
  if (round(status/palen * 100, digits = 0) != prev_status) {
    print(round(status/palen * 100, digits = 0))
    prev_status <- round(status/palen * 100, digits = 0)
    }
  #Loading temporary variables
  #1)for each match
  temp <- fromJSON(file = path)
  #get all goals to see if a player scored a penalty during the game
  goal_list <- data.frame(player = character(),pen = integer())
  for (goal in temp$goals) {
    if (goal$player %in% goal_list$player) {
      index <- match(goal$player,goal_list$player)
      goal_list[index,] <- data.frame(player = goal$player,pen = ifelse(
          grepl('penalty',goal$info, ignore.case = TRUE),goal_list[index,2]
          + 1,goal_list[index,2]))
    } else {
      goal_list <- rbind(goal_list,data.frame(player=goal$player, pen=ifelse
          (grepl('penalty',goal$info, ignore.case = TRUE),1,0)))
      }
  }
  competition <- get_competition(temp$competition)
  type <- get_type(competition)
  continent <- get_continent(competition)
  date <- temp$date
  year <- as.integer(temp$year)
  stage <- temp$stage
  attendees <- as.integer(temp$attendees)
```
```

```
referee <- temp$referee
stadium <- temp$stadium
team_A <- temp$team_A
team_B <- temp$team_B
home_team <- ifelse(temp$home_team==temp$team_A,'A','B')
rm_A <- ifelse(temp$return_match=="NA","NA",strsplit(temp$return_match,' -
    ')[[1]][1])
rm_B <- ifelse(temp$return_match=="NA","NA",strsplit(temp$return_match,' -
    ')[[1]][2])
ft_A <- as.integer(strsplit(temp$full_time,' - ')[[1]][1])
ft_B <- as.integer(strsplit(temp$full_time,' - ')[[1]][2])
so_A <- 0
so_B <- 0
id <- id + 1
#2) for each rounds
round <- 0
shots <- 0
missed <- 0
for (r in temp$shoutout_shots){
  round <- round + 1
  shots <- shots + 1
  #player starting the round
  #update the shootout score
  shots_db <- rbind(shots_db,data.frame(
    match_id = id,
    round = round,
    so_A = so_A,
    so_B = so_B,
    team = 'A',
    player = r$player_A,
    scored = r$scored_A,
    playtime = r$playtime_player_A,
    goals = r$goals_player_A,
    yellows = r$yellowcard_player_A,
    pen = get_pen(r$player_A),
    missed_pen = r$missed_penalties_A,
    decisive = get_decisive(so_A,so_B,round,'A'),
    advantage = ifelse(so_A > so_B,TRUE,FALSE),
    take = ifelse(so_A == so_B,TRUE,FALSE)))
  so_A <- so_A + as.integer(r$scored_A)
  missed <- missed - (as.integer(r$scored_A) - 1)
  #second player
  #update the shootout score and the dataframe

  if (r$shooter_B != 'NA'){
    shots <- shots + 1
    shots_db <- rbind(shots_db,data.frame(
      match_id = id,
      round = round,
      so_A = so_A,
```

```
        so_B = so_B,
        team = 'B',
        player = r$shooter_B,
        scored = r$scored_B,
        playtime = r$playtime_player_B,
        goals = r$goals_player_B,
        yellows = r$yellowcard_player_B,
        pen = get_pen(r$shooter_B),
        missed_pen = r$missed_penalties_B,
        decisive = get_decisive(so_A,so_B,round,'B'),
        advantage = ifelse(so_A < so_B,TRUE,FALSE),
        take = ifelse(so_A == so_B,TRUE,FALSE)))
      so_B <- so_B + as.integer(r$scored_B)
      missed <- missed - (as.integer(r$scored_B) - 1)
    }
  }

  #update the match dataframe
  #being sure that winner is the shootout winner (using actual shots)
  winner <- ifelse(temp$winner == temp$team_A,'A','B')
  match_db <- rbind(match_db,data.frame(
    id = id,
    competition = competition,
    type = type,
    continent = continent,
    date = date,
    year = year,
    stage = stage,
    attendees = attendees,
    referee = referee,
    stadium = stadium,
    team_A = team_A,
    team_B = team_B,
    winner = winner,
    home_team = home_team,
    rounds = round,
    shots = shots,
    missed = missed,
    rm_A = rm_A,
    rm_B = rm_B,
    ft_A = ft_A,
    ft_B = ft_B,
    so_A = so_A,
    so_B = so_B)
  )
}
```

```
## Cleaning the dataframes
```

```{r}
clean_match_db <- match_db
clean_shots_db <- shots_db
status = 0
prev_status = 0
malen = nrow(match_db)
#Cleaning rules :
#1) no shootouts in table "shots_db" or less than 3 rounds of data
no_shootouts <- function(row){
  if(match_db$rounds[row]<3){return(TRUE)}
  return(FALSE)
}
#2) unbalanced number of shootouts (so_B>so_A or so_A > so_B + 1)
no_balance <- function(row){
  if (no_shootouts(row)) {return(TRUE)}
  count <- subset(shots_db, match_id == match_db[row, "id"]) %>% count(team,
      sort = TRUE)
  if (nrow(count) < 2) {return(TRUE)}
  countA <- subset(count, team == "A")$n
  countB <- subset(count, team == "B")$n
  if (countB > countA | countA > countB + 1){return(TRUE)}
  else {return(FALSE)}
}

#Cleaning
#utilisation de match_db pour le tri afin de ne pas boucler sur un set qui
    change de taille
for (row in 1:nrow(match_db)){
  status <- status + 1
  if (round(status/malen * 100, digits = 0) != prev_status) {
    print(round(status/malen * 100, digits = 0))
    prev_status <- round(status/malen * 100, digits = 0)
  }
  if (no_shootouts(row) | no_balance(row)){
    clean_match_db <- subset(clean_match_db, id != match_db[row, "id"])
    clean_shots_db <- subset(clean_shots_db, match_id != match_db[row, "id
        "])
  }
}
```

```
##
    ------------------------------------Analysis---------------------------------

##
    ------------------------------------------------------------------------------

## ==================================General
    ==================================
```

```
## tab:4.1 - Proportions per competition
Proportion of team_A (shooting first) winning for each competition and the
    total
```{r}
for (comp in unique(clean_match_db$competition)){
  old_temp_df <- subset(match_db, competition == comp)
  temp_df <- subset(clean_match_db, competition == comp)
  temp_shots <- clean_shots_db[clean_shots_db$match_id %in% temp_df$id,]
  metadata <- rbind(metadata, data.frame(
    type = subset(competition_list, name == comp)$type[1] ,
    continent = subset(competition_list, name == comp)$continent[1] ,
    competition = comp,
    first_year = min(temp_df$year),
    last_year = max(temp_df$year),
    number_matches_1 = nrow(old_temp_df),
    number_matches = nrow(temp_df),
    number_shots = sum(temp_df$shots),
    phase_1 = nrow(subset(temp_df, rounds <= 5)),
    phase_2 = nrow(subset(temp_df, rounds > 5)),
    p = round(proportions(table(subset(temp_shots, team == 'A')$scored))
        ['1'],3),
    q = round(proportions(table(subset(temp_shots, team == 'B')$scored))
        ['1'],3),
    winning_A = round(proportions(table(temp_df$winner))['A'], 2),
    p_05 = round(prop.test(x = table(temp_df$winner)['A'], n = length(
        temp_df$winner), p=0.5, alternative = "greater", conf.level = 0.95)
        $p.value, 3),
    winning_A_1 = round(proportions(table(old_temp_df$winner))['A'], 2),
    p_05_1 = round(prop.test(x = table(old_temp_df$winner)['A'], n = length(
        old_temp_df$winner), p=0.5, alternative = "greater", conf.level =
        0.95)$p.value, 3)
  ))
}
```


tab:4.1 - Total of proportions
```{r}
metadata <- rbind(metadata, data.frame(
    type = "TOTAL",
    continent = "TOTAL",
    competition = "TOTAL",
    first_year = min(clean_match_db$year),
    last_year = max(clean_match_db$year),
    number_matches_1 = nrow(match_db),
    number_matches = nrow(clean_match_db),
    number_shots = sum(clean_match_db$shots),
    phase_1 = nrow(subset(clean_match_db, rounds <= 5)),
    phase_2 = nrow(subset(clean_match_db, rounds > 5)),
    p = round(proportions(table(subset(clean_shots_db, team == 'A')$scored))
```
```

```
        [['1']],3),
    q = round(proportions(table(subset(clean_shots_db, team == 'B')$scored))
        [['1']],3),
    winning_A = round(proportions(table(clean_match_db$winner))[['A']], 2),
    p_05 = round(prop.test(x = table(clean_match_db$winner)['A'], n = length
        (clean_match_db$winner), p=0.5, alternative = "greater", conf.level
        = 0.95)$p.value, 3),
    winning_A_1 = round(proportions(table(match_db$winner))[['A']], 2),
    p_05_1 = round(prop.test(x = table(match_db$winner)['A'], n = length(
