

Estatística no futebol

André Rodrigues Vizzoni

Orientador: Dani Gamerman



UFRJ

Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Métodos Estatísticos

2018

Estatística no futebol

André Rodrigues Vizzoni

Projeto Final de Curso submetido ao Departamento de Métodos Estatísticos do Instituto de Matemática da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção dos títulos de Bacharel em Ciências Atuariais e de Bacharel em Estatística.

Aprovado por:

Dani Gamerman

PhD - IM - UFRJ - Orientador.

Flavia Maria Pinto Ferreira Landim

Dr.Sc. - IM - UFRJ.

João Batista de Moraes Pereira

Dr.Sc. - IM - UFRJ.

Rio de Janeiro, RJ - Brasil

2018

Dedico este trabalho a meus pais, por seus diversos investimentos em meu futuro e por sua confiança em mim e no que posso alcançar.

Agradecimentos

Agradeço a meus pais por terem me incentivado durante toda minha vida, por terem acreditado em meu sucesso, por terem me apoiado em momentos difíceis e por terem me dado uma criação perfeita.

Agradeço a meu orientador, Dani Gamerman, por sua orientação e por ter sempre exigido e aceitado apenas o melhor que posso dar.

À UFRJ, por ter me proporcionado um lugar perfeito para o aprendizado.

À minha família, por todo seu apoio. Especialmente a Camila, Malu e Vinícius.

A meus amigos, por tudo que fizeram por mim. Não há como citá-los todos, mas alguns têm de ser citados, pelo quanto se doaram para me ajudar, em todos os momentos. Por isso, agradeço especialmente a Larissa, Izabel, Rafael, Gabriel e Vitor.

De forma geral, agradeço a todos que participaram desse processo, nas mais variadas maneiras.

Resumo

Quando se pensa em analisar esportes via métodos estatísticos, um dos primeiros campos a se tentar modelar é o de resultados de partidas individuais. O modelo utilizado nesse trabalho, um modelo de Poisson, tenta modelar o número de gols marcados por cada time a partir das qualidades dos ataques e das defesas dessas equipes - além da qualidade do mando de campo do mandante - e de um fator global - o intercepto - que representa as particularidades de uma liga, tal como tendências dos treinadores, qualidade do futebol jogado no país, etc. O objetivo do trabalho é comparar os resultados desse modelo aos de outros. Portanto, depois do modelo e sua adaptação dinâmica serem definidos, eles são aplicados a diferentes conjuntos de dados (duas temporadas do Campeonato Brasileiro de Futebol e uma temporada do Campeonato Inglês de Futebol), e suas verossimilhanças preditivas para cada conjunto de dados são comparadas com as verossimilhanças preditivas de outros modelos. O intuito é verificar se os modelos cumprem bem seu objetivo de prever partidas de futebol e se uma adaptação dinâmica é realmente necessária.

Palavras-Chaves: modelo de Poisson, modelos dinâmicos, comparação de modelos, resultados de futebol.

Sumário

1	Introdução	1
2	Inferência	6
2.1	Inferência frequentista	6
2.2	Inferência bayesiana	8
2.3	MCMC	10
2.4	Previsões	12
2.5	Qualidade de previsões	15
3	Modelo estático	18
3.1	Definição do modelo	18
3.2	Inferência no modelo	22
3.3	Ajuste do modelo	29
3.4	Previsões no modelo	32
3.5	Qualidade de previsões no modelo	33
3.6	Comentários	37
4	Modelo dinâmico	45
4.1	Adaptação	45
4.2	Inferência no modelo dinâmico	48
4.3	Previsões no modelo dinâmico	56
4.4	Qualidade de previsões no modelo dinâmico	58
5	Conclusão	60
A		63
A.1	Siglas utilizadas	64
A.2	Comparação entre modelos dinâmicos	70

A.3	Campeonato brasileiro de 2016	74
A.3.1	Análises e resultados do Modelo estático	74
A.3.2	Análises e resultados do Modelo dinâmico	79
A.3.3	Verossimilhanças preditivas para os modelos	98
A.4	Campeonato brasileiro de 2017	113
A.4.1	Análises e resultados do Modelo estático	113
A.4.2	Análises e resultados do Modelo dinâmico	117
A.4.3	Verossimilhanças preditivas para os modelos	137
A.5	Campeonato inglês de 2016/2017	152
A.5.1	Análises e resultados do Modelo estático	152
A.5.2	Análises e resultados do Modelo dinâmico	156
A.5.3	Verossimilhanças preditivas para os modelos	175

