UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

LEONARDO MONTEIRO DE ALMEIDA

PREVISÃO DE RESULTADOS DE PARTIDAS DA SÉRIE A DO CAMPEONATO BRASILEIRO DE FUTEBOL: APLICAÇÕES DO MODELO DE POISSON

PORTO ALEGRE
2024

Leonardo Monteiro de Almeida

PREVISÃO DE RESULTADOS DE PARTIDAS DA SÉRIE A DO CAMPEONATO BRASILEIRO DE FUTEBOL: APLICAÇÕES DO MODELO DE POISSON

Trabalho de Conclusão de Curso apresentado à comissão de Graduação do Departamento de Estatística da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Orientador: Prof. Dr. Rodrigo Citton Padilha dos Reis

Porto Alegre
2024

(A ficha de catalogação vai no verso da folha de rosto - Item opcional)

(O Sistema para Geração Automática de Ficha Catalográfica de Teses, Dissertações e Trabalhos de Conclusão de Curso http://sabi.ufrgs.br/servicos/publicoBC/ficha.php)

CIP - CATALOGAÇÃO NA PUBLICAÇÃO
Instituto de Matemática e Estatística
Departamento

Leonardo Monteiro de Almeida

PREVISÃO DE RESULTADOS DE PARTIDAS DA SÉRIE A DO CAMPEONATO BRASILEIRO DE FUTEBOL: APLICAÇÕES DO MODELO DE POISSON

Trabalho de Conclusão de Curso apresentado à comissão de Graduação do Departamento de Estatística da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Porto Alegre, __ de fevereiro de 2024

Resultado:

BANCA EXAMINADORA:

_____

Profa. Dra. Márcia Helena Barbian

Universidade Federal do Rio Grande do Sul

_____

Prof. Dr. Rodrigo Citton Padilha dos Reis

Universidade Federal do Rio Grande do Sul

AGRADECIMENTOS

Primeiramente agradeço aos meus pais por serem a base da minha vida, me incentivando a sempre buscar pelo melhor, e à minha avó Gloria e dinda Andreia por todo apoio e carinho que sempre me encheu de alegria.

Aos meus amigos, pelo incentivo e momentos divertidos.

Ao meu professor e orientador Rodrigo Citton do Reis, por toda a atenção, suporte e orientação que me levaram a conseguir realizar este trabalho.

Também gostaria de expressar meu profundo agradecimento para uma pessoa em especial, que não apenas caminhou ao meu lado, mas também iluminou cada etapa desta jornada acadêmica: o amor da minha vida, Letícia Palmeiro Rubim.

DEDICATÓRIA

Dedico este trabalho à mulher da minha vida Letícia Palmeiro Rubim.

Quero começar esta dedicatória expressando o quanto sou grato por todo o seu apoio ao longo dessa jornada acadêmica.

Deixo documentado aqui o quão importante tu és para mim. Te conhecer foi acontecimento que faltava para eu conseguir enxergar no horizonte o fim dessa jornada. Agradeço pela tua paciência, compreensão e amor. Este trabalho não seria completo sem reconhecer que tu és não apenas a musa inspiradora por trás do meu sucesso, mas também a pessoa incrível que me faz querer evoluir a cada dia.

Com todo o meu amor, Leonardo Monteiro de Almeida.

RESUMO

Este trabalho teve por objetivo abordar a tentativa de previsão de partidas da série do Campeonato Brasileiro de Futebol, analisando dados disponíveis das partidas que foram previamente jogadas por cada time. Portanto, para tentar realizar a previsão destes resultados, foi utilizado o modelo de Maher, que supõe que a distribuição de gols de cada time segue uma distribuição de Poisson, dada a força de ataque e de defesa e o fator de vantagem local como algumas das variáveis a serem consideradas. Primeiramente, foram descritos os diferentes modelos já utilizados para prever as partidas de futebol, com maior enfoque no modelo apresentado por Maher. Em seguida, será utilizado um conjunto de dados obtidos a partir das partidas do Campeonato Brasileiro de futebol masculino de 2023 da Série A. O software R foi utilizado. Verificou-se que a distribuição de Poisson foi adequada para modelar os dados e, por fim, de realizar previsões acertadas em relação às vitórias, derrotas e empates, mas não em relação à quantidade de gols dos participantes da partida. Por isso, novas pesquisas com outros softwares podem proporcionar previsões mais acuradas quanto ao número de gols.

Palavras-chave: Futebol; Distribuição de Poisson; previsões; Estatística.

ABSTRACT

This work aimed to address the attempt to predict matches in the Brazilian football championship series, analyzing available data from matches that were previously played by each team. Therefore, in order to try to predict these results, the Maher model was used, which assumes that the distribution of goals for each team follows a Poisson distribution, given the strength of attack and defense and the local advantage factor as some of the variables to be considered. Firstly, the different models already used to predict football matches were described, with greater focus on the model presented by Maher. Next, a set of data obtained from the matches of the 2023 Brazilian men's football championship Series A will be used. The R software was used. It was found that the Poisson distribution was adequate to model the data and, ultimately, to make accurate predictions in relation to wins, losses and draws, but not in relation to the number of goals scored by participants in the match. Therefore, further research with other software models can provide more accurate predictions regarding the number of goals.

Keywords: Soccer; Poisson Distribution; predictions; Statistics.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Desde aproximadamente o século XVII, o futebol se mostra um esporte popular entre todas as classes, seja como brincadeira de crianças, seja como de forma profissional entre os mais abastados (Reis, 2005). É, portanto, possível dizer que este esporte faz parte da cultura não somente brasileira, como também mundial. No momento atual, o futebol é o esporte mais popular do mundo (Statistics and Data, 2020), podendo ser chamado de “esporte rei”, pela sua massiva/expressiva quantidade de adeptos. Esporte este que conta com aproximadamente 4 bilhões de torcedores, e faz com que mais de 250 (duzentos e cinquenta) milhões de pessoas estejam ativamente envolvidas em atividades relacionadas às suas partidas.

De acordo com um relatório apresentado pela Confederação Brasileira de Futebol em 2019, a indústria futebolística movimentou R$ 48,8 bilhões de reais na economia brasileira em 2018, movimentando, ao todo, aproximadamente R$ 52,9 bilhões de reais, gerando um impacto de 0,72% no Produto Interno Bruto (PIB) do Brasil. Portanto, é possível dizer que o setor esportivo tem um impacto financeiro, direto e indireto, significativo na economia.

A popularização das casas de apostas online, de forma rápida e desenfreada, fez com que o futebol deixasse de ser tão somente um meio de entretenimento para as pessoas ordinárias/comuns, e passasse a ser um negócio. Assim, não somente grandes empresários se tornaram investidores em times, como também aqueles que se interessam em apostar na vitória, derrota ou empate de algum time em determinado jogo. Ou seja, o futebol ultrapassa o jogo em si. Deste modo, não é difícil, ao navegar na *internet*, encontrar diversos *sites* que propõem realizar previsões dos resultados das partidas, oferecendo leituras de probabilidades para que as pessoas façam apostas. As estatísticas oferecidas, em sua grande maioria das vezes, se baseiam em optar por apostar em times com bons desempenhos; contudo, a performance de um time em campo não somente depende do talento de seus jogadores.

Dessa forma, ao longo dos últimos dez anos, este ramo tem se mostrado um tema de grande interesse de pesquisa. Devido ao alto investimento de mercado, o futebol se tornou um negócio muito lucrativo (Gasparetto, 2013), e aí é que se torna interessante falar sobre como é possível detectar um padrão de sequência de partidas para cada time e realizar previsões dos seus resultados.

A Distribuição de Poisson se trata de uma distribuição de variável aleatória discreta muito utilizada pelo campo matemático e estatístico para calcular a probabilidade de algo acontecer dentro de um determinado intervalo. No caso, “esse intervalo contínuo de observação pode ser um intervalo qualquer em que se vai observar a ocorrência dos sucessos, como, por exemplo, um intervalo de tempo, de comprimento, de área ou de volume” (Ara, A. B.; Musetti, A. V.; Schneiderman, B., 2003, p. 38).

Muitas têm sido as aplicações da Distribuição de Poisson, tais como na previsão da quantidade de veículos que passam por um pedágio de rodovia em certo momento do dia durante a semana, na quantidade de colisões de carros em um cruzamento determinado sob condições específicas, o número diário de novos casos de câncer de estômago, entre outras situações. Se percebermos uma série de partidas de um campeonato de futebol como uma sequência de variáveis aleatórias, em tempo discreto, temos uma potencial aplicação para a distribuição de Poisson, ao analisar a modelagem de gols marcados na partida por um time A.

1.1 Objetivos

O presente trabalho tem como objetivo principal realizar inferência estatística utilizando-se da Distribuição de Poisson aplicada a dados discretos, na previsão dos resultados das partidas dos jogos de futebol da série A do Campeonato Brasileiro de Futebol do ano de 2023. De maneira mais específica, o modelo apresentado por Maher (1982), que utiliza a distribuição de Poisson, é empregado para realizar previsões de resultados de partidas futuras.

1.2 Organização do trabalho

O restante deste trabalho está organizado da seguinte forma. No Capítulo 2 são descritos os métodos utilizados para modelar dados de partidas de futebol. O enfoque é dado para o modelo de Maher. O conjunto de dados utilizado para a realização da previsão dos resultados das partidas também é descrito neste capítulo. No Capítulo 3 são apresentados os resultados da análise dos dados. Por fim, o Capítulo 4 apresenta as conclusões.

2 MÉTODOS

O método estatístico escolhido para ser utilizado no desenvolvimento deste trabalho foi testando a Distribuição de Poisson, usada pelo modelo de Maher (1982), que demonstrou como utilizá-lo nas previsões futebolísticas. Serão aplicadas variáveis da amostra do banco de dados, no caso, os jogos do Campeonato Brasileiro de Futebol, em uma função para a otimizar, ou seja, tentaremos maximizar a função de verossimilhança dentro destes parâmetros.

Os dados foram observados a partir dos resultados dos jogos do Campeonato Brasileiro de Futebol na era de pontos corridos de 2003 até 2023. No teste x^2 (qui-quadrado) será verificado se a população de dados observados segue esta distribuição, não rejeitando a hipótese nula de que os dados seguem uma distribuição de Poisson se os valores de *p* se apresentarem < 0,05.

2.1 Modelos de previsão de partidas de futebol

Para que sejam analisados os resultados que se pretende alcançar neste trabalho, será feita uma inferência estatística, que consiste na retirada de conclusões a partir dos dados coletados, ou seja, será testada a hipótese a partir das amostras coletadas da população de interesse (Zanetta, 2017). Serão testadas no modelo as estimativas de chances de ganhar de um time com base na distribuição de gols feitos na partida.

Dentro do campo matemático e estatístico, é possível observar os chamados processos estocásticos, comumente utilizados como método para estudar os fenômenos aleatórios e suas ditas trajetórias evolutivas no campo da probabilidade; ou seja, existem variações imprevisíveis sobre um determinado tempo enquanto ele passa (Paula, 2020). Nas palavras de Cobre (2005), “são um conjunto de todas as possíveis trajetórias de um certo processo”. Ainda, de acordo com Souza (2013), “portanto, um processo estocástico nada mais é que uma coleção de variáveis aleatórias que descrevem o comportamento de algum processo com o passar do tempo.” Dito isto, percebe-se que se trata de uma família de variáveis aleatórias[1].