        match_db$winner), p=0.5, alternative = "greater", conf.level = 0.95)
        $p.value, 3)
  ))
```
```

## tab:4.2 - Proportions per type
```{r}
for (ty in unique(clean_match_db$type)){
  old_temp_df <- subset(match_db, type == ty)
  temp_df <- subset(clean_match_db, type == ty)
  temp_shots <- clean_shots_db[clean_shots_db$match_id %in% temp_df$id,]
  metadata <- rbind(metadata, data.frame(
    type = ty,
    continent = "ALL",
    competition = "ALL",
    first_year = min(temp_df$year),
    last_year = max(temp_df$year),
    number_matches_1 = nrow(old_temp_df),
    number_matches = nrow(temp_df),
    number_shots = sum(temp_df$shots),
    phase_1 = nrow(subset(temp_df, rounds <= 5)),
    phase_2 = nrow(subset(temp_df, rounds > 5)),
    p = round(proportions(table(subset(temp_shots, team == 'A')$scored))
        [['1']],3),
    q = round(proportions(table(subset(temp_shots, team == 'B')$scored))
        [['1']],3),
    winning_A = round(proportions(table(temp_df$winner))[['A']], 2),
    p_05 = round(prop.test(x = table(temp_df$winner)['A'], n = length(
        temp_df$winner), p=0.5, alternative = "greater", conf.level = 0.95)
        $p.value, 3),
    winning_A_1 = round(proportions(table(old_temp_df$winner))[['A']], 2),
    p_05_1 = round(prop.test(x = table(old_temp_df$winner)['A'], n = length(
        old_temp_df$winner), p=0.5, alternative = "greater", conf.level =
        0.95)$p.value, 3)
  ))
}
```

tab:4.2 - Proportions per continent
```{r}
```
```

```
for (cont in unique(clean_match_db$continent)){
  old_temp_df <- subset(match_db, continent == cont)
  temp_df <- subset(clean_match_db, continent == cont)
  temp_shots <- clean_shots_db[clean_shots_db$match_id %in% temp_df$id,]
  metadata <- rbind(metadata, data.frame(
    type = "ALL",
    continent = cont,
    competition = "ALL",
    first_year = min(temp_df$year),
    last_year = max(temp_df$year),
    number_matches_1 = nrow(old_temp_df),
    number_matches = nrow(temp_df),
    number_shots = sum(temp_df$shots),
    phase_1 = nrow(subset(temp_df, rounds <= 5)),
    phase_2 = nrow(subset(temp_df, rounds > 5)),
    p = round(proportions(table(subset(temp_shots, team == 'A')$scored))
        [['1']],3),
    q = round(proportions(table(subset(temp_shots, team == 'B')$scored))
        [['1']],3),
    winning_A = round(proportions(table(temp_df$winner))[['A']], 2),
    p_05 = round(prop.test(x = table(temp_df$winner)['A'], n = length(
        temp_df$winner), p=0.5, alternative = "greater", conf.level = 0.95)
        $p.value, 3),
    winning_A_1 = round(proportions(table(old_temp_df$winner))[['A']], 2),
    p_05_1 = round(prop.test(x = table(old_temp_df$winner)['A'], n = length(
        old_temp_df$winner), p=0.5, alternative = "greater", conf.level =
        0.95)$p.value, 3)
  ))
}
```

```
## tab:4.2 - Proportions per continent and per type
```{r}
for (cont in unique(clean_match_db$continent)){
  for (ty in unique(clean_match_db$type)){
    old_temp_df <- subset(subset(match_db, continent == cont), type == ty)
    temp_df <- subset(subset(clean_match_db, continent == cont), type == ty)
    temp_shots <- clean_shots_db[clean_shots_db$match_id %in% temp_df$id,]
    if (nrow(temp_df)!=0){
      metadata <- rbind(metadata, data.frame(
    type = ty,
    continent = cont,
    competition = "ALL",
    first_year = min(temp_df$year),
    last_year = max(temp_df$year),
    number_matches_1 = nrow(old_temp_df),
    number_matches = nrow(temp_df),
    number_shots = sum(temp_df$shots),
    phase_1 = nrow(subset(temp_df, rounds <= 5)),
```
```

```
    phase_2 = nrow(subset(temp_df, rounds > 5)),
    p = round(proportions(table(subset(temp_shots, team == 'A')$scored))
        [['1']],3),
    q = round(proportions(table(subset(temp_shots, team == 'B')$scored))
        [['1']],3),
    winning_A = round(proportions(table(temp_df$winner))[['A']], 2),
    p_05 = round(prop.test(x = table(temp_df$winner)['A'], n = length(
        temp_df$winner), p=0.5, alternative = "greater", conf.level = 0.95)
        $p.value, 3),
    winning_A_1 = round(proportions(table(old_temp_df$winner))[['A']], 2),
    p_05_1 = round(prop.test(x = table(old_temp_df$winner)['A'], n = length(
        old_temp_df$winner), p=0.5, alternative = "greater", conf.level =
        0.95)$p.value, 3)
  ))
    }
}}
```

```
## tab:4.3 - Proportions per competition stage
Proportion of team_A (shooting first) winning for each competition and the
    total
```{r}
for (sta in unique(clean_match_db$stage)){
  old_temp_df <- subset(match_db, stage == sta)
  if(nrow(old_temp_df)>=30){
      temp_df <- subset(clean_match_db, stage == sta)
      temp_shots <- clean_shots_db[clean_shots_db$match_id %in% temp_df$id,]
      try_one <- try(round(proportions(table(temp_df$winner))[['A']], 2))
      if (!inherits(try_one, "try-error")) {
        rate_escape <- try_one
      } else {
        ifelse(temp_df[1]$winner =='A', rate_escape <- 1, rate_escape <- 0)
      }
      try_two <- try(round(proportions(table(old_temp_df$winner))[['A']], 2)
          )
      if (!inherits(try_two, "try-error")) {
        rate_escape_old <- try_two
      } else {
        ifelse(temp_df[1]$winner =='A', rate_escape_old <- 1,
            rate_escape_old <- 0)
      }
      metadata_stages <- rbind(metadata_stages, data.frame(
        stage = sta,
        first_year = min(temp_df$year),
        last_year = max(temp_df$year),
        number_matches_1 = nrow(old_temp_df),
        number_matches = nrow(temp_df),
        number_shots = sum(temp_df$shots),
        phase_1 = nrow(subset(temp_df, rounds <= 5)),
```
```

```
        phase_2 = nrow(subset(temp_df, rounds > 5)),
        p = round(proportions(table(subset(temp_shots, team == 'A')$scored))
            [['1']],3),
        q = round(proportions(table(subset(temp_shots, team == 'B')$scored))
            [['1']],3),
        winning_A = rate_escape,
        p_05 = ifelse(nrow(temp_df)>=30,round(prop.test(x = table(
            temp_df$winner)['A'], n = length(temp_df$winner), p=0.5,
            alternative = "greater", conf.level = 0.95)$p.value, 3),1),
        winning_A_1 = rate_escape_old,
        p_05_1 = ifelse(nrow(old_temp_df)>=30,round(prop.test(x = table(
            old_temp_df$winner)['A'], n = length(old_temp_df$winner), p=0.5,
             alternative = "greater", conf.level = 0.95)$p.value, 3),1)
      ))
  }
}
```

```
## ========================General plot analysis
    ========================
## Setting the colors for the plots
```{r}
fifa_lightblue <- "#9BD4FF"
fifa_red <- "#E8271E"
fifa_blue <- "#035AAA"
fifa_yellow <- "#FEC310"
fifa_green <- "#3CAC3B"
```
Fig:4.1 - A winning through years
graph on % A winning through the years, estimate professionalization of
    player, potential decrease in "pressure"
```{r}
for (ye in sort(unique(clean_match_db$year), decreasing=FALSE)){
  temp_df <- subset(clean_match_db, year == ye)
  year_winning <- rbind(year_winning, data.frame(
    year = ye,
    number = nrow(temp_df),
    rate = proportions(table(temp_df$winner))[['A']]
    ))
}
ggplot(data=year_winning)+
  geom_col(aes(x=year, y=number/max(year_winning$number)),size = 1, fill =
      fifa_lightblue)+
  geom_line(aes(x=year, y=rate), color = fifa_yellow, size = 1) +
  geom_point(aes(x=year, y=rate), color = "lightgrey") +
  geom_line(aes(x=year, y=0.5), linetype="dashed", color=fifa_red)+
  scale_x_continuous(name = "Year")+
  scale_y_continuous(name = "Rates", limits = c(0,1),sec.axis = sec_axis(~.*
      max(year_winning$number), name = "Number of matches")) +
```
```

```
  theme_minimal()
```


```
## Fig:4.2 - Success rate (p,q) of shots per round
```{r}
for (r in sort(unique(clean_shots_db$round), decreasing=FALSE)){
  temp_df <- subset(clean_shots_db, round == r)
  round_pq <- rbind(round_pq, data.frame(
    round = r,
    p = proportions(table(subset(temp_df, team == 'A')$scored))['1'],
    q = proportions(table(subset(temp_df, team == 'B')$scored))['1'],
    number = nrow(temp_df)
  ))
}