B		190
----------	--	------------

Lista de Figuras

2.1	Valores das cadeias de Markov.	11
3.1	Distribuição normal com intervalos de probabilidade marcados.	23
3.2	Estimativas frequentistas e bayesianas para os fatores de ataque do Campeonato brasileiro de 2016.	27
3.3	Estimativas frequentistas e bayesianas para os fatores de ataque do Campeonato brasileiro de 2016.	28
3.4	Estimativas frequentistas e bayesianas para os fatores de ataque do Campeonato brasileiro de 2016.	28
3.5	Estimativas do fator ataque para o Campeonato inglês 2016/17.	30
3.6	Estimativas dos fatores ataque a cada rodada, para o Campeonato brasileiro de 2016.	39
3.7	Estimativas dos fatores campo a cada rodada, para o Campeonato brasileiro de 2016.	39
3.8	Estimativas dos fatores defesa a cada rodada, para o Campeonato brasileiro de 2016.	40
4.1	Estimativas do fator ataque para o Campeonato brasileiro de 2017.	52
4.2	Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 4.	53
4.3	Estimativas do fator defesa para o Campeonato brasileiro de 2016.	54
4.4	Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 1.	55
A.1	Estimativas do fator ataque para o Campeonato brasileiro de 2016.	71
A.2	Estimativas do fator campo para o Campeonato brasileiro de 2016.	72
A.3	Estimativas do fator defesa para o Campeonato brasileiro de 2016.	73
A.4	Estimativas do fator ataque para o Campeonato brasileiro de 2016.	75
A.5	Estimativas do fator campo para o Campeonato brasileiro de 2016.	76
A.6	Estimativas do fator defesa para o Campeonato brasileiro de 2016.	77

A.7 Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 1.	79
A.8 Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 2.	80
A.9 Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 3.	81
A.10 Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 4.	82
A.11 Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 5.	83
A.12 Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 1.	84
A.13 Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 2.	85
A.14 Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 3.	86
A.15 Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 4.	87
A.16 Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 5.	88
A.17 Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 1.	89
A.18 Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 2.	90
A.19 Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 3.	91
A.20 Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 4.	92
A.21 Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 5.	93
A.22 Estimativas da variância das perturbações para o Campeonato brasileiro de 2016.	94
A.23 Estimativas da variância das perturbações para o Campeonato brasileiro de 2016.	95
A.24 Estimativas da variância das perturbações para o Campeonato inglês brasileiro de 2016.	96
A.25 Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2016.	99
A.26 Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2016.	100
A.27 Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2016.	101
A.28 Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2016.	102
A.29 Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2016.	103
A.30 Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2016.	104

A.31 Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2016.	105
A.32 Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2016.	106
A.33 Amostras aproximadas da posteriori do fator de ataque do Coritiba. . . .	109
A.34 Amostras aproximadas da posteriori do fator campo do Figueirense. . . .	110
A.35 Amostras aproximadas da posteriori do fator de defesa do América-MG. .	111
A.36 Amostras aproximadas da posteriori do intercepto do Campeonato brasileiro de 2016.	112
A.37 Estimativas do fator ataque para o Campeonato brasileiro de 2017. . . .	114
A.38 Estimativas do fator campo para o Campeonato brasileiro de 2017. . . .	115
A.39 Estimativas do fator defesa para o Campeonato brasileiro de 2017. . . .	116
A.40 Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 1.	119
A.41 Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 2.	120
A.42 Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 3.	121
A.43 Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 4.	122
A.44 Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 5.	123
A.45 Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 1.	124
A.46 Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 2.	125
A.47 Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 3.	126
A.48 Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 4.	127
A.49 Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 5.	128
A.50 Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 1.	129
A.51 Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 2.	130
A.52 Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 3.	131
A.53 Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 4.	132
A.54 Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 5.	133
A.55 Estimativas da variância das perturbações para o Campeonato brasileiro de 2017.	134
A.56 Estimativas da variância das perturbações para o Campeonato brasileiro de 2017.	135
A.57 Estimativas da variância das perturbações para o Campeonato brasileiro de 2017.	136