Estas variáveis aleatórias dos processos estocásticos estão todas na mesma condição: valoradas no mesmo espaço S. Isto é, todas assumem valores no mesmo

1“Podemos considerar funções que associam números reais aos eventos de um espaço amostral. Tais funções são ditas “variáveis aleatórias”. Isso equivale a descrever os resultados de um experimento aleatório por meio de números ao invés de palavras, o que apresenta a vantagem de possibilitar melhor tratamento matemático [...]” (Costa Neto, 1939, p. 29).

espaço. Uma variável aleatória é uma função, que está definida em um conjunto em que se encontram os dados no espaço amostral.

Um processo estocástico é representado pelo símbolo $\{X_{(t)}, t \in T\}$, e, considerando que se refere a uma família, para cada t, tem-se uma variável aleatória X, sendo representado por $X_t : U \rightarrow S$. U diz respeito ao conjunto dos estados no espaço amostral, e S ao espaço dos resultados dos valores. Para classificá-los, é necessário analisar o espaço de estados, a natureza do conjunto T e as características estatísticas das variáveis aleatórias que os definem (Alves; Delgado, 1997). Estes processos podem ser classificados[2] em estado discreto quando o conjunto de índices for finito (X = {0,1,2,...}; assim, $X(_t)$ é uma cadeia, e podem ser classificados como: em estado contínuo quando não finito; no tempo discreto quanto t é enumerável; e em contínuo sendo o caso contrário do discreto.

Desta maneira, a fim de conseguir prever o resultado de uma partida, utilizou-se da modelagem estatística, que é um conjunto de técnicas probabilísticas aplicadas, pois são capazes de estruturar e organizar informações em um sistema. Desta forma, é possível dizer que "a modelagem pode ser representada como um modelo matemático, que é uma representação simplificada utilizando-se de conjuntos de símbolos e de relações matemáticas, tendo como finalidade representar um fenômeno ou um problema de uma situação real." (Mafalda *et al*., 2016, p. 2).

Ainda, de acordo com Barbosa (2001, p. 29), os modelos estatísticos "(...) são considerados como um meio de indagar e questionar situações reais por meio de métodos matemáticos, evidenciando o caráter cultural e social da matemática". E, nas palavras de Biembengut e Hein (2003), este método é "uma arte, ao formular e elaborar expressões que valham não apenas para uma solução particular, mas que também sirvam, posteriormente, como suporte para outras aplicações e teorias."

Ou seja, em suma, os modelos estatísticos servem, neste caso, para tentar representar a realidade de uma forma mais simples, aplicando métodos de probabilidade para tentar evidenciar as variáveis observadas e como elas se

2 "Os processos estocásticos podem ser classificados tanto de acordo com os valores que ele assume quanto de acordo com os valores que seu parâmetro pode assumir. Assim, um processo estocástico que toma valores em um conjunto discreto de pontos do eixo real é dito um processo estocástico discreto. Por outro lado, se o processo toma valores em uniões de subconjuntos contínuos de R, ele é considerado um processo estocástico contínuo. De maneira análoga, processos cujo parâmetro toma valores em uniões de subconjuntos contínuos de R são ditos processos estocásticos de parâmetro contínuo". (ALBUQUERQUE, J. P. de A. e. **Probabilidades, variáveis aleatórias e processos estocásticos**. Rio de Janeiro: Interciência, 2008. p. 225.)

comportam e relacionam, a fim de evidenciar os resultados dos jogos. Nas palavras de Neoway (2022), servem para, em suma, “criar um modelo que consiga descrever os elementos mais importantes para uma análise”.

Neste campo, existem diversos modelos variados que foram testados para serem utilizados na previsão dos resultados das partidas de futebol. Lee (1987) também propôs um modelo linear generalizado com base na Distribuição de Poisson, analisando os gols marcados e sofridos por cada equipe, havendo independência entre estes gols. Nele se levou em consideração o poder de ataque, capacidade de defesa e o fator local (Santos, 2019, p. 25).

Um processo estocástico de Poisson é um “processo de contagem de eventos aleatórios pontuais” (Pires, 2023, p. 10), ou seja, tem a característica de conseguirem efetuar os referidos saltos discretos, contudo, de forma não frequente ao longo do tempo. Desta forma, há de se levar em consideração que a Distribuição de Poisson mostra a probabilidade[3] de se conseguir um resultado considerando a aleatoriedade dos eventos, mas sem deixar de fazer associação a uma taxa média.

Para testarmos este modelo, ele seguirá necessariamente uma Distribuição de Poisson, representada pela seguinte função de probabilidade f(x) = $P(X=k)=\frac{e^{-\lambda}\lambda^{k}}{k!}$, em que X é uma variável aleatória discreta que pode assumir os valores 0, 1, 2,... Ainda, fazem-se necessários alguns requisitos: que a variável aleatória X seja o nº de ocorrências de um evento em um intervalo; que as ocorrências sejam aleatórias; que as ocorrências sejam independentes uma das outras; que as ocorrências tenham a mesma probabilidade sobre o intervalo considerado (Portnoi, 2006, p. 8).

Existem propriedades que também são importantes quando falamos sobre Poisson, sendo elas: a média, variância, desvio padrão, coeficiente de assimetria, coeficiente de curtose, função geratriz de momentos e função características (Spiegel *et al*., 2009, p. 111). Quando falamos sobre a distribuição de Poisson, é importante, também, referir a distribuição Binomial[4], isto porque, na distribuição

3De acordo com Kovács, Poisson se explica do seguinte modo: “a probabilidade de um ocorrer um único evento neste intervalo é diretamente proporcional a duração do mesmo, através de um fator de proporcionalidade λ(t), enquanto que a probabilidade de ocorrerem no mesmo intervalo dois ou mais eventos é zero.” (Kovács, 1996, p. 76)

4 “Suponha que tenhamos um experimento como lançar uma moeda ou um dado repetidamente ou escolher uma bola de gude de uma urna repetidamente. Cada lançamento ou seleção é chamado de um *ensaio*. Em cada ensaio, haverá uma probabilidade associada a um evento particular, como cara no caso da moeda, 4 no caso do dado ou seleção de uma bola de gude vermelha. Em alguns casos,

binomial (1), se *N* for grande, dada a probabilidade *p* do evento ocorrer for próximo a 0, de forma que *q* = (1 - *p*) tende para 1, o evento será raro caso possua pelo menos tentativas que sejam iguais a 50 (*N*⩾ 50). Assim, a distribuição binomial se aproxima consideravelmente de Poisson (Spiegel, 1985, p. 156).

2.1.1 Especificação do Modelo de Maher

De acordo com o modelo apresentado por Maher (1982), existem motivos fundamentados para pensar que o número de gols marcados por um time em uma partida de futebol venha a ser uma distribuição de Poisson variável; a posse de bola de um time durante o jogo segue a lógica de que a probabilidade *p* que o ataque do time irá resultar em gol é pequena, mas aumenta pela quantidade de vezes que determinado time estivesse em posse de bola ao longo da partida. Assim, se *p* é a constante e o ataque é independente, o número de gols será binomial. A média desta distribuição de Poisson irá variar de acordo com a qualidade do time e considerará a quantidade de gols marcados por todos os times.

Ao longo do artigo, o autor propôs a utilização de um modelo independente de Poisson para as pontuações dos times. De modo simplificado, em seu artigo ele considera que os gols marcados na partida seguem independentes na distribuição Poisson bivariada, com as médias do ataque e da defesa dos times, utilizando uma possível relação entre os gols feitos por cada elenco que está jogando (Santos, 2019, p. 25).

Em seu trabalho, Maher supôs que se o time *i* está jogando em casa (vantagem do fator local) contra o time *j* e o resultado da partida é (x_{ij}, y_{ij}), deve-se assumir que X_{ij} é uma variável que segue uma distribuição de Poisson com média $\alpha_i\beta_j$, e que Y_{ij} também é uma variável que segue uma distribuição de Poisson com média $\gamma_i\delta_j$ e que X_{ij} e Y_{ij} são independentes. Desta forma, α_i representa a força de ataque do time i quando com a vantagem de jogar em casa, β_j na fraqueza da

---

esta probabilidade não muda de um ensaio para o próximo (como ao lançar uma moeda ou um dado). Tais ensaios são ditos serem independentes e frequentemente são chamados de ensaios de Bernoulli [...] Seja *p* a probabilidade de um evento acontecer em um ensaio de Bernoulli qualquer (chamada a probabilidade de sucesso). Então *q = 1 - p* é a probabilidade do evento vir a não acontecer exatamente *x* vezes em *n* ensaios (i.e., *x* sucessos e *n - x* fracassos ocorrerão) é dada pela função de probabilidade $P(X=k)=\binom{n}{k}p^k(1-p)^{n-k}$ onde a variável aleatória X denota o número de sucessos em *n* ensaios e *x* = 0,1,...,*n*." (SPIEGEL, M. R; SCHILLER, J. J.; SRINIVASAN, R. A. **Schaum's Outline**: Probability and Statistics. 3ª ed. Nova York: The McGraw-Hill Companies, 2009. p. 119)

defesa do time *j* quando jogando fora de casa, γ_i na fraqueza da defesa do time i em casa e δ_j na força de ataque quanto time *j* joga fora.

No caso, seja $X_1, X_2, \ldots, X_n$ gols do time mandante de *n* partidas de futebol de dois times e de uma mesma competição, e seja $Y_1, Y_2, \ldots, Y_n$ os respectivos gols do time visitante de *n* partidas. Ainda, considere que X_i ~ Poisson (λ_m), *i* = 1, …,n e Y_i ~ Poisson (λ_v), de tal forma que $(\lambda_m) = \alpha \ . \ \beta$.