ggplot(data=round_pq)+
  geom_col(aes(x=round, y=number/max(round_pq$number)),size = 1, fill =
      fifa_lightblue)+
  geom_line(aes(x=round, y=p, colour = "p"),size = 1)+
  geom_line(aes(x=round, y=q, colour = "q"),size = 1)+
  geom_line(aes(x=round, y=3/4, linetype="75%"), color=fifa_red)+
  geom_line(aes(x=round, y=2/3, linetype="66%"), color=fifa_red)+
  scale_colour_manual("",
                      breaks = c("p", "q"),
                      values = c(fifa_green, fifa_blue))+
  scale_linetype_manual("",
                        breaks = c("75%","66%"),
                        values = c("dashed","dotted"))+
  scale_x_continuous(name = "Rounds")+
  scale_y_continuous(name = "Rates", limits = c(0,1),sec.axis = sec_axis(~.*
      max(round_pq$number), name = "Number of shots")) +
  theme_minimal()
```


```
## Fig:4.3 - Boxplot de shots par competition
```{r}
ggplot(clean_match_db, aes(x=competition))+
  stat_count(aes(y=after_stat(count*max(clean_match_db$shots)/max(table(
      clean_match_db$competition)))),  fill = fifa_lightblue)+
  scale_y_continuous(name = "Shots",limits = c(0,max(clean_match_db$shots)),
      sec.axis = sec_axis(~.*max(table(clean_match_db$competition))/max(
      clean_match_db$shots), name = "Number of matches"))+
  geom_boxplot(aes(y = shots),outlier.color = fifa_red, color=fifa_red, fill
      =fifa_yellow)+
  xlab("Competitions")+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90))
```

```
## ==============================Other general analysis
    ==========================
## Init the new shot db for analysis
```{r}
csdb2 <- clean_shots_db
csdb2$decisiveWIN <- ifelse(csdb2$decisive==1,1,0)
csdb2$decisiveLOSE <- ifelse(csdb2$decisive==-1,1,0)
csdb2$playtime <- normalize(csdb2$playtime)
csdb2$year <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$year)
csdb2$attendees <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$attendees)[,1]