A.58 Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2017.	139
A.59 Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2017.	140
A.60 Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2017.	141
A.61 Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2017.	142
A.62 Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2017.	143
A.63 Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2017.	144
A.64 Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2017.	145
A.65 Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2017.	146
A.66 Amostras aproximadas da posteriori do fator de ataque do Bahia.	148
A.67 Amostras aproximadas da posteriori do fator campo do Fluminense.	149
A.68 Amostras aproximadas da posteriori do fator de defesa do Atlético-PR.	150
A.69 Amostras aproximadas da posteriori do intercepto do Campeonato brasileiro de 2017.	151
A.70 Estimativas do fator ataque para o Campeonato inglês 2016/17.	153
A.71 Estimativas do fator campo para o Campeonato inglês 2016/17.	154
A.72 Estimativas do fator defesa para o Campeonato inglês 2016/17.	155
A.73 Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 1.	157
A.74 Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 2.	158
A.75 Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 3.	159
A.76 Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 4.	160
A.77 Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 5.	161
A.78 Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 1.	162
A.79 Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 2.	163
A.80 Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 3.	164
A.81 Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 4.	165
A.82 Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 5.	166

A.83 Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 1.	167
A.84 Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 2.	168
A.85 Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 3.	169
A.86 Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 4.	170
A.87 Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 5.	171
A.88 Estimativas da variância das perturbações para o Campeonato inglês 2016/17.	172
A.89 Estimativas da variância das perturbações para o Campeonato inglês 2016/17.	173
A.90 Estimativas da variância das perturbações para o Campeonato inglês 2016/17.	174
A.91 Verossimilhanças preditivas padronizadas por jogo, para o Campeonato inglês 2016/17.	177
A.92 Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato inglês 2016/17.	178
A.93 Verossimilhanças preditivas padronizadas por rodada, para o Campeonato inglês 2016/17.	179
A.94 Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato inglês 2016/17.	180
A.95 Verossimilhanças preditivas padronizadas por jogo, para o Campeonato inglês 2016/17.	181
A.96 Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato inglês 2016/17.	182
A.97 Verossimilhanças preditivas padronizadas por rodada, para o Campeonato inglês 2016/17.	183
A.98 Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato inglês 2016/17.	184
A.99 Amostras aproximadas da posteriori do fator de ataque do Atlético-MG.	186
A.100 Amostras aproximadas da posteriori do fator campo do Grêmio.	187
A.101 Amostras aproximadas da posteriori do fator de defesa do Atlético-PR. .	188
A.102 Amostras aproximadas da posteriori do intercepto do Campeonato brasileiro de 2017.	189

Lista de Tabelas

3.1	Previsões para a rodada 35 do Campeonato brasileiro de 2016.	31
3.2	Probabilidades de placares específicos para AME x FLA.	34
3.3	Verossimilhanças preditivas para o Campeonato brasileiro de 2016.	36
3.4	Medidas de qualidade dos modelos.	44
4.1	Verossimilhanças preditivas para o Campeonato inglês 2016/2017.	58
4.2	Medidas de qualidade dos modelos.	58
A.1	Verossimilhanças preditivas para o Campeonato brasileiro de 2016.	107
A.2	Verossimilhanças preditivas para o Campeonato brasileiro de 2016.	107
A.3	Verossimilhanças preditivas para o Campeonato brasileiro de 2017.	147
A.4	Verossimilhanças preditivas para o Campeonato brasileiro de 2017.	147
A.5	Verossimilhanças preditivas para o Campeonato inglês 2016/2017.	185
A.6	Verossimilhanças preditivas para o Campeonato inglês 2016/2017.	185

Capítulo 1

Introdução

A definição mais comum do termo “esporte” é o de uma atividade com dois grandes objetivos: melhorar o preparo físico dos atletas participantes e prover entretenimento tanto aos atletas quanto aos espectadores. Assim, não é surpreendente que muitos tratem com desdém a ideia de encarar esportes como algo além de uma distração que não merece atenção científica.

No entanto, são vários os motivos para discordar dessa posição, e um dos primeiros e mais efetivos deles é o fato de que esportes movem quantidades imensas de dinheiro em todo o globo. Num universo em que o dinheiro é tão importante, é difícil desdenhar de atividades com essa característica.

Se for aceito que esportes merecem atenção científica, não é difícil concordar que o futebol é muito merecedor, uma vez que é o mais popular esporte do mundo. E essa afirmação sobre sua popularidade é considerada senso comum por excelentes motivos, alguns dos quais serão rapidamente enumerados a seguir.

Primeiramente, pode-se avaliar a popularidade do esporte via números de telespectadores para alguns de seus principais eventos. Os dados mais impressionantes são os seguintes: 3,2 bilhões de pessoas assistiram a pelo menos um jogo das últimas duas Copas do Mundo; cerca de 700 milhões assistiram à final de 2014 em suas TVs; juntando espectadores que assistiram pelo menos um minuto do jogo, tanto dentro quanto fora de casa, as estimativas sobem para 1 bilhão de espectadores.