Assim, a restrição imposta pelo modelo para produzir um conjunto único de parâmetros em que seja possível concretizar a previsão suposta, pode-se utilizar a função $\sum_i \alpha_i = \sum_i \beta_i$, e da mesma forma, também é possível utilizar a função $\sum_i \gamma_i = \sum_i \delta_i$, se assumirmos que $\underline{X}$ e $\underline{Y}$ são independentes, a estimação de $\underline{\alpha}$ e $\underline{\beta}$ será a partir de $\underline{x}$ e a estimação de $\underline{\gamma}$ e será $\underline{\delta}$ por meio de $\underline{\gamma}$. Para a pontuação dos times se utiliza a função log de verossimilhança é descrita como sendo:

$$logL(\underline{\alpha}, \underline{\beta}) = \sum_i \sum_{j \neq i} \left(-\alpha_i \beta_j + x_{ij} \log(\alpha_i \beta_j) - \log(x_{ij}!) \right)$$

E, portanto,

$$\frac{\partial logL}{\partial \alpha_i} = \sum_{j \neq i} \left(-\beta_j + \frac{x_{ij}}{\alpha_i} \right)$$

Assim, a estimação de máxima verossimilhança $\hat{\alpha}_i$ e $\hat{\beta}_j$ satisfaz a seguinte função: $\hat{\alpha}_i = \frac{\sum_{j \neq i} x_{ij}}{\sum_{j \neq i} \hat{\beta}_j}$ e $\hat{\beta}_j = \frac{\sum_{i \neq j} x_{ij}}{\sum_{i \neq j} \hat{\alpha}_i}$

2.1.2 Previsão no Modelo de Maher

Maher (1982) analisou também a influência da vantagem do time jogando em casa e a desvantagem do time jogando fora de casa. Em seu trabalho ele realizou a previsão do Campeonato Inglês de Futebol, retirando dados do *Rothmans Football Year-book* dos anos de 1973, 1974 e 1975. Analisou-se 04 divisões do campeonato, com parâmetros medindo a quantidade média de gols dos times mandantes, e medindo a quantidade média de gols do time visitante, utilizando-se da suposição de que a variável resposta segue uma Distribuição de Poisson. Portanto, para conseguir alcançar essa quantidade referida, o autor utilizou, como estimadores, a

quantidade média de gols de mandantes e a quantidade média de gols de visitante, respectivamente. No contexto da função utilizada por Maher, ficaria:

$X_1, X_2, \ldots, X_{n-1}$ e $Y_1, Y_2, \ldots, Y_{n-1}, \bar{X}_n = E\left[X_n \vee X_1, X_2, \ldots, X_{n-1}, Y_1, Y_2, \ldots, Y_{n-1}\right] = \hat{\alpha} + \hat{\beta}$

Para Maher (1982), já que X_{ij} e Y_{ij} assumem-se de que são duas distribuições de Poisson e independentes, então, as probabilidades de que $X_{ij} = x$ e $Y_{ij} = y$ poderiam ser facilmente calculadas repetindo os pares de *i* e *j*. Ao explicar o modelo de Poisson Bivariado, o autor argumenta que uma partida não necessariamente consiste somente dos dois times em campo, e, com isto, os estimadores de máxima verossimilhança com as médias desta distribuição são $\hat{\alpha}_i = \frac{\sum_{j \neq i} \left(x_{ij} + y_{ij}\right)}{\left(1 + \hat{k}^2\right) \sum_{j \neq i} \hat{\beta}_i}$ e

$\hat{\beta}_i = \frac{\sum_{j \neq i} \left(x_{ij} + y_{ij}\right)}{\left(1 + \hat{k}^2\right) \sum_{j \neq i} \hat{\alpha}_i}$ para quaisquer *i,j* e $\hat{k}^2 = \frac{\sum_i \left(\sum_{j \neq i} y_{ij}\right)}{\sum_i \left(\sum_{j \neq i} x_{ij}\right)}$ que segue

$$\sum_i \left(\sum_{j \neq i} \hat{\alpha}_i \hat{\beta}_i\right) = \sum_i \left(\sum_{j \neq i} x_{ij}\right)$$

e

$$\sum_i \left(\sum_{j \neq i} \hat{k}^2 \hat{\alpha}_j \hat{\beta}_j\right) = \sum_i \left(\sum_{j \neq i} y_{ij}\right)$$

, o que demonstra que a soma das médias das Distribuições de Poisson, quando ajustadas, é igual ao número observado de gols marcados.

2.2 Conjunto de dados: Campeonato Brasileiro dos anos 2003-2023

Para o desenvolvimento deste trabalho, os dados foram retirados a partir do site Github (na pasta Willamorim/brasileirao). Foram coletados dados dos anos de 2003 até 2023 da série A do Campeonato Brasileiro de Futebol para serem tratados, e contam com um total de milhares de jogos analisados para realizar a estimação dos parâmetros que foram otimizados. Para realizar as previsões, analisou-se 360 jogos do ano de 2023, até a trigésima sexta rodada, para obter os resultados das rodadas de nº 37 e nº 38.

As variáveis disponíveis neste conjunto de dados são as seguintes:

- season (serve para indicar o ano da temporada da partida);
- date (serve para indicar a data da partida);
- home (time mandante);

- away (time visitante);
- score (placar da partida);
- gols_mandante (variável criada a partir do comando matches$gols_mandante <- as.numeric(sapply(strsplit(matches$score, "x"), `[`, 1)) para extrair, da variável score, apenas a quantidade de gols do mandante);
- gols_visitante (variável criada a partir do comando:

matches$gols_visitante <- as.numeric(sapply(strsplit(matches$score, "x"), `[`, 2)) para extrair, da variável score, apenas a quantidade de gols do visitante.

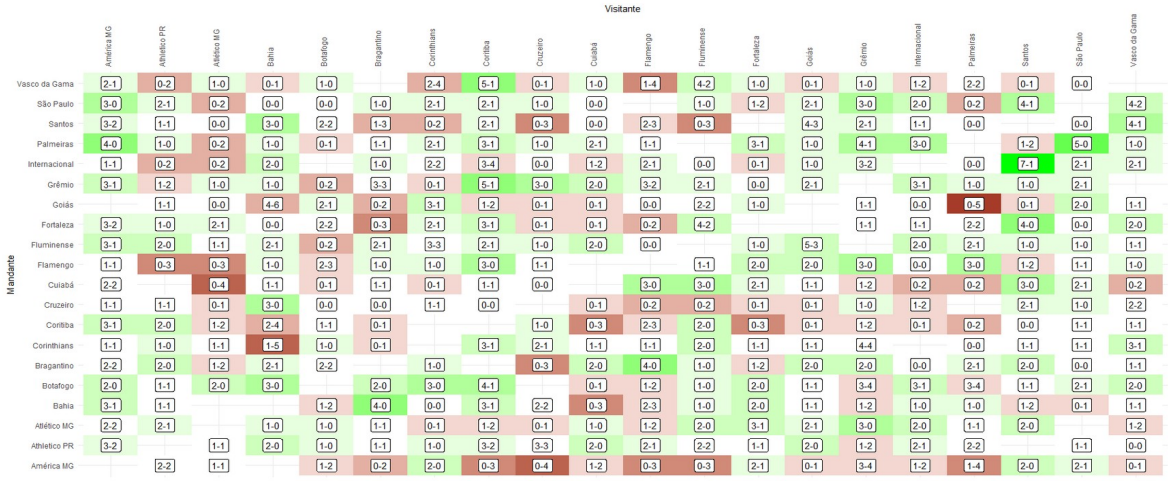
2.3 Implementação computacional

Para o desenvolvimento deste trabalho, foi usado o software R, com apoio do RStudio, por ser capaz de realizar as análises e manipulações dos dados estatísticos. No software, aplicou-se a programação para realizar a otimização dos parâmetros, depuração e execução dos códigos na linguagem R, com o intuito de alcançar as estimações através das equações descritas no Capítulo 2.1.1.

3 RESULTADOS

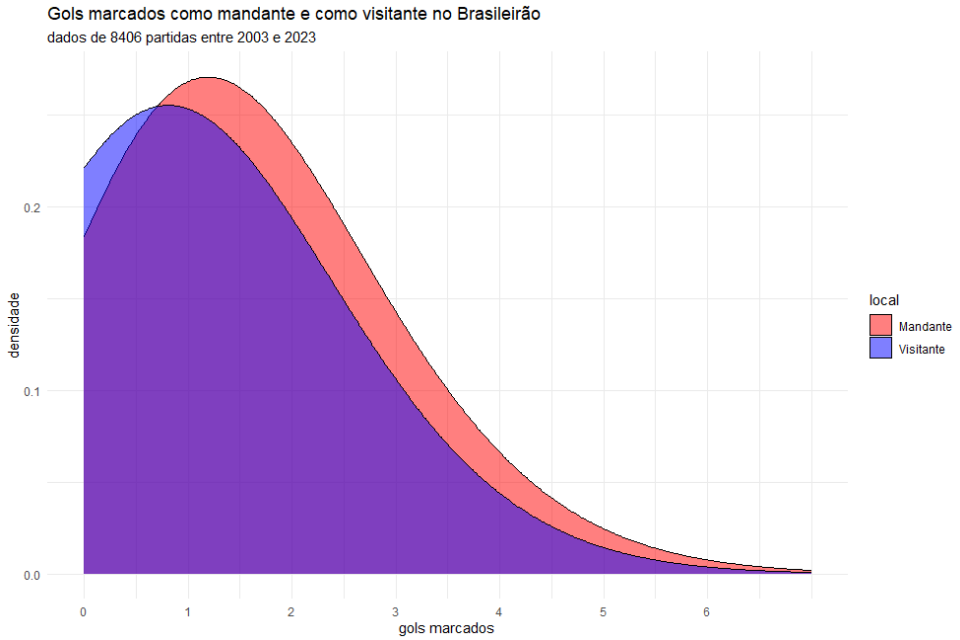
Neste capítulo serão abordados os resultados das análises feitas a partir da modelagem realizada. Como já mencionado, foram analisados os dados do Campeonato Brasileiro de Futebol dos anos de 2003 até 2023, e para a previsão, analisou-se especificamente os jogos da primeira rodada até a trigésima sexta rodada do ano de 2023, conforme apresentado na Figura 1 abaixo:

Figura 1 – vitórias, derrotas e empates com gols marcados no campeonato brasileiro de futebol de 2023, da primeira até a trigésima sexta rodada



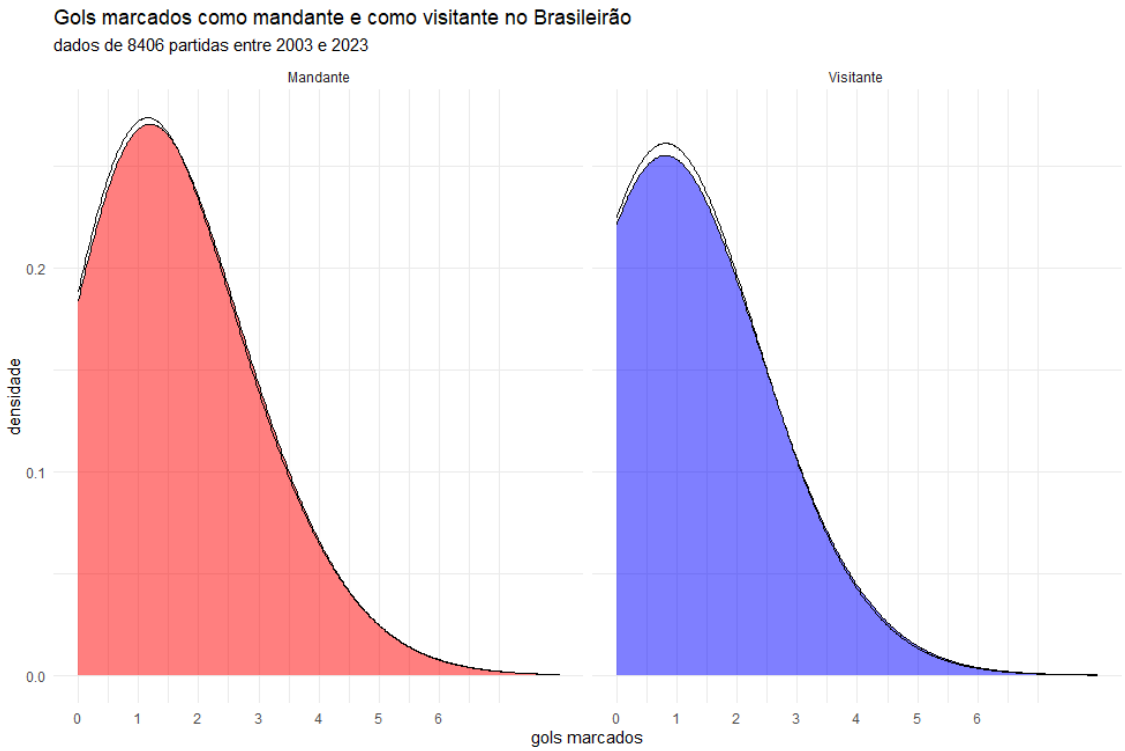
Para entender melhor os dados analisados, verificou-se a distribuição de gols marcados por times mandantes e visitantes no Campeonato Brasileiro de Futebol, conforme a Figura 2 abaixo:

Figura 2 – Distribuição de gols marcados no Campeonato Brasileiro de Futebol entre 2003-2023



Como pode-se ver na Figura 2, temos curvas referentes às quantidades de gols em uma partida, tanto do time visitante quanto do time mandante. No eixo horizontal temos as possíveis quantidades de gols por partida, e no eixo vertical temos as probabilidades com as quais acontecem cada quantidade de gols. Temos que as médias de gols de mandantes e visitantes, por partida, é diferente: mandantes marcam 1,54 gols por partida, enquanto visitantes marcam 1,03 gols por partida. Isso é ilustrado na Figura 3 abaixo:

Figura 3 – Distribuição de gols marcados de mandantes e visitantes



Ao simular dados com 10000 observações distribuídas como uma Poisson e comparando com as curvas de gols de mandantes e de gols de visitantes, temos que os dados parecem ser distribuídos por uma Poisson. Da mesma forma pelo teste *qui-quadrado* em que a hipótese nula diz que certo conjunto de variáveis aleatórias segue uma distribuição de Poisson, temos que os *p*-valores não foram significativos, e, portanto, não rejeitamos a hipótese nula. Para as quantidades de gols por partida dos times mandantes, *p*-valor foi igual a 0,4178 e para as quantidades de gols por partida dos times visitantes foi 0,07846.

Podemos considerar que a diferença entre os valores apresentados dos gols do mandante para os gols do visitante, embora relativamente aproximados, diferenciam-se pelo fator local (vantagem em casa).

Na Tabela 1, abaixo, temos os parâmetros α eβ de cada time, que são as médias de gols a favor e média de gols contra de cada time, respectivamente, ao considerar todos os jogos de 2003 até 2023.

Tabela 1 – Média de gols dos times mandantes e visitantes

Nº	Time	α	β
1	América MG	1.08	2.17
2	Athletico PR	1.33	1.11
3	Atlético MG	1.36	0,75
4	Bahia	1.22	1.36
5	Botafogo	1.58	0.944
6	Bragantino	1.31	0.917
7	Corinthians	1.22	1.28
8	Coritiba	1.14	1.94
9	Cruzeiro	0.944	0.861
10	Cuiabá	1	1.03
11	Flamengo	1.5	1.11
12	Fluminense	1.36	1.19
13	Fortaleza	1.17	1.19
14	Goiás	0.972	1.44
15	Grêmio	1.64	1.5
16	Internacional	1.14	1.19
17	Palmeiras	1.72	0.889
18	Santos	1.06	1.64
19	São Paulo	1.06	1
20	Vasco da Gama	1.08	1.36

Então, foi feita a previsão dos jogos das duas últimas rodadas do Campeonato Brasileiro de Futebol de 2023, fazendo de conta que esses jogos não aconteceram, para saber se o modelo estava adequadamente ajustado e conseguiria acertar o resultado. Aqui estão as previsões dos dois últimos jogos em que o Grêmio estava envolvido levando em consideração o parâmetro alfa

$$\alpha_i = \frac{1}{N} \sum_{i=1}^{N} x$$

Na tabela 2, abaixo, temos o resultado da previsão dos jogos nos quais o Grêmio esteve envolvido:

Tabela 2 – Resultado da previsão dos jogos em que o Grêmio está envolvido levando *α* em consideração

Grêmio-Vasco da Gama	Fluminense-Grêmio
“1. 63888888888889-1. 08333333333333”	“1. 36111111111111-1. 63888888888889”

Foi utilizado o parâmetro de ataque (*α*) para estimar os resultados das partidas, e conclui-se que elas são boas estimativas, relativamente. Porém, é intuitivo pensar que o Grêmio faria mais gols no Vasco, do que no Fluminense. É possível mostrar isso usando o parâmetro *β*, referente a defesa, das equipes.

Na Tabela 3, abaixo, temos o resultado das previsões dos jogos do Grêmio levando em consideração *α* e *β*:

Tabela 3– Resultado das previsões dos jogos em que o Grêmio está envolvido levando *α* e *β* em consideração

Grêmio-Vasco da Gama	Fluminense-Grêmio
“2.231-1.625”	“2.042-1.958”

Agora, levando em consideração o gamma $\gamma = \frac{\sum x}{\sum y}$, este é o resultado da previsão, segundo a Tabela 4:

Tabela 4 – Resultados das previsões dos jogos em que o Grêmio está envolvido levando *α*, *β* e *γ* em consideração

Grêmio-Vasco da Gama	Fluminense-Grêmio
“2.894-1625”	“2.649-1958”

Essas previsões parecem razoáveis, mas sabemos que este é um modelo muito básico. Dessa forma, iremos quantificar para sabermos o quão bom ele é, e para isso usamos os jogos das 36 primeiras rodadas como dados de treino. Se o modelo for bom, ele irá prever placares semelhantes aos observados

O valor esperado da distribuição de Poisson é igual a *λ*, e então podemos substituir como nossos gols previstos e x como os gols reais, e calcular a

probabilidade de esses resultados ocorrerem dados os parâmetros ataque/defesa/fator local que imaginamos serem corretos. Então, fizemos isso para todas as partidas e obtivemos os resultados dos times mandantes e visitantes.

Quando somamos os logs desses valores de probabilidade, tivemos uma medida de quão erradas estavam nossas previsões no caso 1055,49. Para termos uma ideia se isso é bom ou não, executamos rapidamente o modelo com todos os parâmetros definidos como zero, e nesse caso encontramos 1039,942. A pior probabilidade é melhor do que vimos anteriormente, e isso sugere que o modelo é bom.

Com a otimização, nos parâmetros mencionados (*α* e *β* para cada equipe e *γ* para *home advantage)* a tendência é de existir uma minimização das logs de probabilidades negativas, e, portanto, uma maior precisão na previsão dos resultados dos jogos já disputados. Assim como no artigo, a função foi configurada para que cada iteração vá pegando os valores mais baixos a serem alcançados, a fim de encontrar o menor valor.

Para prever os resultados, *α* e *β* foram convertidos em vetores numéricos não listados, e então usados para prever os resultados dos jogos anteriores, como mencionado acima. E, então, fez-se necessário calcular a log de probabilidade dos objetivos observados, dados os objetivos que foram previstos e somá-los. Após, multiplicou-se por -1, já que a soma das probabilidades logarítmicas acabou por ser negativa, e o intuito é chegar o mais perto possível do zero.

As funções para o time *i* e *j* a partida *k* se dão por

$$\left(\lambda_i, \beta_i, \gamma; i \ldots n\right) = \prod_{i=1}^{n} \frac{e^{-\lambda_k} \lambda_k^{x_k}}{x_k!} \frac{e^{-\mu_k} \mu_k^{y_k}}{y_k!}$$

Em que a partida *k* e os times *i* e *j*, gols do time da casa, x é definido como

$$x_k \sim \text{Poisson}\left(\lambda_k = \alpha_{i(k)} \beta_{j(k)} \gamma\right)$$

E para gols dos times visitantes y se dá pela função

$$y_k \sim \text{Poisson}\left(\mu_k = \alpha_{j(k)} \beta_{i(k)}\right)$$

Em suma, o objetivo final é de minimizar o resultado da multiplicação na probabilidade de marcarem x e y no jogo. Essa probabilidade dos gols marcados é considerada uma distribuição de Poisson, controlada pelos parâmetros *α*, *β* e *γ*. No

caso em tela, o parâmetro da distribuição é feito através da multiplicação de α e β; mas, ao invés disso, se for dividido, uma defesa forte irá reduzir o valor de xij/yij, reduzindo a expectativa de gols marcados do time oponente.

$$x_{ij} \sim \text{Poisson}\left(\frac{\alpha_i \gamma}{\beta}\right)$$

$$y_{ij} \sim \text{Poisson}\left(\frac{\alpha_i}{\beta}\right)$$

Em seguida, muda-se como o cálculo da expectativa de gols é feito, dado que

$A = \frac{B \cdot C}{D}$ é igual a

$$A = e^{\log(A) + \log(B) - \log(C)}$$

Converte-se os parâmetros que estão se buscando log(parâmetros) e, daí,pegamos o expoente da soma com os gols previstos. Em caso de algum dos parâmetros ficar negativo, lambda será negativo, o que indicará que os eventos não irão acontecer. Se pegarmos os parâmetros de log ao invés, temos a seguinte função:

$$x_{ij} \sim \text{Poisson}\left(e^{\alpha_i - \beta_j + \gamma}\right)$$

Também foi trocado o algoritmo para acharmos a máxima verossimilhança, de Nelder-Mead para BFGS, que é mais rápido. No entanto, isso requer a minimização da função. E, por último, o cálculo final da máxima verossimilhança é feito corrigindo a soma de todos os parâmetros de ataque e a soma de todos os parâmetros de defesa para equivaler a zero.

Para fazer isso, é possível largar o primeiro parâmetro de ataque ou defesa e então calcular o parâmetro da soma dos restantes multiplicados por -1. Vejamos

$$\alpha_n = -\sum_{i=1}^{n-1} \alpha_i$$

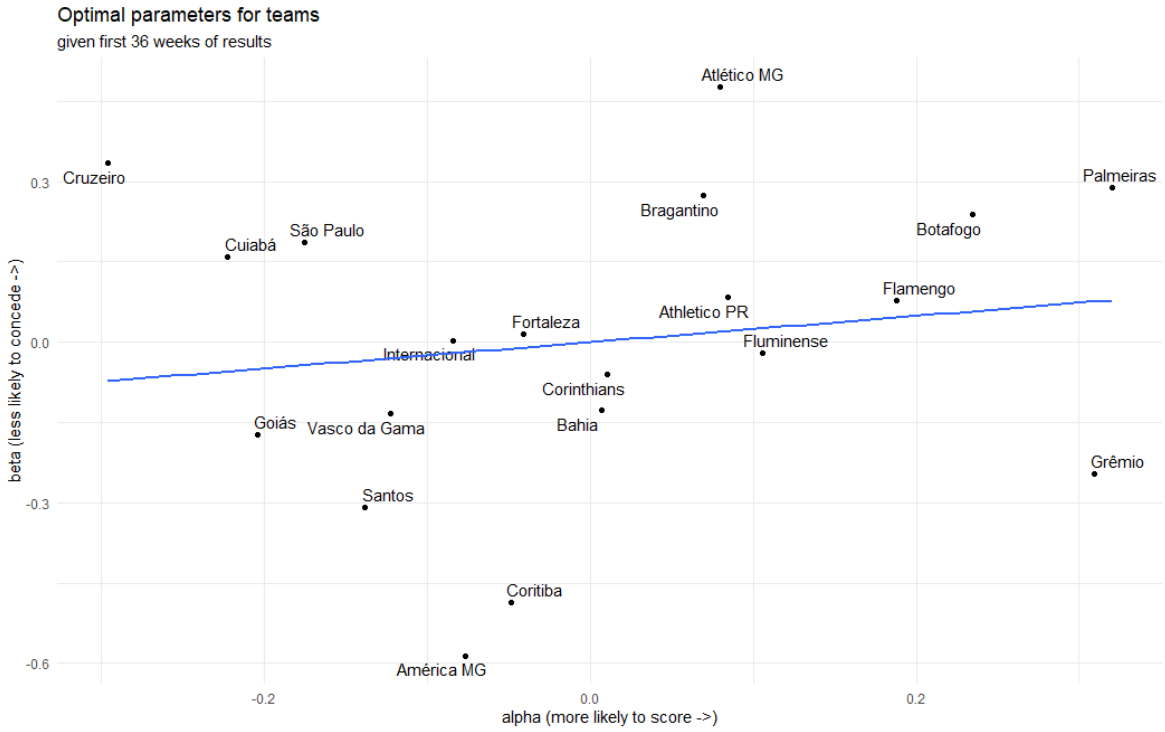
e

$$\beta_n = -\sum_{i=1}^{n-1} \beta_i$$

.