csdb2$stage <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$stage)
csdb2$diff <- ifelse(csdb2$team == 'A', csdb2$so_A - csdb2$so_B, csdb2$so_B
    - csdb2$so_A)
csdb2$roundsep <- ifelse(csdb2$round <= 5, csdb2$round, 6)
csdb2$factdiff <- factor(csdb2$diff)
csdb2$type <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$type)
csdb2$home_team <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$home_team)
csdb2$ground <- ifelse(csdb2$type == "club", ifelse(csdb2$team ==
    csdb2$home_team, "home", "away"), "neutral")
csdb2$continent <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$continent)
csdb2$competition <- col_extract(csdb2$match_id, clean_match_db$id,
    clean_match_db$competition)

```

Tab:4.5-6 - Median and mean of shots/missed shots
```{r}
med_mea_shots <<- data.frame(row = c('Median','Mean'), shots = c(median(
    clean_match_db$shots),mean(clean_match_db$shots)), scored = c(median(
    clean_match_db$shots-clean_match_db$missed),mean(clean_match_db$shots-
    clean_match_db$missed)), missed = c(median(clean_match_db$missed),mean(
    clean_match_db$missed)))

clean_match_db_p1 <<- subset(clean_match_db, rounds <= 5)
clean_match_db_p2 <<- subset(clean_match_db, rounds > 5)

med_mea_shots_p1 <<- data.frame(row = c('Median','Mean'), shots = c(median(
    clean_match_db_p1$shots),mean(clean_match_db_p1$shots)), scored = c(
    median(clean_match_db_p1$shots-clean_match_db_p1$missed),mean(
    clean_match_db_p1$shots-clean_match_db_p1$missed)), missed = c(median(
```
```

```
    clean_match_db_p1$missed),mean(clean_match_db_p1$missed)))

med_mea_shots_p2 <<- data.frame(row = c('Median','Mean'), shots = c(median(
    clean_match_db_p2$shots),mean(clean_match_db_p2$shots)), scored = c(
    median(clean_match_db_p2$shots-clean_match_db_p2$missed),mean(
    clean_match_db_p2$shots-clean_match_db_p2$missed)), missed = c(median(
    clean_match_db_p2$missed),mean(clean_match_db_p2$missed)))

t.test((clean_match_db_p1$shots-clean_match_db_p1$missed)/
    clean_match_db_p1$shots,(clean_match_db_p2$shots-
    clean_match_db_p2$missed)/clean_match_db_p2$shots, alternative = "less")
```
```
## Tab:4.7 - Proportions pre and post 2003
```{r}
cmdb_pre <- subset(clean_match_db, year<2003)
cmdb_post <- subset(clean_match_db, year>=2003)

temp_df <- cmdb_pre
for (i in range(1:2)){
  metadata_sub <- rbind(metadata_sub, data.frame(
    first_year = min(temp_df$year),
    last_year = max(temp_df$year),
    number_matches = nrow(temp_df),
    number_shots = sum(temp_df$shots),
    phase_1 = nrow(subset(temp_df, rounds <= 5)),
    phase_2 = nrow(subset(temp_df, rounds > 5)),
    winning_A = round(proportions(table(temp_df$winner))['A'], 2),
    p_05 = round(prop.test(x = table(temp_df$winner)['A'], n = length(
        temp_df$winner), p=0.5, alternative = "greater", conf.level = 0.95)
        $p.value, 2)))


  temp_df <- cmdb_post
  }


```


```
## Tab:4.8-10 - Probabilities for the difference in successful shots

```{r}
diff_frame <<- data.frame(team = character(), s = integer(), prob = numeric
    ())
for(s in unique(csdb2$diff)){
  for(t in unique(csdb2$team)){
    temp_df <<- subset(subset(subset(csdb2, diff == s), round <=5), team ==
        t)
    diff_frame <- rbind(diff_frame, data.frame(team = t, s = s, prob = round
        (mean(temp_df$scored),3)))
```

```
  }
}
for(s in unique(csdb2$diff)){
  temp_df <<- subset(subset(csdb2, diff == s), round <=5)
  diff_frame <- rbind(diff_frame, data.frame(team = "total", s = s, prob =
      round(mean(temp_df$scored),3)))
}
```