Pode-se, também, utilizar dados relacionados à quantidade de dinheiro movimentada pelo futebol. Para começar, as competições futebolísticas estão entre as que mais pagam em prêmios para os vencedores. Além disso, os mais caros direitos televisivos do mundo são os da *Premier League* (campeonato inglês) e cerca de 30 dos 50 times esportivos mais caros do mundo são de futebol. Finalmente, o futebol europeu, como um todo, teve receitas em 2009/2010 de cerca de 16,3 bilhões de euros.

Por último, pode-se falar de popularidade em redes sociais, nas quais Real Madrid e Barcelona são os dois times esportivos mais seguidos do mundo, muito a frente do resto. O atleta mais seguido é Cristiano Ronaldo, também com grande vantagem. Sabendo de todos esses dados, as estimativas de que há entre 3,5 e 4 bilhões de fãs de futebol no mundo parecem razoáveis.

Visto que esse esporte é tão importante, é desapontadora a escassez de análises objetivas, rigorosas e sistemáticas por parte dos que são pagos para fazê-lo, como técnicos, dirigentes e comentaristas, entre outros. Em sua maioria, as opiniões dos analistas profissionais são muito subjetivas e raramente baseadas em fatos. Visto que esses profissionais são os principais responsáveis pela criação e solidificação de crenças que se tornam senso comum, o conhecimento popular sobre futebol também apresenta problemas de qualidade.

Por outro lado, isso significa, também, que há muito espaço de ação para a aplicação do método e do rigor científicos em análises sobre futebol. O presente trabalho é mais um ponto de uma jornada nessa direção. Trabalhos anteriores também fizeram parte dela, começando com um projeto de iniciação científica (IC), Vizzoni (2015), em que foi feita uma análise descritiva de dados futebolísticos.

Nessa análise foram usados dados referentes aos desempenhos ofensivo e defensivo de times em quatro das principais ligas de futebol do mundo (alemã, espanhola, inglesa e italiana), durante seis temporadas seguidas, começando na temporada de 2009/2010 e terminando na de 2014/2015.

Foram calculadas nesta IC as correlações lineares entre cada possível par de variáveis, não somente dentro de uma mesma temporada, mas entre uma temporada e a próxima. Um exemplo é o pareamento dos gols feitos em 2009/2010 com os gols sofridos na mesma temporada e outro seria entre gols feitos em 2009/2010 e gols feitos em 2010/2011.

O objetivo era compreender melhor o funcionamento do futebol, usando essas estatísticas e suas relações lineares como um prisma, já que uma correlação linear mede o efeito que o crescimento linear de uma quantidade tem sobre a outra, com que está sendo comparada.

Um exemplo simples não relacionado com o futebol pode ser usado para explicar o que é a correlação linear. Sabe-se que crianças e adolescentes quando ficam mais velhos crescem, não só no sentido de amadurecer, mas no sentido literal, o de aumentar de altura. Portanto, caso fosse calculada a correlação linear entre altura e idade num grupo de crianças e adolescentes, o que se veria seria uma correlação linear positiva e alta, significando que há uma relação linear forte entre as variáveis e que elas crescem, ou

diminuem, juntas.

Voltando ao conjunto de dados futebolísticos, escolheu-se calcular as correlações lineares exatamente por que elas conferem valores numéricos às relações lineares entre duas variáveis. Dessa forma, denotam não só a magnitude dessa relação, mas, também, se ela é proporcional ou inversamente proporcional. Assim, uma análise correlacional permite que se identifique os relacionamentos entre as variáveis presentes no conjunto de dados.

Por conseguinte, queria-se entender melhor o funcionamento do esporte, por meio dos cálculos das correlações lineares. Por isso, comparou-se - para cada estatística - os poderes explicativo e preditivo, sendo o primeiro definido como a correlação linear entre a estatística e o número de pontos conquistados pelo time no mesmo ano e o segundo como a correlação linear entre a estatística e o número de pontos conquistados pelo time no ano seguinte.

Buscou-se, também, avaliar a estabilidade de cada estatística, definida como a correlação linear entre a mesma estatística em anos seguidos. Finalmente, todas essas informações foram usadas para dividir as estatísticas em grupos semelhantes, para que se pudesse, por exemplo, separar estatísticas instáveis de estáveis, estatísticas com forte poder explicativo das outras e estatísticas relacionadas a uma parte do desempenho ofensivo das outras.