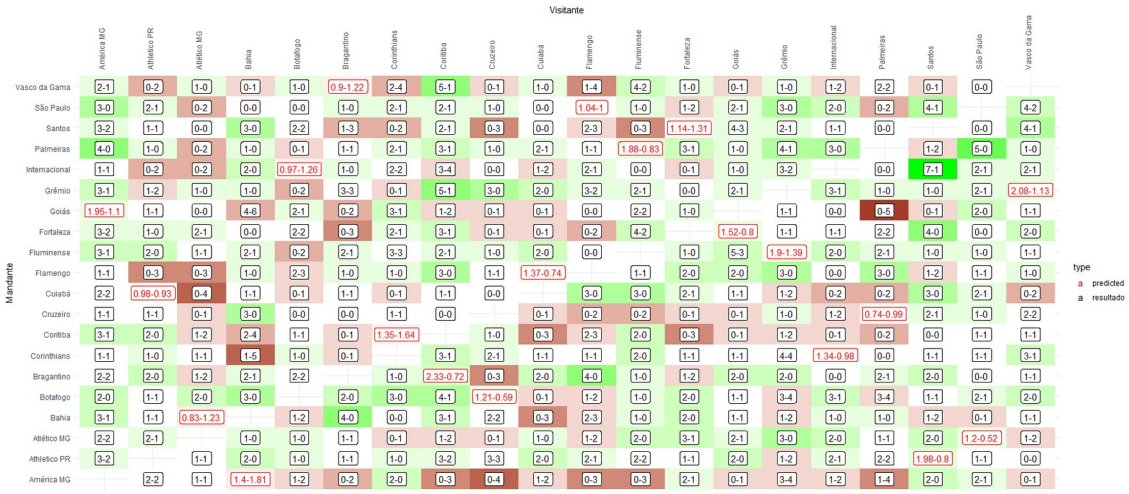
Para estas otimizações, os parâmetros originais são semelhantes aos zerados, conforme o gráfico da Figura 4, abaixo. No gráfico, os parâmetros *α* e *β*para cada time se distribuem, correspondendo o eixo horizontal ao parâmetro *α* e o eixo vertical ao parâmetro *β*:

Figura 4 – Parâmetros otimizados para os times após ajustes



Abaixo, na Figura 5, temos um comparativo dos placares das partidas das rodadas 37 e 38 do Campeonato Brasileiro de Futebol. Em preto, tem-se os resultados dos jogos que supomos que já haviam sido jogados em preto, e, em vermelho, os resultados previstos dos jogos que supomos que não haviam sido jogados ainda:

Figura 5 – Vitórias, derrotas e empates com gols marcados no Campeonato Brasileiro de Futebol de 2023 com as previsões das rodadas 37 e 38



Ao se comparar os resultados reais com os resultados previstos, é possível perceber que houve 12 acertos e 8 erros, de acordo com a Tabela 5. Na Tabela abaixo, “Mgols” e “Vgols” expressam a quantidade de gols reais que ocorreram nas partidas, enquanto “e_Mgols” e “e_Vgols” são as previsões que aparecem em vermelho na Figura 5.

Tabela 5 – Relação de acertos e erros da previsão

Mandante	Visitante	Mgols	Vgols	e_Mgols	e_Vgols	Rodada	Predição
Flamengo	Cuiabá	2	1	1.37	0.74	37	Acertou
Botafogo	Cruzeiro	0	0	1.21	0.59	37	Errou
Palmeiras	Fluminense	1	0	1.88	0.83	37	Acertou
Corinthians	Internacional	1	2	1.34	0.98	37	Errou
Bragantino	Coritiba	1	0	2.33	0.72	37	Acertou
Atlético MG	São Paulo	2	1	1.20	0.52	37	Acertou
Grêmio	Vasco da Gama	1	0	2.08	1.13	37	Acertou
Athletico PR	Santos	3	0	1.98	0.80	37	Acertou
Fortaleza	Goiás	1	0	1.52	0.80	37	Acertou
América MG	Bahia	3	2	1.40	1.81	37	Errou
Fluminense	Grêmio	2	3	1.90	1.39	38	Errou

Vasco da Gama	Bragantino	2	1	0.90	1.22	38	Errou
São Paulo	Flamengo	1	0	1.04	1.00	38	Acertou
Santos	Fortaleza	1	2	1.14	1.31	38	Acertou
Goiás	América MG	1	0	1.95	1.10	38	Acertou
Cruzeiro	Palmeiras	1	1	0.74	0.99	38	Errou
Internacional	Botafogo	3	1	0.97	1.26	38	Errou
Coritiba	Corinthians	0	2	1.35	1.64	38	Acertou
Bahia	Atlético MG	4	1	0.83	1.23	38	Errou
Cuiabá	Athletico PR	3	0	0.98	0.93	38	Acertou

Em relação aos resultados da quantidade de gols marcados por cada time, não houve nenhum acerto com o uso do modelo testado, apenas em relação às vitórias, derrotas e empates.

4 CONCLUSÕES

A proposta do trabalho foi realizar as previsões de resultados de partidas de futebol da série A do Campeonato Brasileiro de Futebol através da aplicação da distribuição de Poisson. Foi possível observar que os jogos analisados e modelados

através do R seguem, de fato, uma distribuição de Poisson, de acordo com o observado no gráfico da Figura 3.

Portanto, é notório que a distribuição de Poisson foi importante para o desenvolvimento deste trabalho; contudo, apenas ela não é suficiente para realizar as previsões de modo mais apurado. É possível concluir, a partir dos dados obtidos e de sua análise, que o time da casa se aproveita da vantagem do fator local, inevitavelmente marcando mais gols que o time visitante.

Conclui-se, portanto, que é necessário, talvez, um outro modelo com melhorias e aperfeiçoamentos, e que analise mais variáveis para que se tenha um bom desempenho e, assim, seja possível conseguir obter resultados mais precisos para uma previsão da quantidade de gols das partidas do Campeonato Brasileiro de Futebol.

REFERÊNCIAS

ALBUQUERQUE, J. P. de A. e. **Probabilidades, variáveis aleatórias e processos estocásticos**. Rio de Janeiro: Interciência, 2008.

ALVES, R.; DELGADO, C**.** Processos Estocásticos. **Faculdade de Economia**, Universidade do Porto, setembro 1997. Disponível em: https://repositorio-aberto.up.pt/bitstream/10216/71434/2/40417.pdf. Acesso em: 14 nov. 2023.

AMORA, A. S. **Minidicionário Soares da língua portuguesa.** 20ª ed. São Paulo: Saraiva, 2014.

BARBOSA, J. C. Modelagem na Educação Matemática: contribuições para o debate teórico. *In*: REUNIÃO ANUAL DA ANPED, 24, 2001, Caxambu. **Anais**. Rio Janeiro: ANPED, p. 1-14, 2001.

BIEMBENGUT, M. S.; HEIN, N. **Modelagem Matemática no Ensino.** 3ª ed. São Paulo: Contexto, 2003.

COBRE, J. **Modelos estocásticos contínuos e discretos aplicados em finanças**. 2005. 91 f. Dissertação (Mestrado em Ciências) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2005.

COSTA NETO, P. L. de O. **Probabilidades:** resumos teóricos, exercícios resolvidos, exercícios propostos. São Paulo: Edgar Blucher, 1974.

DOS REIS, H. H. B.; ESCHER, T. A. **A relação entre futebol e sociedade:** Uma análise histórico-social a partir da teoria do processo civilizador. *In*: IX Simpósio Internacional Processo Civilizador: Tecnologia e Civilização, 2005, Ponta Grossa. Tecnologia e Civilização. Ponta Grossa, Paraná, Brasil, 2005. p. 1.

GASPARETTO, T. M. **O futebol como negócio**: uma comparação financeira com outros segmentos. **Revista Brasileira de Ciências do Esporte**, 2013, v. 35, n. 4, p. 825-845. Disponível em: https://doi.org/10.1590/S0101-32892013000400003. Acesso em: 13 jul. 2023. ISSN 2179-3255.

GITHUB. Pasta Willamorim/brasileirao – no R. Disponível em: https://github.com/. Acesso em: 16 nov. 2023.

KARLIN, S.; PINSKY, M. A. **An Introduction to Stochastic Modeling**.4ª ed. Califórnia: ElsevierInc, 2011.

KIRKENDALL, D. T. Evolution of soccer as a research topic. **Progress in Cardiovascular Diseases**, 2020, v. 63, n. 6, p. 723-729.

KOVÁCS, Z. L. **Teoria da Probabilidade e Processos Estocásticos com Aplicações em Engenharia de Sistemas e Processamento de Sinais**. São Paulo: Escola Politécnica, Universidade de São Paulo, 1996.

LEE, A. J**.** Modeling scores in the premier league: is Manchester United really the best?**Chance**, Taylor & Francis Group, 1997, v. 10, n. 1, p. 15-19.

MAFALDA, C. P. *et al*.; O ensino da modelagem matemática através da estatística.*In*: XXI Jornada de pesquisa, 2016, Ijuí. **Ensaio teórico**. Ijuí: Unijuí, 2016.

MAHER, M. J. Modelling association football scores. **Statistica Neerlandica**, Wiley Online Library, v. 36, n. 3, p. 109-118, 1982.

NEOWAY. Modelos estatísticos: o que são e como usá-los para tomar decisões. **Neoway**, 2 ago. 2022. Disponível em: https://blog.neoway.com.br/modelos-estatisticos. Acesso em: 01 dez. 2023.

PAULA, D. N. T. **Análise estocástica da contratação de energia elétrica de grandes consumidores no ambiente de contratação livre considerando cenários correlacionados de preços de curto prazo, energia e demanda**. 2020. 92 f. Dissertação (Mestrado em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2020.