Proportion for the difference in successful shots

```{r}
diffprop_frame <<- data.frame(team = character(), s = integer(), prop =
    numeric(), p.value = numeric())
for(s in unique(csdb2$diff)){
  for(t in unique(csdb2$team)){
    temp_df <<- subset(subset(subset(csdb2, diff == s), round <=5), team ==
        t)
    temp_df2 <<- subset(subset(csdb2, diff == s), round <=5)
    proptest <- prop.test(x=nrow(temp_df), n=nrow(temp_df2), p=0.5, correct
        = FALSE)
    diffprop_frame <- rbind(diffprop_frame, data.frame(team = t, s = s, prop
         = round(proptest$estimate,3), p.value = round(proptest$p.value,5)))
  }
}
```

probabilities for each round and team

```{r}
round_t_frame <<- data.frame(teamA = numeric(), teamB = numeric(), round =
    integer(), p.value = numeric())
for(r in unique(csdb2$roundsep)){
  temp_df <<- subset(csdb2, roundsep == r)
  round_t_frame <- rbind(round_t_frame, data.frame(teamA = round(mean(subset
      (temp_df, team == 'A')$scored),3), teamB = round(mean(subset(temp_df,
      team == 'B')$scored),3), round = r, p.value = round(t.test(subset(
      temp_df, team == 'A')$scored,subset(temp_df, team == 'B')$scored,
      alternative = 'greater')$p.value,4)))
}
nrow(subset(csdb2, team == 'A'))
nrow(subset(csdb2, team == 'B'))
```

4.1 - Shot prediction model

Tab:4.11 - Logistic Regression

```{r}
logit0 <- glm( scored ~ round + team + so_A + so_B + playtime + goals +
    yellows + pen + missed_pen + decisive + advantage + take + attendees +
    diff + ground + type + competition + stage + continent , data = csdb2,
    family = binomial)
```

```
summary(logit0)

logit1 <- glm( scored ~ round + team + decisiveWIN + decisiveLOSE +
    advantage + attendees, data = csdb2, family = binomial)
summary(logit1)

logit2 <- glm( scored ~ round + team + decisiveLOSE + advantage , data =
    csdb2, family = binomial)
summary(logit2)
```

Tab:4.11 - Probit Regression

```{r}
probit0 <- glm(scored ~ round + team + so_A + so_B + playtime + goals +
    yellows + pen + missed_pen + decisive + advantage + take + attendees +
    diff + ground + type + competition + stage + continent , data = csdb2,
    family = binomial(link = "probit"))
summary(probit0)

probit1 <- glm( scored ~ round + team + decisiveWIN + decisiveLOSE +
    advantage + attendees, data = csdb2, family = binomial(link = "probit"))
summary(probit1)

probit2 <- glm(scored ~ round + team + decisiveLOSE + advantage, data =
    csdb2, family = binomial(link = "probit"))
summary(probit2)
```

Tab:4.12 - Proportions

```{r}

pred_props <<- data.frame(factor = character(), A = numeric(), B = numeric()
    )
pred_props <- rbind(pred_props, data.frame(factor = "shots", A = proportions
    (table(csdb2$team))['A'], B= proportions(table(csdb2$team))['B'] ))
pred_props <- rbind(pred_props, data.frame(factor = "decisivewin",
                                           A = nrow(subset(subset(csdb2,
                                               decisiveWIN == 1), team == 'A
                                               '))/nrow(subset(csdb2,
                                               decisiveWIN == 1)),
                                           B = nrow(subset(subset(csdb2,
                                               decisiveWIN == 1), team == 'B
                                               '))/nrow(subset(csdb2,
                                               decisiveWIN == 1))
                                             ))
pred_props <- rbind(pred_props, data.frame(factor = "decisivelose",
                                           A = nrow(subset(subset(csdb2,
                                               decisiveLOSE == 1), team == '
                                               A'))/nrow(subset(csdb2,
                                               decisiveLOSE == 1)),
```

```
                                          B = nrow(subset(subset(csdb2,
                                              decisiveLOSE == 1), team == '
                                              B'))/nrow(subset(csdb2,
                                              decisiveLOSE == 1))
                                            ))
pred_props <- rbind(pred_props, data.frame(factor = "advantage",
                                          A = nrow(subset(subset(csdb2,
                                              advantage == TRUE), team == '
                                              A'))/nrow(subset(csdb2,
                                              advantage == TRUE)),
                                          B = nrow(subset(subset(csdb2,
                                              advantage == TRUE), team == '
                                              B'))/nrow(subset(csdb2,
                                              advantage == TRUE))
                                            ))
```

```

##
    ----------------------------------Simulation----------------------------------

##
    ------------------------------------------------------------------------------

##simulation
Functions:
get_advantage: Returns TRUE or FALSE if a team is leading the score before
    taking a shot
get_decisive_bool: Returns a data frame with 1 or 0 if a team is taking a
    decisive shot. Whether it is a decisive LOSE or WIN.
get_shot_pos: returns a shot and its information after aligning if it is
    taken at the beginning or at the end of the round
generate_match: generates a penalty shoot-out
generate_round: generates a round
generate_shot: generates a single shot

```{r}
get_advantage <- function(score_A, score_B, team){
  if(team == 'A'){
    if(score_A>score_B){
      return(TRUE)
    }else{return(FALSE)}
  }else{
    if(score_B>score_A){
      return(TRUE)
    }else{return(FALSE)}
  }
}