Os objetivos do projeto de IC foram alcançados, e algumas conclusões puderam ser tiradas a partir desse conjunto de dados: que a qualidade do ataque de um time tem uma influência maior sobre o número de pontos que ele conquista do que a de sua defesa, mas que a defesa tem maior influência sobre o número de pontos que o time conquistará no ano seguinte; que os gols que um time faz ou sofre são uma boa medida de sua qualidade não só num mesmo ano, mas no ano seguinte; e que quando se divide um gol em duas partes, uma que engloba tudo que acontece com a bola até ela ser chutada e uma que engloba o que acontece com ela depois do chute, vê-se que ambas são importantes, mas que a primeira é uma medida mais confiável da qualidade de um time.

O passo seguinte na jornada foi um novo projeto de IC, Vizzoni (2016), cujo objetivo era prever resultados de partidas de futebol. Entretanto, para atingir esse objetivo, é necessário passar por vários obstáculos inerentes à previsão estatística. Modelos preditivos precisam lidar com várias fontes de variância, e dar a essas fontes diferentes pesos. Erros acontecerão nessa definição de pesos, mas, se forem muito grandes, o funcionamento do modelo ficará deficiente demais. Por causa desses problemas, diz-se que Niels Bohr uma vez afirmou que “Fazer previsões é difícil, especialmente sobre o futuro”.

O modo mais comumente escolhido de lidar com essa dificuldade de prever o futuro é

o de avaliar as tentativas que já foram feitas de contornar os obstáculos. Não surpreendentemente, já foram desenvolvidos vários trabalhos com o mesmo objetivo, de prever resultados de partidas esportivas. Essa literatura é muito extensa, mas pode-se ressaltar alguns dos trabalhos, como os de Cattelan et al (2013), Gamerman e Souza Junior (2004), Glickman (1999), Glickman e Stern (1998), Knorr-Held (2000) e Rue e Salvensen (1997).

Em Vizzoni (2016) foram considerados modelos de comparação pareada, como o de Bradley-Terry (1952). As aplicações de modelos desse tipo a dados esportivos geralmente consideram apenas a informação de quem ganhou e quem perdeu a partida (ou se ela terminou empatada), o que levou à decisão de não se utilizar modelos desse tipo, porque se considerou que as informações de quantos gols um time fez e levou também são importantes.

A seguir, foram considerados modelos como os de Glickman e Stern, que modelam a diferença de pontuação de um jogo (o número de gols que um time sofreu subtraído do número de gols que marcou). Mas, a aproximação utilizada por Glickman e Stern, de que a diferença de placar pode ser uma variável aleatória com distribuição normal, não parece apropriada para esportes como futebol, uma vez que partidas desses esportes raramente presenciam uma quantidade suficiente de gols para que essa aproximação seja considerada razoável.

Finalmente, decidiu-se a favor de um modelo de Poisson. Porém, há diversos tipos de modelos de Poisson. No modelo escolhido, vê-se a quantidade de gols marcados por uma equipe num jogo como produto do seu ataque, da defesa de seu adversário, de seu mando de campo (caso seja o mandante) e de particularidades de sua liga. Assim, trabalha-se com a ideia de que cada plantel tem suas forças inerentes, e que elas são diferentes entre si, ou seja, que jogar em casa afeta de modos diferentes os desempenhos de Flamengo e Grêmio, por exemplo.

Em Vizzoni (2016) o que se fez, então, foi usar esse modelo para prever os resultados das partidas de algumas rodadas do campeonato brasileiro de futebol de 2016. Além disso, comparou-se as previsões desse modelo com a de alguns outros (esse processo de comparação será discutido de forma mais aprofundada na seção 2.5), e julgou-se que o desempenho do modelo escolhido para o projeto de IC tinha sido o melhor.

O presente trabalho desenvolveu o que foi feito em Vizzoni (2016). Além de se aplicar o mesmo modelo a mais rodadas do mesmo campeonato que tinha sido analisado, ele também foi aplicado a outros conjuntos de dados, de outros campeonatos. Adicionalmente, foram levadas em conta preocupações levantadas em Vizzoni (2016), sobre o fato de o

modelo não permitir que a qualidade dos times variasse ao longo do tempo.