PIRES, D. M. **Teoria do Risco** – Aula 17. Nov. 2023. Apresentação de Power Point. Disponível em: https://atuaria.github.io/portalhalley/PDF/TeoriadoRisco/aula%2017%20-%20O%20processo%20de%20Poisson.pdf. Acesso em: 22 jan. 2024.

PORTNOI, M. **Probabilidade e Estatísticas**. 2006. Apresentação de Power Point. Disponível em: https://www.eecis.udel.edu/~portnoi/classroom/prob_estatistica/2006_1/lecture_slides/aula11.pdf. Acesso em: 14 dez. 2023.

SANTOS, J. M. A. dos. **Previsões de resultados em partidas do Campeonato Brasileiro de Futebol**. 2019. 78 f. Dissertação (Mestado em Modelagem Matemática) – Escola de Matemática Aplicada, Fundação Getulio Vargas, Rio de Janeiro, 2019.

SPIEGEL, M. R. **Estatística**. Tradução, revisão e adaptação de Carlos Augusto Crusius. 2ª ed. São Paulo: McGraw-Hill do Brasil, 1985.

SPIEGEL, M. R; SCHILLER, J. J.; SRINIVASAN, R. A. **Schaum's Outline**: Probability and Statistics. 3ª ed. Nova York: The McGraw-Hill Companies, 2009.

SOUZA, D. M. **Modelos ocultos de Markov**: uma Abordagem em Controle de Processos. 2013. Trabalho de Conclusão de Curso (Bacharelado em Estatística) – Universidade Federal de Juiz de Fora, Juiz de Fora, 2013.

STATISTICS AND DATA. Most Popular Sports in the World – (1930/2020). **Statistics and Data**, 2020. Disponível em: https://statisticsanddata.org/most-popular-sports-in-the-world/#pll_switcher. Acesso em: 21 dez. 2023.

TCHILIAN, F. **Modelo preditivo**: o que é, para que serve e como aplicá-lo? **ClearSale**, 11 jan. 2022. Disponível em: https://blogbr.clear.sale/modelo-preditivo-saiba-como-aplica-lo#:~:text=Um%20modelo%20preditivo%20%C3%A9%2C

%20de,matem%C3%A1tica%2C%20com%20probabilidade%20e%20estat%C3%ADstica.. Acesso em: 28 dez. 2023.

ZANETTA, D. M. T. **Conceitos básicos de inferência estatística**: apostila. São Paulo: USP/UNIVESP, s/d. Disponível em: http://midia.atp.usp.br/plc/plc0503/impressos/plc0503_02.pdf. Acesso em: 13 nov. 2023.