get_decisive_bool <- function(score_A, score_B, round, team, pos){
  if(pos == 1){
```
```

```
    if(team == 'A'){
      win = ifelse(get_decisive(score_A,score_B,round,'A') == 1, 1, 0)
      lose = ifelse(get_decisive(score_A,score_B,round,'A') == -1, 1, 0)
    }else{
      win = ifelse(get_decisive(score_B,score_A,round,'A') == 1, 1, 0)
      lose = ifelse(get_decisive(score_B,score_A,round,'A') == -1, 1, 0)
    }
  } else {
    if(team == 'A'){
      win = ifelse(get_decisive(score_B,score_A,round,'B') == 1, 1, 0)
      lose = ifelse(get_decisive(score_B,score_A,round,'B') == -1, 1, 0)
    }else{
      win = ifelse(get_decisive(score_A,score_B,round,'B') == 1, 1, 0)
      lose = ifelse(get_decisive(score_A,score_B,round,'B') == -1, 1, 0)
    }
  }
  return(data.frame(WIN = win, LOSE = lose))
}

get_shot_pos <- function(lastrow, row){
  if(nrow(lastrow)==0){return(row)
    }else{
    ret_row <- row
    if(row$round == lastrow$round){
      ret_row$team <- 'B'
    } else {
      ret_row$team <- 'A'
    }
    return(ret_row)
  }
}

generate_match <- function(rule = 'ABAB', model, id, attendees) {
  won <- FALSE
  round <- 0
  round_result <<- data.frame(won = logical(), winner = character() )
  while(!won){
    round <- round + 1
    round_results <- generate_round(round, rule, model, id, attendees)
    won <- round_results$won
  }
  temp_df <- subset(sim_shots, match_ID == id)
  lastrow <- temp_df[nrow(temp_df),]
  sim_match <<- rbind(sim_match, data.frame(match_ID = id,
                                            winner = round_results$winner,
                                            score_A = ifelse(lastrow$team ==
                                                'A', lastrow$score_A +
                                                lastrow$scored ,
                                                lastrow$score_A),
                                            score_B = ifelse(lastrow$team ==
```

```
                                                    'B', lastrow$score_B +
                                                    lastrow$scored ,
                                                    lastrow$score_B),
                                                rounds = temp_df[nrow(temp_df),]
                                                    $round,
                                                attendees = attendees
                                                ))
}

generate_round <- function(round, rule, model, id, attendees){
  shot_results <- generate_shot(round, rule, model, id, attendees)
  if (!shot_results$won){
    return(generate_shot(round, rule, model, id, attendees))
  } else {
    return(shot_results)
  }
}

generate_shot <- function(round, rule, model, id, attendees){
  temp_df <- subset(sim_shots, match_ID == id)
  won <- FALSE
  winner <- "NA"
  if(nrow(temp_df) == 0){
    team <- 'A'
    score_A <- 0
    score_B <- 0
    decisiveWIN <- 0
    decisiveLOSE <- 0
    advantage <- FALSE
  }else{
    lastrow <- temp_df[nrow(temp_df),]
    if (rule == 'ACU'){
      acu <- TRUE
      if(round <= 5){
        rule <- 'CU'
      }else{
        rule <- 'ABBA'
      }
    } else{acu <- FALSE}
    if(rule == 'ABAB'){
       if (lastrow$team == 'A'){
         team <- 'B'
         score_A <- lastrow$score_A + lastrow$scored
         score_B <- lastrow$score_B
         decisiveWIN <- get_decisive_bool(score_A, score_B, round, team, 2)
             $WIN
         decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team, 2)
             $LOSE
         advantage <- get_advantage(score_A,score_B,team)
       } else {
```

```
      team <- 'A'
      score_B <- lastrow$score_B + lastrow$scored
      score_A <- lastrow$score_A
      decisiveWIN <- get_decisive_bool(score_A, score_B, round, team, 1)
          $WIN
      decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team, 1)
          $LOSE
      advantage <- get_advantage(score_A,score_B,team)
}}
if(rule == 'ABBA'){
  if(nrow(temp_df) == 1){
    team <- 'B'
    score_A <- lastrow$scored
    score_B <- 0
    decisiveWIN <- 0
    decisiveLOSE <- 0
    advantage <- FALSE
  }else{
    if(round == 6 && acu){
      if(lastrow$round < 6){
        team <- 'B'
        score_A <- lastrow$score_A + ifelse(lastrow$team == 'A' &&
            lastrow$scored,1,0)
        score_B <- lastrow$score_B + ifelse(lastrow$team == 'B' &&
            lastrow$scored,1,0)
        decisiveWIN <- 0
        decisiveLOSE <- 0
        advantage <- FALSE
      }else{
        team <- 'A'
        score_A <- lastrow$score_A
        score_B <- lastrow$score_B + lastrow$scored
        decisiveWIN <- ifelse(lastrow$scored,0,1)
        decisiveLOSE <- ifelse(lastrow$scored,1,0)
        advantage <- FALSE
      }
    }else{
      prelastrow <- temp_df[nrow(temp_df)-1,]
      if(lastrow$team == prelastrow$team){
        if(lastrow$team == 'A'){
        team <- 'B'
        score_A <- lastrow$score_A + lastrow$scored
        score_B <- lastrow$score_B
        decisiveWIN <- get_decisive_bool(score_A, score_B, round, team,
            2)$WIN
        decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team,
             2)$LOSE
        advantage <- get_advantage(score_A,score_B,team)
        }else{
        team <- 'A'
```

```
            score_A <- lastrow$score_A
            score_B <- lastrow$score_B + lastrow$scored
            decisiveWIN <- get_decisive_bool(score_A, score_B, round, team,
                2)$WIN
           decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team,
               2)$LOSE
            advantage <- get_advantage(score_A,score_B,team)
            }
          }else{
            if(lastrow$team == 'A'){
            team <- 'A'
            score_A <- lastrow$score_A + lastrow$scored
            score_B <- lastrow$score_B
            decisiveWIN <- get_decisive_bool(score_A, score_B, round, team,
                1)$WIN
           decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team,
               1)$LOSE
            advantage <- get_advantage(score_A,score_B,team)
            }else{
            team <- 'B'
            score_A <- lastrow$score_A
            score_B <- lastrow$score_B + lastrow$scored
            decisiveWIN <- get_decisive_bool(score_A, score_B, round, team,
                1)$WIN
           decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team,
               1)$LOSE
            advantage <- get_advantage(score_A,score_B,team)
            }
          }
        }
      }
      if(rule == 'CU'){
        if(nrow(temp_df) == 1){
          team <- 'B'
          score_A <- lastrow$scored
          score_B <- 0
          decisiveWIN <- 0
          decisiveLOSE <- 0
          advantage <- FALSE
        }else{
          prelastrow <- temp_df[nrow(temp_df)-1,]
          if(lastrow$round == round){
            if(lastrow$team == 'A'){
              team <- 'B'
              score_A <- lastrow$score_A + lastrow$scored
              score_B <- lastrow$score_B
              decisiveWIN <- get_decisive_bool(score_A, score_B, round, team,
                  2)$WIN
              decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team,
```

```
        2)$LOSE
    advantage <- get_advantage(score_A,score_B,team)
  }else{
    team <- 'A'
    score_A <- lastrow$score_A
    score_B <- lastrow$score_B + lastrow$scored
    decisiveWIN <- get_decisive_bool(score_A, score_B, round, team,
        2)$WIN
    decisiveLOSE <- get_decisive_bool(score_A, score_B, round, team,
        2)$LOSE
    advantage <- get_advantage(score_A,score_B,team)
  }
}else{
  if(lastrow$scored == 1 && prelastrow$scored == 0){
    if(prelastrow$team == 'A'){
      team <- 'A'
      score_A <- lastrow$score_A
      score_B <- lastrow$score_B + lastrow$scored
      decisiveWIN <- get_decisive_bool(score_A, score_B, round, team
          , 1)$WIN
      decisiveLOSE <- get_decisive_bool(score_A, score_B, round,
          team, 1)$LOSE
      advantage <- get_advantage(score_A,score_B,team)
    }else{
      team <- 'B'
      score_A <- lastrow$score_A + lastrow$scored
      score_B <- lastrow$score_B
      decisiveWIN <- get_decisive_bool(score_A, score_B, round, team
          , 1)$WIN
      decisiveLOSE <- get_decisive_bool(score_A, score_B, round,
          team, 1)$LOSE
      advantage <- get_advantage(score_A,score_B,team)
    }
  } else {
    if(prelastrow$team == 'A'){
      team <- 'B'
      score_A <- lastrow$score_A
      score_B <- lastrow$score_B + lastrow$scored
      decisiveWIN <- get_decisive_bool(score_A, score_B, round, team
          , 1)$WIN
      decisiveLOSE <- get_decisive_bool(score_A, score_B, round,
          team, 1)$LOSE
      advantage <- get_advantage(score_A,score_B,team)
    }else{
      team <- 'A'
      score_A <- lastrow$score_A + lastrow$scored
      score_B <- lastrow$score_B
      decisiveWIN <- get_decisive_bool(score_A, score_B, round, team
          , 1)$WIN
      decisiveLOSE <- get_decisive_bool(score_A, score_B, round,
```

```
                    team, 1)$LOSE
                advantage <- get_advantage(score_A,score_B,team)
              }
            }
          }
        }
      }
    }
    temp_row <- data.frame(match_ID = id,
                           team = team,
                           round = round,
                           score_A = score_A,
                           score_B = score_B,
                           decisiveWIN = decisiveWIN,
                           decisiveLOSE = decisiveLOSE,
                           advantage = advantage,
                           attendees = attendees
                           )

    scored <- ifelse(runif(1, min=0, max=1000) < predict(model, newdata =
        get_shot_pos(temp_df[nrow(temp_df),],temp_row), type = "response")
        *1000, 1,0)[[1]]
    sim_shots <<- rbind(sim_shots, data.frame(match_ID = id,
                                              team = team,
                                              round = round,
                                              score_A = score_A,
                                              score_B = score_B,
                                              scored = scored,
                                              decisiveWIN = decisiveWIN,
                                              decisiveLOSE = decisiveLOSE,
                                              advantage = advantage,
                                              attendees = attendees))
    if(decisiveWIN & scored){
      won <- TRUE
      winner <- team
    } else {
      if(decisiveLOSE & !scored){
        won <- TRUE
        winner <- ifelse(team == 'A','B','A')
      } else {
        won <- FALSE
        winner <- 'NA'
      }
    }
    return(data.frame(won = won, winner = winner))
}

sim_shots <<- data.frame(match_ID = integer(), team = character(), round =
    integer(), score_A = integer(),
                                              score_B = integer(), scored =
```

```
                                                integer(), decisiveWIN =
                                                integer(), decisiveLOSE =
                                                integer(), advantage =
                                                logical(), attendees =
                                                integer())

sim_match <<- data.frame(ID = integer(), winner = character(), score_A =
    integer(), score_B = integer(), rounds = integer(), attendees = integer
    ())

mean_attendees <- mean(clean_match_db$attendees, na.rm = TRUE)

sd_attendees <- sd(clean_match_db$attendees, na.rm = TRUE)

matches <- 10000
mechanism <- 'ACU'
model <- probit2
prev <- 0
for (id in 1:matches){
  generate_match(mechanism, model, id, round(abs(rnorm(1, mean=
      mean_attendees, sd=sd_attendees)),0))
  now <- round(id/matches * 100, digits = 0)
  if(prev < now){
    print(now)
    prev <- now
  }
}

sim_shots_ACU_probit2 <- sim_shots
sim_match_ACU_probit2 <- sim_match
print("ok")