O modelo usado em Vizzoni (2016) não permite influência temporal e dá o mesmo peso a diferentes rodadas no cálculo das qualidades ofensiva, defensiva e de campo de um time, independentemente de quando elas aconteceram. Essa é uma presunção do modelo que não parece razoável, visto que pode-se pensar em vários motivos e exemplos para o porquê de uma rodada merecer ter peso maior que outra: jogadores se contundem ou voltam de contusão, são comprados e vendidos, treinadores são demitidos e contratados, etc.

Todavia, viu-se em Vizzoni (2016) que uma outra presunção muito comum no mundo de futebol, a de que só importam os resultados recentes, também não parece ser razoável. De acordo com essa forma de encarar o esporte, partidas que ocorreram há muito tempo seriam irrelevantes para a análise da qualidade dos times, seu efeito no presente seria zero. O que se viu no trabalho é que quando se confere peso zero a partidas que foram jogadas antes de um certo ponto do campeonato estudado o desempenho do modelo piora, independentemente de que ponto se escolhe para fazer esse corte.

O presente trabalho dá continuidade ao que foi estudado em Vizzoni (2016). Usa-se o mesmo modelo que dá peso igual a todas as rodadas e não permite variação temporal dos parâmetros - a ele é dado o nome de modelo estático -. E não se usa qualquer modelo que desconsidere certas rodadas, só porque elas foram disputadas há certo tempo. Em seguida, aplicou-se o modelo a um conjunto maior de dados, que abrangem o conjunto usado em Vizzoni (2016), e se avaliou, novamente, o desempenho do modelo, comparando seus resultados com os de modelos concorrentes, criados por outros pesquisadores.

Adicionalmente, gerou-se uma segunda estrutura para modelos, baseada na primeira (a usada para gerar o modelo estático) e modificada de forma a permitir que o tempo influencie no cálculo da qualidade atual de um time, que diferentes rodadas tenham diferentes influências sobre esse cálculo.

Finalmente, a partir dessa estrutura foram gerados diversos modelos que permitissem variação temporal na qualidade de um time. Então, esses modelos foram comparados, assim como suas previsões, e foi escolhido o melhor, tanto do ponto de vista teórico quanto do ponto de vista da qualidade de suas previsões. Essas previsões foram, então comparadas com as do modelo estático e as dos modelos concorrentes.

Capítulo 2

Inferência

Infelizmente, os fatores de ataque, campo e defesa que definem o modelo estático que será utilizado são quantidades que nunca serão conhecidas. Contudo, é possível usar as observações do modelo para fazer afirmações sobre seus parâmetros. Esse é o objetivo da inferência estatística, inferir estatisticamente informações sobre quantidades desconhecidas.

São duas as principais abordagens na inferência estatística: a frequentista e a bayesiana. Há muita discussão sobre qual seria a melhor abordagem, com muitas diferenças e similaridades entre elas, algumas das diferenças sendo sutis, outras sendo bem claras.

Uma forma rápida, simples e imperfeita de diferenciá-las é: bayesianos acreditam que os parâmetros são variáveis aleatórias, enquanto os frequentistas os veem como constantes. Neste trabalho foram utilizados conceitos das duas abordagens.

2.1 Inferência frequentista

Da inferência frequentista foi utilizado o conceito de estimação de máxima verossimilhança. De forma simples, tal conceito pode ser definido como um processo de estimação dos parâmetros de um modelo. Suas estimativas são calculadas de forma a maximizar a verossimilhança de um modelo, levando em conta o que foi observado.

Dessa forma, esse processo depende da função de verossimilhança do modelo, que é uma função dos parâmetros, da definição do modelo avaliado e também dos dados observados. Por conseguinte, qualquer mudança nos dados, seja a inclusão ou exclusão de observações, leva a uma função de verossimilhança diferente.

A definição usada para a função de verossimilhança será a dada em DeGroot e Schervish

(2002), que tem a forma dada na equação (2.1),

$$l(\theta; x) = \prod_{i=1}^n f_i(x_i|\theta), \quad i = 1, \dots, n, \quad (2.1)$$

onde $l(\theta; x)$ é a função de verossimilhança (ou verossimilhança, somente) de um modelo, X é o conjunto dos dados para esse modelo, tal que

$$X = (X_1, \dots, X_n), \quad (2.2)$$

x é o valor observado de X , θ é o conjunto de parâmetros (ou parâmetro) definido pelo modelo e $f_i(x_i|\theta)$ representa a função de densidade de probabilidade (ou densidade) dos X_i , $i = 1, \dots, n$, quando se fala de variáveis contínuas, mas representa a função de probabilidade dos mesmos quando se fala de variáveis discretas. Já x_i representa o i -ésimo componente de x . Finalmente, é preciso dizer que a verossimilhança só tem a forma dada em (2.1) quando há independência condicional entre os componentes de X .