APÊNDICE – Código em R

Para aplicar o código do R, este trabalho seguiu o passo a passo apresentado por Robert Hickman no R bloggers, conforme segue abaixo:

```
okremotes::install_github("williamorim/brasileirao")
library(dplyr)
library(brasileirao)
library(purrr)

View(matches)
matches$gols_mandante <- as.numeric(sapply(strsplit(matches$score, "x"), `[`, 1))
matches$gols_visitante <- as.numeric(sapply(strsplit(matches$score, "x"), `[`, 2))
matches2023<-matches[matches$season==2023,]
View(matches2023)
times<-levels(as.factor(matches[matches$season==2023,]$home))

jogos<-
cbind.data.frame(matches[matches$season==2023,]$home,matches[matches$season==2023,]$away)
colnames(jogos)<-c("Mandante","Visitante")
jogos$rodada <- rep(1:(nrow(jogos)/10), each = 10)
View(jogos)

model <- list()
# stratify times abilities in attack and defense
model$parameters <- list(attack = seq(1, -1 + 2/length(times), by = -2/(length(times)-1)) %>%
append(-sum(.)) %>%
`names<-`(times),
defense = seq(1, -1 + 2/length(times), by = -2/(length(times)-1)) %>%
append(-sum(.)) %>%
`names<-`(times),
# no base rate of goals
intercept = 0,
# roughly accurate hfa for English professional football
hfa = 0.3)
```

```
# add in times
model$all_times <- times
# use a simple Poisson model with 8 goals max
model$model <- "poisson"
model$maxgoal <- 8
resultados<-cbind.data.frame(jogos[c(1:360),-
3],matches2023$gols_mandante[1:360],matches2023$gols_visitante[1:360],jogos[,3]
[1:360])
colnames(resultados)[3:5]<-c("Mgols","Vgols","rodada")

#supondo que os jogos do returno do brasileirao 2023 não ocorreram ainda
library(ggplot2)
p1 <- resultados %>%
# remove unplayed games
filter(!is.na(Mgols)) %>%
ggplot(., aes(x = Visitante, y = Mandante, fill = Mgols-Vgols)) +
geom_tile() +
# add the scorelines
geom_label(aes(label = paste(Mgols, Vgols, sep = "-")), fill = "white") +
# colour where green shows Mandante win and red an Visitante win
scale_fill_gradient2(low = "darkred", high = "green", midpoint = 0, guide = FALSE) +
scale_x_discrete(limits = levels(resultados$Mandante), position = "top") +
scale_y_discrete(limits = rev(levels(resultados$Visitante))) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0))

# plot
p1
#Aqui temos os resultados dos jogos das partidas até a rodada 36 do campeonato
brasileiro
resultados
melt_results <- function(results_df) {
results_df %>%
# select only relevant columns
select(Mandante, Visitante, Mgols, Vgols) %>%
gather(local, time,  -Mgols, -Vgols) %>%
# calculate goals for/against the team
mutate(gols_favor = case_when(
```

```
local == "Mandante" ~ Mgols,
local == "Visitante" ~ Vgols
)) %>%
mutate(gols_contra = case_when(
local == "Mandante" ~ Vgols,
local == "Visitante" ~ Mgols
))
}
real_data
jogos_returno <- jogos %>%
filter(rodada > 36) %>%
print()
#Iremos supor que as partidas do campeonato a partir da rodada 37 não foram jogadas
#para fazermos previsões dos resultados dessas partidas.
##################################################################
##################################################

real_data_home<-
cbind.data.frame(rep("Mandante",nrow(matches)),matches$home,matches$gols_manda
nte,matches$gols_visitante)
View(real_data_home)
colnames(real_data_home)<-c("local","time","gols_favor","gols_contra")
nrow(real_data_home)

real_data_away<-
cbind.data.frame(rep("Visitante",nrow(matches)),matches$away,matches$gols_visitante,
matches$gols_mandante)
View(real_data_away)
colnames(real_data_away)<-c("local","time","gols_favor","gols_contra")

real_data<-rbind.data.frame(real_data_home,real_data_away)

#Aqui criamos um data frame com os jogos dos campeonatos brasileiros de 2003 até
2023

p2 <- real_data %>%
ggplot(., aes(x = gols_favor, fill = local)) +
# smooth densities
```

```
geom_density(adjust = 8, alpha = 0.5) +
scale_fill_manual(values = c("red", "blue")) +
scale_x_continuous(breaks = 0:6) +
labs(title = "Gols marcados como mandante e como visitante no Brasileirão",
subtitle = "dados de 8406 partidas entre 2003 e 2023",
x = "gols marcados",
y = "densidade") +
theme_minimal()

# plot
p2
#Aqui temos um gráfico com as curvas referentes às quantidades de gols marcados
#uma referente às quantidades de gols marcados pelo mandante
#e outra referente à quantidade de gols marcados pelo visitante,
#no eixo horixontal temos as quantidades possiveis de gols que um time pode marcar em um jogo,
#no eixo vertical temos as densidades, ou seja, as probabilidades de ocorrer as quantidades
#de gols que estão no eixo horizontal. notamos que gande parte da curva dos mandantes está
#acima da curva dos visitantes. é mais provavel que o time visitante marque 0 gols do que o time mandante
# e é mais provavel que o time mandante marque um gol ou mais do que o time visitante.

real_data_means <- real_data %>%
group_by(local) %>%
summarise(mean_scored = mean(gols_favor)) %>%
print()

# A tibble: 2 × 2
#local     mean_scored
#<chr><dbl>
#  1 Mandante        1.54
#2 Visitante       1.03

# Aqui temos as médias de gols de mandantes e visitantes, respectivamente.
```

```
# Notamos que a média de gols dos mandantes é maior que a média de gols dos
visitantes.

simulated_poisson <- real_data_means %>%
split(f = .$local) %>%
lapply(., function(x) df = data.frame(dist = rpois(10000, x$mean_scored),
local = x$local)) %>%
# map it all together and label
map_df(I) %>%
mutate(data = "simulated")

# add these distributions to the plot
p2 + geom_density(data = simulated_poisson, aes(x = dist),
fill = NA, adjust = 8, alpha = 0.2) +
scale_fill_manual(values = c("red", "blue"), guide = FALSE) +
facet_wrap(~local)



calc_chi_squared <- function(game_location) {
goals_scored <- filter(real_data, local == game_location)$gols_favor

observed_goal_counts <- table(goals_scored)

mean_goals <- mean(goals_scored)

probs = dpois(sort(unique(goals_scored)), lambda = mean_goals) %>%
append(., 1-sum(.))

# the chi squared test
test <- chisq.test(x = c(observed_goal_counts,0), p = probs, simulate.p.value = TRUE)
test$data.name <- game_location

return(test)
}

# run test for both home and away goals

calc_chi_squared(game_location = "Mandante")
```

```
calc_chi_squared(game_location = "Visitante")

lapply(c("Mandante", "Visitante"), calc_chi_squared)
# Temos que os p-valores são maiores que 0.05,
# portanto não rejeitamos a hipótese de que as quantidades de gols, tanto dos
mandantes quanto dos visitantes,
# seguem uma distribuição de poisson

library(tidyr)

basic_model <- resultados %>%
melt_results() %>%
group_by(time) %>%

summarise(alpha = mean(gols_favor),
beta = mean(gols_contra)) %>%
print()

gremio_game37 <- jogos_returno %>%

filter(grepl("Grêmio", Mandante)) %>%
print()
gremio_game38 <- jogos_returno %>%

filter(grepl("Grêmio", Visitante)) %>%
print()

gremio_games<-rbind(gremio_game37,gremio_game38)

team_alphas <- basic_model$alpha %>% `names<-`(basic_model$time)

e_results <- paste(team_alphas[gremio_games$Mandante],
team_alphas[gremio_games$Visitante],
sep = "-") %>%

#  Grêmio-Vasco da Gama                Fluminense-Grêmio
#"1.63888888888889-1.08333333333333""1.36111111111111-1.63888888888889"
# Usamos o parâmetro de ataque(alpha) para estimar os resultados das partidas
```

```
# São boa estimativas, relativamente. Porém é intuitivo pensar que o Grêmio faria mais gols no Vasco,
# do que no Fluminense, Podemos mostrar isso usando o parâmetro Beta, referente a defesa, das equipes.


`names<-`(c(paste(gremio_games$Mandante, gremio_games$Visitante, sep = "-"))) %>%
print()

team_betas <- basic_model$beta %>% `names<-`(basic_model$time)

e_results <- paste(round(team_alphas[gremio_games$Mandante]*
team_betas[gremio_games$Visitante], 3),
round(team_alphas[gremio_games$Visitante]*
team_betas[gremio_games$Mandante], 3),
sep = "-") %>%
`names<-`(c(paste(gremio_games$Mandante, gremio_games$Visitante, sep = "-"))) %>%
print()
# Grêmio-Vasco da Gama    Fluminense-Grêmio
#"2.231-1.625""2.042-1.958"
#Agora levando em consideração o parâmetro referente à defesa de cada time (beta) temos essas previsões

home_advantage_gamma <- sum(resultados$Mgols) / sum(resultados$Vgols)

e_results <- paste(round(team_alphas[gremio_games$Mandante]*
team_betas[gremio_games$Visitante] *

home_advantage_gamma, 3),
round(team_alphas[gremio_games$Visitante]*
team_betas[gremio_games$Mandante], 3),
sep = "-") %>%
`names<-`(c(paste(gremio_games$Mandante, gremio_games$Visitante, sep = "-"))) %>%
print()
```

```
#Grêmio-Vasco da Gama    Fluminense-Grêmio
#"2.894-1.625""2.649-1.958"
#Agora levando em consideração o parâmetro gamma, referente ao fator local,
#notamos que há uma diferença nas quantidades de gols dos times mandantes.

gamma <- home_advantage_gamma

predict_results <- function(home, away, parameters) {
e_goals_home <- parameters$alpha[home]*parameters$beta[away] * gamma
e_goals_away <- parameters$alpha[away]*parameters$beta[home]

df <- data.frame(home = home, away = away,
e_hgoal = e_goals_home, e_agoal = e_goals_away)
return(df)
}

basic_parameters <- basic_model %>%

select(-time) %>%

as.list() %>%
lapply(., function(x){names(x) <- times;return(x)})

predicted_jogos <- map2_df(jogos_returno$Mandante, jogos_returno$Visitante,
predict_results,
# parameters forms an extra argument that does not vary
basic_parameters) %>%
# round the outputs
mutate_if(is.numeric, round, digits = 2) %>%
print()

colnames(predicted_jogos)<-c("Mandante","Visitante","e_Mgols","e_Vgols")
predicted_jogos

#Aqui temos uma tabela com os todos os jogos do campeonato das rodadas 37 e 38,
# com os seus respectivos resultados preditos, levando em consideração os três
parâmetros
#alfa, beta e gamma
```

```
predicted_resultados <- map2_df(resultados$Mandante, resultados$Visitante,
predict_results,
basic_parameters) %>%
mutate_if(is.numeric, round, digits = 2) %>%
print()

colnames(predicted_resultados)<-c("Mandante","Visitante","e_Mgols","e_Vgols")
predicted_resultados
#Aqui temos uma tabela com os todos os jogos do campeonato das rodadas 1 até 36,
# com os seus respectivos resultados preditos, levando em consideração os três
parâmetros
#alfa, beta e gamma

likelihoods <- data.frame(lik_Mgols = dpois(resultados$Mgols,
predicted_resultados$e_Mgols),
lik_Vgols = dpois(resultados$Vgols,
predicted_resultados$e_Vgols)) %>%
# round the probabilities
mutate_all(round, 4) %>%
# bind likelihoods to results
cbind(resultados, . ) %>%
# bind in predictions1
left_join(., predicted_resultados, by = c("Mandante", "Visitante")) %>%
# select useful parameters
select(Mandante, Visitante, Mgols, e_Mgols, lik_Mgols, Vgols, e_Vgols, lik_Vgols) %>%
print()

#O valor esperado da distribuição de Poisson é igual a λ,
#então podemos substituir λ como nossos gols previstos,
#por exemplo como os gols reais, e calcular a probabilidade de esses resultados
ocorrerem
#dados os parâmetros de ataque/defesa/fator local que pensamos serem correto.

log_likehood <- sum(log(likelihoods$lik_Mgols), log(likelihoods$lik_Vgols)) * -1

log_likehood
```

```
#[1] 1055.49

#Somando o log desses valores de probabilidade, teremos uma medida
#do quão incorreto estão nossas previsões.

#para termos uma ideia de que se isso ruim ou não, vamos criar um modelo,
#com todos os parâmetros definidos como zero. Dado que podemos ter certeza de que
#o Grêmio será melhor que o Vasco da Gama, então este modelo deverá
#ter um desempenho pior do que o nosso modelo básico

equal_parameters <- list(
alpha = rep(1, length(times)) %>% `names<-`(times),
beta = rep(1, length(times)) %>% `names<-`(times)
)


worse_log_likelihood <- map2_df(resultados$Mandante, resultados$Visitante,
predict_results,
equal_parameters) %>%
mutate_if(is.numeric, round, digits = 2) %>%
# take the log probability straight away this time
mutate(lik_Mgols = dpois(resultados$Mgols, e_hgoal, log = TRUE),
lik_Vgols = dpois(resultados$Vgols, e_agoal, log = TRUE)) %>%
select(lik_Mgols, lik_Vgols) %>%
map_dbl(sum) %>%
sum(.) * -1

worse_log_likelihood
#[1] 1039.942

#Vamos que a pior probabilidade é melhor do que vimos anteriormente
#Isso quer dizer que nosso modelo é bom


#Para avançar no código é necessário realizar a otimização dos parâetros vistos, até o
presente momento.
#alphae beta são os parâmetros para cada equipe e gamma é o parâmetro para o fator
local.
```

```
#parâmetros esses que tem a função de minimalizar o log de probabilidade negativa.
Deste modo,
#a previsão dos resultados de jogos disputados, terá uma precisão maior.
#Para isso, utilizamos a função optim(), que pego um vetor de parâmetros e iterou
enquanto alterou
#ligeiramente os valores até obter o valor considerado mais baixo, na função fornecida.
#ainda, foi necessário um quadro de dados de resultados de jogos entre os times,
#estes serâo previstos e comparados com os dados reais.
#Ao final a função foi configurada também para passar informações de cada iteração
para o ambiente global
#

optimise_params <- function(parameters, results) {
# form the parameters back into a list
# parameters names alpha (attack), beta (defense), and gamma (hfa)
param_list <- relist_params(parameters)

# predict the expected results for the games that have been played
e_results <- map2_df(resultados$Mandante, resultados$Visitante,
predict_results,
param_list)

# calculate the negative log likelihood of those predictions
# given the parameters how likely are those scores
neg_log_likelihood <- calculate_log_likelihood(resultados, e_results)

# capture the parameters and likelihood at each loop
# only do it if i is initialised
if(exists("i")) {
i <<- i + 1
current_parameters[[i]] <<- parameters
current_nll[[i]] <<- neg_log_likelihood
}

# return the value to be minimised
# in this case the negative log likelihood
return(neg_log_likelihood)
}
```

```
relist_params <- function(parameters) {
parameter_list <- list(
# alpha = attack rating
alpha = parameters %>%
.[grepl("alpha", names(.))] %>%
`names<-`(times),
# beta = defence rating
beta = parameters %>%
.[grepl("beta", names(.))] %>%
`names<-`(times),
# gamma = home field advantage
gamma = parameters["gamma"]
)