```
##Tab:5.1 - General analysis of the simulation
```{r}
going_first_prop <- function(shots){
  for(j in 1:nrow(shots)){
    if(j == 1){
      temp_df <- shots$team[j]
    } else {
      if(shots$round[j] != shots$round[j-1]){
        temp_df <- rbind(temp_df,shots$team[j])
      }
    }
  }
  retvar <- proportions(table(temp_df))
  print(retvar)
  return(retvar)
}
```

```
sim_results <- data.frame(
  Mechanism = character(),
  Model = character(),
  A = numeric(),
  p = numeric(),
  q = numeric(),
  pred_p = numeric(),
  pred_q = numeric(),
  decisiveLOSE_A = numeric(),
  advantageTRUE_A = numeric(),
  going_first_A = numeric(),
  Phase_1 = integer(),
  Phase_2 = integer(),
  Phase_1_A = numeric(),
  Phase_2_A = numeric()
)
mechanisms <- c('ABAB','ABBA','CU','ACU')
shots_sims <- list(sim_shots_ABAB_logit2, sim_shots_ABBA_logit2,
    sim_shots_CU_logit2, sim_shots_ACU_logit2, sim_shots_ABAB_probit2,
    sim_shots_ABBA_probit2, sim_shots_CU_probit2, sim_shots_ACU_probit2)
i <- 0
for(sim in list(sim_match_ABAB_logit2, sim_match_ABBA_logit2,
    sim_match_CU_logit2, sim_match_ACU_logit2, sim_match_ABAB_probit2,
    sim_match_ABBA_probit2, sim_match_CU_probit2, sim_match_ACU_probit2)){
  i <- i + 1
  sim_results <- rbind(sim_results,data.frame(
    Mechanism = mechanisms[((i-1) %% 4)+1],
    Model = ifelse(i <= 4, 'logit','probit'),
    A = proportions(table(sim$winner))['A'],
    p = proportions(table(subset(shots_sims[i][[1]], team == 'A')$scored))
        ['1'],
    q = proportions(table(subset(shots_sims[i][[1]], team == 'B')$scored))
        ['1'],
    pred_p = mean(subset(subset(shots_sims[i][[1]], team == 'A'),match_ID %
        in% subset(sim, rounds > 5)$match_ID)$pred),
    pred_q = mean(subset(subset(shots_sims[i][[1]], team == 'B'),match_ID %
        in% subset(sim, rounds > 5)$match_ID)$pred),
    decisiveLOSE_A = proportions(table(subset(shots_sims[i][[1]],
        decisiveLOSE == 1)$team))['A'],
    advantageTRUE_A = proportions(table(subset(shots_sims[i][[1]], advantage
         == TRUE)$team))['A'],
    going_first_A = going_first_prop(shots_sims[i][[1]])['A'],
    Phase_1 = nrow(subset(sim, rounds <= 5)),
    Phase_2 = nrow(subset(sim, rounds > 5)),
    Phase_1_A = proportions(table(subset(sim, rounds <= 5)$winner))['A'],
    Phase_2_A = proportions(table(subset(sim, rounds > 5)$winner))['A']
  ))
}
```

```
##Tab:5.2-3 - Exploring phase 1 and 2
```{r}
sim_shots_ABAB_logit2$pred <- predict(logit2, newdata =
    sim_shots_ABAB_logit2, type = "response")
sim_shots_ABAB_probit2$pred <- predict(probit2, newdata =
    sim_shots_ABAB_probit2, type = "response")
sim_shots_ABBA_logit2$pred <- predict(logit2, newdata =
    sim_shots_ABBA_logit2, type = "response")
sim_shots_ABBA_probit2$pred <- predict(probit2, newdata =
    sim_shots_ABBA_probit2, type = "response")
sim_shots_CU_logit2$pred <- predict(logit2, newdata = sim_shots_CU_logit2,
    type = "response")
sim_shots_CU_probit2$pred <- predict(probit2, newdata = sim_shots_CU_probit2
    , type = "response")
sim_shots_ACU_logit2$pred <- predict(logit2, newdata = sim_shots_ACU_logit2,
     type = "response")
sim_shots_ACU_probit2$pred <- predict(probit2, newdata =
    sim_shots_ACU_probit2, type = "response")