A cada mudança no conjunto de dados avaliados, a função de verossimilhança do modelo também muda. Portanto, as verossimilhanças de modelos diferentes são diferentes, devido à forma que cada modelo tem de definir a função de densidade de seus dados, mas cada modelo também possui múltiplas possíveis funções de verossimilhança, devido às múltiplas possíveis configurações do conjunto de dados a se analisar.

Modelos são representações simplificadas da realidade. Logo, quanto mais verossímil um modelo, melhor é a representação da realidade feita por ele. Para indicar o quanto um modelo se aproxima da realidade emprega-se a função de verossimilhança. E quanto maior o valor assumido por sua verossimilhança, mais verossímil ele é.

Portanto, o processo de estimação de máxima verossimilhança consiste em maximizar essa função para um conjunto de dados conhecidos, e usar os valores que os parâmetros assumem nesse ponto máximo como estimativas dos mesmos. Essas estimativas são chamadas de estimativas de máxima verossimilhança (EMVs), e sua definição está na equação (2.3),

$$\hat{\theta} = \arg\{max\{l(\theta; x), \quad \theta \in \Theta\}\}, \quad (2.3)$$

onde $\hat{\theta}$ é o conjunto de EMVs para θ e Θ é o espaço paramétrico em que θ está definido.

Um exemplo ajuda a compreender a ideia. Uma empresa farmacêutica criou um remédio que ela acredita ser ou muito fraco ou muito forte. Se muito fraco, 20% das

pessoas que usarem o remédio serão curadas de sua doença. Se muito forte, 80% serão curadas.

Para descobrir qual das opções é verdadeira, a empresa testou seu remédio num grupo de 100 pessoas, e 75 delas foram curadas. Baseado nos dados, a ideia de que o remédio é muito forte parece ser a mais razoável, a mais verossímil.

Uma EMV é feita de forma semelhante. Ela é o valor que torna o modelo mais verossímil. Dado que foi observado um conjunto de dados, a EMV é o valor que torna mais provável que se observe o que aconteceu.

2.2 Inferência bayesiana

Sob o ponto de vista da inferência bayesiana, assim chamada por causa de seu criador, o matemático do Século XVIII Thomas Bayes, os parâmetros também são quantidades desconhecidas. Mas, diferentemente da inferência frequentista, a bayesiana permite que a incerteza que se possui sobre os parâmetros seja quantificada via distribuições de probabilidade.

Ela também confere ao estatístico a possibilidade de incluir seu conhecimento teórico e subjetivo sobre o problema em sua pesquisa. Para o estatístico bayesiano, o conhecimento prévio que se tem sobre um assunto é tão importante para seu estudo quanto o conhecimento trazido pelos dados, e pode vir de várias fontes. Um estatístico pode considerar suas próprias crenças sobre um assunto, na definição de sua priori, ou conversar com um especialista, com o objetivo de encontrar a melhor priori possível entre múltiplas opções.

A equação (2.4)

$$\xi(\theta|x) \propto l(\theta;x)\xi(\theta) \quad (2.4)$$

é o Teorema de Bayes, que representa matematicamente a junção dessas características próprias da inferência bayesiana, e é, por isso, considerado o teorema mais importante dessa abordagem. A representação aqui usada foi tirada de DeGroot e Schervish (2002).

Nela, θ é o parâmetro ou conjunto de parâmetros de um modelo e $\xi(\theta)$ representa a função de densidade da distribuição de probabilidade conferida aos parâmetros, tendo em vista o conhecimento prévio que se tinha sobre o tema. Por isso, ela é chamada de distribuição *a priori* do parâmetro. A função de verossimilhança $l(\theta;x)$ é a responsável por trazer as informações dos dados x .

Finalmente, $\xi(\theta|x)$ é a função de densidade da distribuição de probabilidade conferida

aos parâmetros, tendo em vista o conhecimento prévio que se tinha sobre o tema e o conhecimento trazido pelos dados, x . Dessa maneira, ela é chamada de distribuição *a posteriori* do parâmetro.