return(parameter_list)
}


predict_results <- function(home, away, param_list) {
# expected home goals
e_goals_home <- param_list$alpha[home] * param_list$beta[away] *
param_list$gamma
# expected away goals
e_goals_away <- (param_list$alpha[away] * param_list$beta[home])

# bind to df
df <- data.frame(home = home, away = away,
e_hgoal = e_goals_home, e_agoal = e_goals_away)

return(df)
}

calculate_log_likelihood <- function(results, e_results) {
home_likelihoods = dpois(resultados$Mgols, lambda = e_results$e_hgoal, log = TRUE)
away_likelihoods = dpois(resultados$Vgols, lambda = e_results$e_agoal, log = TRUE)

# sum log likelihood and multiply by -1 so we're minimising neg log likelihood
```

```
likelihood_sum <- sum(home_likelihoods, away_likelihoods)
neg_log_likelihood <- prod(likelihood_sum, -1)

return(neg_log_likelihood)
}

equal_parameters <- list(
alpha = rep(1, length(times)) %>% `names<-`(times),
beta = rep(1, length(times)) %>% `names<-`(times),
gamma = 1
)


optimised_parameters <- optim(
# the equal initial parameters
par = unlist(equal_parameters),
# run over the function to optimise parameters
fn = optimise_params,
# extra arguments to function
results = resultados,
# Nelder-Mead equation with 10k iterations max
method = "Nelder-Mead",
control = list(maxit = 10000)
)

optimised_parameters$par

#alpha.América MG  alpha.Athletico PR   alpha.Atlético MG        alpha.Bahia
alpha.Botafogo
#0.8860919         1.0906552          1.0875436          1.0072367          1.2422212
#alpha.Bragantino   alpha.Corinthians     alpha.Coritiba     alpha.Cruzeiro
alpha.Cuiabá
#1.0000634         0.9863920          0.9304289          0.7581381          0.7921953
#alpha.Flamengo    alpha.Fluminense    alpha.Fortaleza        alpha.Goiás
alpha.Grêmio
#1.1637870         1.0893805          0.9400355          0.7978653          1.3882715
#alpha.Internacional    alpha.Palmeiras       alpha.Santos    alpha.São Paulo
alpha.Vasco da Gama
```

```
#0.8930183          1.3982929          0.8699733          0.8249727          0.8809583
#beta.América MG    beta.Athletico PR    beta.Atlético MG          beta.Bahia
beta.Botafogo
#1.7624736          0.9245687          0.6431358          1.1103274          0.8034857
#beta.Bragantino     beta.Corinthians       beta.Coritiba       beta.Cruzeiro
beta.Cuiabá
#0.7662321          1.0637951          1.6146213          0.7293119          0.8503985
#beta.Flamengo      beta.Fluminense       beta.Fortaleza          beta.Goiás
beta.Grêmio
#0.9004421          1.0339010          1.0051078          1.1844845          1.2495793
#beta.Internacional       beta.Palmeiras          beta.Santos      beta.São Paulo  beta.Vasco
da Gama
#1.0033189          0.7547612          1.3687390          0.8656762          1.1296415
#gamma
#1.3657569

optimised_parameters$value
#[1] 990.8959

predict_results <- function(home, away, param_list) {
e_goals_home <- (param_list$alpha[home] / param_list$beta[away]) *
param_list$gamma
e_goals_away <- (param_list$alpha[away] / param_list$beta[home])

df <- data.frame(home = home, away = away,
e_hgoal = e_goals_home, e_agoal = e_goals_away)

return(df)
}



optimised_parameters2 <- optim(
par = unlist(equal_parameters),
fn = optimise_params,
results = resultados,
method = "Nelder-Mead",
control = list(maxit = 10000))
```

```
# check this does what we want
optimised_parameters2$par

#alpha.América MG  alpha.Athletico PR   alpha.Atlético MG        alpha.Bahia
alpha.Botafogo
#0.8878546          1.0770433          1.0576609          0.9806228          1.2213296
#alpha.Bragantino   alpha.Corinthians      alpha.Coritiba      alpha.Cruzeiro
alpha.Cuiabá
#.0296293          1.0064423          0.9290028          0.7355747          0.7867021
#alpha.Flamengo    alpha.Fluminense     alpha.Fortaleza        alpha.Goiás
alpha.Grêmio
#1.1462731          1.0865280          0.9387860          0.7822923          1.3144831
#alpha.Internacional     alpha.Palmeiras        alpha.Santos     alpha.São Paulo
alpha.Vasco da Gama
#0.8987185          1.3516151          0.8551177          0.8183357          0.8602899
#beta.América MG   beta.Athletico PR    beta.Atlético MG         beta.Bahia
beta.Botafogo
#0.5312031          1.0438325          1.4920949          0.8653477          1.2322760
#beta.Bragantino    beta.Corinthians       beta.Coritiba       beta.Cruzeiro
beta.Cuiabá
#1.2641292          0.9076655          0.5915431          1.2877321          1.1724656
#beta.Flamengo     beta.Fluminense      beta.Fortaleza         beta.Goiás
beta.Grêmio
#1.0484974          0.9356483          0.9890784          0.8137465          0.7447752
#beta.Internacional      beta.Palmeiras         beta.Santos      beta.São Paulo  beta.Vasco
da Gama
#0.9784766          1.3151242          0.7071595          1.1361464          0.8448213
#gamma
#1.3229529

optimised_parameters2$value
#990.4508

predict_results <- function(home, away, param_list) {
e_goals_home <- exp(param_list$alpha[home] - param_list$beta[away] +
param_list$gamma)
e_goals_away <- exp(param_list$alpha[away] - param_list$beta[home])
```

```
df <- data.frame(home = home, away = away,
e_hgoal = e_goals_home, e_agoal = e_goals_away)

return(df)
}

equal_parameters <- list(
alpha = rep(0, length(times)) %>% `names<-`(times),
beta = rep(0, length(times)) %>% `names<-`(times),
gamma = 0
)


optimised_parameters3 <- optim(
par = unlist(equal_parameters),
fn = optimise_params,
results = resultados,
# using log will avoid non-finite differences
# so can use BFGS model
method = "BFGS",
control = list(maxit = 10000))

optimised_parameters3$par

#alpha.América MG  alpha.Athletico PR   alpha.Atlético MG         alpha.Bahia
alpha.Botafogo
#-0.061700247         0.097646010         0.093197605         0.021371137
0.246607229
#alpha.Bragantino    alpha.Corinthians      alpha.Coritiba      alpha.Cruzeiro
alpha.Cuiabá
#0.082530605         0.025034084        -0.033356996        -0.277605025        -
0.205093267
#alpha.Flamengo    alpha.Fluminense     alpha.Fortaleza         alpha.Goiás
alpha.Grêmio
#0.200281552         0.118760499        -0.026030817        -0.187480804
0.320059725
#alpha.Internacional     alpha.Palmeiras         alpha.Santos     alpha.São Paulo
alpha.Vasco da Gama
```

```
#-0.068291854         0.331286065        -0.122364770        -0.158875214        -
0.106654894
#beta.América MG    beta.Athletico PR    beta.Atlético MG          beta.Bahia
beta.Botafogo
#-0.594632225         0.068231226         0.455881914        -0.139300498
0.220334281
#beta.Bragantino     beta.Corinthians       beta.Coritiba       beta.Cruzeiro
beta.Cuiabá
#0.257003665        -0.073314291        -0.495776587         0.315490044
0.144309588
#beta.Flamengo      beta.Fluminense      beta.Fortaleza          beta.Goiás
beta.Grêmio
#0.062268045        -0.035009910         0.001781604        -0.185703873        -
0.256810820
#beta.Internacional       beta.Palmeiras          beta.Santos      beta.São Paulo  beta.Vasco
da Gama
#-0.010531912         0.271412409        -0.319678957         0.170502349        -
0.145776674
#gamma
#0.260559134

optimised_parameters3$value
#990.2221

relist_params <- function(parameters) {
parameter_list <- list(
alpha = parameters %>%
.[grepl("alpha", names(.))] %>%
append(prod(sum(.), -1), .) %>%
`names<-`(times),
beta = parameters %>%
.[grepl("beta", names(.))] %>%
append(prod(sum(.), -1), .) %>%
`names<-`(times),
gamma = parameters["gamma"]
)

return(parameter_list)
```

```
}


equal_parameters <- list(
alpha = rep(0, length(times)-1) %>% `names<-`(times[2:length(times)]),
beta = rep(0, length(times)-1) %>% `names<-`(times[2:length(times)]),
gamma = 0
)


i <- 0
# collect current parameter values and neg log likelihood at each iteration
current_parameters <- list()
current_nll <- list()


optimised_parameters4 <- optim(
par = unlist(equal_parameters),
fn = optimise_params,
results = resultados,
method = "BFGS",
control = list(maxit = 10000))

optimised_parameters4$par

#alpha.Athletico PR  alpha.Atlético MG       alpha.Bahia     alpha.Botafogo
alpha.Bragantino
#0.084436556       0.079751143       0.006849459       0.234771063
0.069259622
#alpha.Corinthians     alpha.Coritiba     alpha.Cruzeiro       alpha.Cuiabá
alpha.Flamengo
#0.010553424      -0.048230391      -0.295946227      -0.222415284
0.187743338
#alpha.Fluminense     alpha.Fortaleza         alpha.Goiás       alpha.Grêmio
alpha.Internacional
#0.105809384      -0.040913082      -0.204127281       0.309256328      -
0.083944934
```

```
#alpha.Palmeiras        alpha.Santos     alpha.São Paulo alpha.Vasco da Gama
beta.Athletico PR
#0.320456379        -0.138208083        -0.175427786        -0.122790405
0.083287972
#beta.Atlético MG          beta.Bahia       beta.Botafogo     beta.Bragantino
beta.Corinthians
#0.477377111        -0.126629989         0.238003355         0.274889470        -
0.059908445
#beta.Coritiba       beta.Cruzeiro         beta.Cuiabá       beta.Flamengo
beta.Fluminense
#-0.486481621         0.334554866         0.160336202         0.077302724        -
0.021290143
#beta.Fortaleza          beta.Goiás         beta.Grêmio  beta.Internacional
beta.Palmeiras
#0.015821916        -0.173620393        -0.245589973         0.003645385
0.289376401
#beta.Santos      beta.São Paulo  beta.Vasco da Gama              gamma
#-0.308961146         0.187067735        -0.133345718         0.288294125

optimised_parameters4$value
#990.3768

p3 <- data.frame(likelihood = unlist(current_nll),
iteration = seq(length(current_nll))) %>%
ggplot(aes(x = iteration, y = likelihood)) +
geom_line(colour = "red") +
# cut out some cases where optim() has been a bit ambitious
coord_cartesian(ylim = c(0, 100000)) +
labs(title = "Negative log likelihood of parameters over iterations",
y = "negative log likelihood",
x = "iteration") +
theme_minimal()

p3

p4 <- optimised_parameters4$par %>%
# relist to add in first team
relist_params() %>%
```

```
unlist() %>%
# select team parameters
.[grepl("beta|alpha", names(.))] %>%
data.frame(value = .,
parameter = names(.)) %>%
separate(parameter, into = c("parameter", "team"), "\\.") %>%
# spread into wide format
spread(parameter, value) %>%
# pipe into a plot
ggplot(aes(x = alpha, y = beta)) +
geom_point() +
ggrepel::geom_text_repel(aes(label = team)) +
stat_smooth(method = "lm", se = FALSE) +
labs(title = "Optimal parameters for teams",
subtitle = "given first 36 weeks of results",
x = "alpha (more likely to score ->)",
y = "beta (less likely to concede ->)") +
theme_minimal()

p4

p5 <- current_parameters %>%
# get the parameters for arsenal for each iteration
lapply(., function(x){ unlist(relist_params(x))}) %>%
map_df(bind_rows, .id = "iteration") %>%
# melt data and split parameters into team and parameter
gather("parameter", "value", -iteration) %>%
# get rid of the gamma parameter
filter(parameter != "gamma.gamma") %>%
separate(parameter, into = c("parameter", "team"), sep = "\\.") %>%
# spread data back by parameter
spread(parameter, value) %>%
mutate(iteration = as.numeric(iteration)) %>%
# plot alpha against beta for each iteration
ggplot(aes(x = alpha, y = beta)) +
geom_text(aes(label = team)) +
labs(title = 'Parameters for Iteration {floor(frame_time)}',
subtitle = "given first 8 weeks of results",
```

```
x = "alpha (more likely to score ->)",
y = "beta (less likely to concede ->)") +
# using gganimate package
gganimate::transition_time(iteration) +
gganimate::ease_aes('linear') +
gganimate::view_follow()

# animate the plot
gganimate::animate(p5, nframes = i)

predicted_resultados <- predict_results(jogos_returno$Mandante,
jogos_returno$Visitante,
relist_params(optimised_parameters4$par)) %>%
mutate_if(is.numeric, round, 2) %>%
print()
colnames(predicted_resultados)<-colnames(resultados)[-5]

p6 <- rbind(
predicted_resultados %>%
rename_if(is.numeric, gsub, pattern = "e_", replacement = "") %>%
mutate(type = "predicted"),
resultados %>%
select(-rodada) %>%
mutate(type = "resultado")
) %>%
ggplot(., aes(x = Visitante, y = Mandante, fill = Mgols - Vgols)) +
geom_tile() +
# add the scorelines
geom_label(aes(label = paste(Mgols, Vgols, sep = "-"), colour = type), fill = "white") +
# colour where black for actual results and red for predictions
scale_colour_manual(values = c("red", "black")) +
# colour where green shows home win and red an away win
scale_fill_gradient2(low = "darkred", high = "green", midpoint = 0, guide = FALSE) +
scale_x_discrete(limits = levels(resultados$Mandante), position = "top") +
scale_y_discrete(limits = rev(levels(resultados$Visitante))) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0))
```

```
p6
resultados_returno<-cbind.data.frame(jogos[c(361:nrow(jogos)),-
3],matches2023$gols_mandante[361:nrow(jogos)],matches2023$gols_visitante[361:nro
w(jogos)],jogos[,3][361:nrow(jogos)])
colnames(resultados_returno)[3:5]<-c("Mgols","Vgols","rodada")
resultados_returno

resultados_returno_preditos<-cbind.data.frame(resultados_returno[,-
5],predicted_resultados[,c(3,4)],resultados_returno[,5])
colnames(resultados_returno_preditos)[7]<-"rodada"
colnames(resultados_returno_preditos)[5:6]<-c("e_Mgols","e_Vgols")

for (i in 1:nrow(resultados_returno_preditos)) {
if (resultados_returno_preditos$Mgols[i] > resultados_returno_preditos$Vgols[i]) {
# Time da casa ganhou
resultados_returno_preditos$previsão[i] <-
ifelse(resultados_returno_preditos$e_Mgols[i] > resultados_returno_preditos$e_Vgols[i],
"Acertou", "Errou")
} else if (resultados_returno_preditos$Mgols[i] < resultados_returno_preditos$Vgols[i])
{
# Time visitante ganhou
resultados_returno_preditos$previsão[i] <-
ifelse(resultados_returno_preditos$e_Mgols[i] < resultados_returno_preditos$e_Vgols[i],
"Acertou", "Errou")
} else {
# Empate
resultados_returno_preditos$previsão[i] <-
ifelse(resultados_returno_preditos$e_Mgols[i] ==
resultados_returno_preditos$e_Vgols[i], "Acertou", "Errou")
}
}

resultados_returno_preditos
sum(resultados_returno_preditos$previsão=="Acertou")
sum(resultados_returno_preditos$previsão=="Errou")
```