sim_results_1 <- data.frame(
  Mechanism = character(),
  Model = character(),
  A = numeric(),
  p = numeric(),
  q = numeric(),
  decisiveLOSE_A = numeric(),
  advantageTRUE_A = numeric(),
  going_first_A = numeric()
)
shots_sims_ACU <- list(sim_shots_ABAB_logit2, sim_shots_ABBA_logit2,
    sim_shots_CU_logit2, sim_shots_ACU_logit2, sim_shots_ABAB_probit2,
    sim_shots_ABBA_probit2, sim_shots_CU_probit2, sim_shots_ACU_probit2)
i <- 0
for(simi in list(sim_match_ABAB_logit2, sim_match_ABBA_logit2,
    sim_match_CU_logit2, sim_match_ACU_logit2, sim_match_ABAB_probit2,
    sim_match_ABBA_probit2, sim_match_CU_probit2, sim_match_ACU_probit2)){
  i <- i + 1
  ssacu <- subset(shots_sims_ACU[i][[1]], round <= 5)
  sim_results_1 <- rbind(sim_results_1,data.frame(
    Mechanism = mechanisms[((i-1) %% 4)+1],
    Model = ifelse(i <= 4, 'logit','probit'),
    A = proportions(table(subset(simi, rounds <= 5)$winner))['A'],
    p = proportions(table(subset(ssacu, team == 'A')$scored))['1'],
    q = proportions(table(subset(ssacu, team == 'B')$scored))['1'],
    decisiveLOSE_A = proportions(table(subset(ssacu, decisiveLOSE == 1)$team
        ))['A'],
    advantageTRUE_A = proportions(table(subset(ssacu, advantage == TRUE)
        $team))['A'],
```
```

```
    going_first_A = going_first_prop(ssacu)['A']
  ))
}

sim_results_2 <- data.frame(
  Mechanism = character(),
  Model = character(),
  A = numeric(),
  p = numeric(),
  q = numeric(),
  decisiveLOSE_A = numeric(),
  advantageTRUE_A = numeric(),
  going_first_A = numeric()
)
i <- 0
for(simi in list(sim_match_ABAB_logit2, sim_match_ABBA_logit2,
    sim_match_CU_logit2, sim_match_ACU_logit2, sim_match_ABAB_probit2,
    sim_match_ABBA_probit2, sim_match_CU_probit2, sim_match_ACU_probit2)){
  i <- i + 1
  ssacu <- subset(shots_sims_ACU[i][[1]], round > 5)
  sim_results_2 <- rbind(sim_results_2,data.frame(
    Mechanism = mechanisms[((i-1) %% 4)+1],
    Model = ifelse(i <= 4, 'logit','probit'),
    A = proportions(table(subset(simi, rounds > 5)$winner))['A'],
    p = proportions(table(subset(ssacu, team == 'A')$scored))['1'],
    q = proportions(table(subset(ssacu, team == 'B')$scored))['1'],
    decisiveLOSE_A = proportions(table(subset(ssacu, decisiveLOSE == 1)$team
        ))['A'],
    advantageTRUE_A = proportions(table(subset(ssacu, advantage == TRUE)
        $team))['A'],
    going_first_A = going_first_prop(ssacu)['A']
  ))
}

```

----------Exporting the datasets----------

```{r}
write.table(file = "clean_match.csv",clean_match_db, sep = ";")
write.table(file = "clean_shots.csv",clean_shots_db, sep = ";")
write.table(file = "metadata.csv",metadata, sep = ";")
write.table(file = "metadata_stages.csv",metadata_stages, sep = ";")
write.table(file = "metadata_sub.csv",metadata_sub, sep = ";")
write.table(file = "logit2.csv",logit2$coefficients, sep = ";")
write.table(file = "probit2.csv",probit2$coefficients, sep = ";")
write.table(file = "pred_props.csv",pred_props, sep = ";")
```

```
write.table(file = "sim_results_1.csv",sim_results_1, sep = ";")
write.table(file = "sim_results_2.csv",sim_results_2, sep = ";")
```