Na inferência bayesiana, o processo de estimação de um parâmetro depende de sua posteriori. Há varios estimadores de Bayes possíveis, mas o usado neste trabalho será o que usa uma função de perda quadrática. Isso significa que as estimativas de Bayes (EBs) pontuais usadas neste trabalho serão sempre a média da posteriori do parâmetro analisado, como definido em (2.5),

$$\hat{\theta} = E[\theta|x], \quad (2.5)$$

onde $\hat{\theta}$ é a EB de θ e $E[\theta|x]$ representa a esperança, a média, da posteriori de θ . Essa esperança é definida em (2.6),

$$E[\theta|x] = \int_{-\infty}^{\infty} \theta \xi(\theta|x) d\theta, \quad (2.6)$$

onde $\xi(\theta|x)$ é a posteriori de θ .

Já as EBs intervalares usadas neste trabalho serão intervalos de credibilidade de 95%, que são definidos de forma que a probabilidade do valor do parâmetro estar dentro deles seja de 95%, usando a informação *a posteriori* que se tem sobre esse parâmetro. A definição matemática desses intervalos está em (2.7),

$$\int_{x_1}^{x_2} \xi(\theta|x) d\theta = 0,95, \quad (2.7)$$

de forma que o intervalo entre x_1 e x_2 é definido como o intervalo de 95% de credibilidade para θ , e de forma que x_1 seja o quantil de 2,5% da posteriori enquanto x_2 é o quantil de 97,5%. Ou seja, 2,5% da massa de probabilidade da posteriori se encontra antes de x_1 e 97,5% antes de x_2 , como pode-se ver em (2.8),

$$P(\theta < x_1|x) = 0,025 \quad e \quad P(\theta < x_2|x) = 0,975. \quad (2.8)$$

Neste trabalho, foram usadas EMVs e EBs, mas as primeiras só foram usadas para que se pudesse avaliar se o processo de estimação usando estimadores de Bayes estava gerando resultados aceitáveis. Como será mostrado na seção 3.2, as EMVs e EBs para o modelo estático foram muito semelhantes. Assim, decidiu-se usar as EBs, tanto pontuais quanto intervalares, de cada modelo, e as EMVs foram deixadas de lado.

Visto que não há regras sobre que prioris usar para um modelo, o pesquisador tem bastante liberdade na escolha de suas prioris. Dessa forma, muitos resolvem usar a priori menos informativa possível. Isso significa que a posteriori de um parâmetro dependerá muito mais da verossimilhança dos dados do que do conhecimento prévio que se tem sobre o parâmetro, mesmo que o número de dados seja pequeno.

O uso de prioris não informativas, ou vagas, por estatísticos bayesianos diminui em muito, pois, o peso que o conhecimento prévio sobre um assunto tem sobre as inferências feitas sobre o mesmo. Quando prioris vagas são utilizadas, os estimadores usados por um estatístico bayesiano terão resultados extremamente semelhantes aos resultados que ele teria caso tivesse usado métodos frequentistas de estimação. Como serão usadas prioris vagas para os parâmetros dos modelos definidos no presente trabalho, era esperado que as EMVs e as EBs dos modelos fossem semelhantes como foram.

Dependendo do pesquisador que trabalha com um modelo, as prioris para o modelo podem diferir muito umas entre as outras. Além disso, como visto na seção 2.1, um mesmo modelo pode ter diferentes funções de verossimilhança, uma para cada possível conjunto de dados analisados. Consequentemente, para cada conjunto haverá também uma distribuição *a posteriori* para os parâmetros do modelo. Para os modelos apresentados no presente trabalho, não é possível encontrar nenhuma de suas posteriores analiticamente e por isso foram utilizados métodos numéricos para encontrá-las.

2.3 MCMC

Para lidar com a falta de uma expressão analítica para essas posteriores, escolheu-se seguir o caminho traçado em Gamerman e Lopes (2006). Portanto, foram usados métodos de Monte Carlo via Cadeias de Markov (MCMC) para gerar amostras aproximadas para as posteriores do modelo.

Nesses métodos, sobre os quais pode-se ver mais em Gamerman e Lopes (2006), constrói-se uma cadeia de Markov que tem como distribuição limite a distribuição alvo, que nesse caso é a posteriori, desde que sejam respeitadas certas condições.

Por meio de um processo iterativo, é gerada uma amostra, e analisa-se a convergência da distribuição da cadeia para a distribuição alvo. Quando se considera que ocorreu convergência, são descartados os valores da amostra anteriores a ela. Finalmente, os valores amostrados são usados como uma amostra aproximada da posteriori.

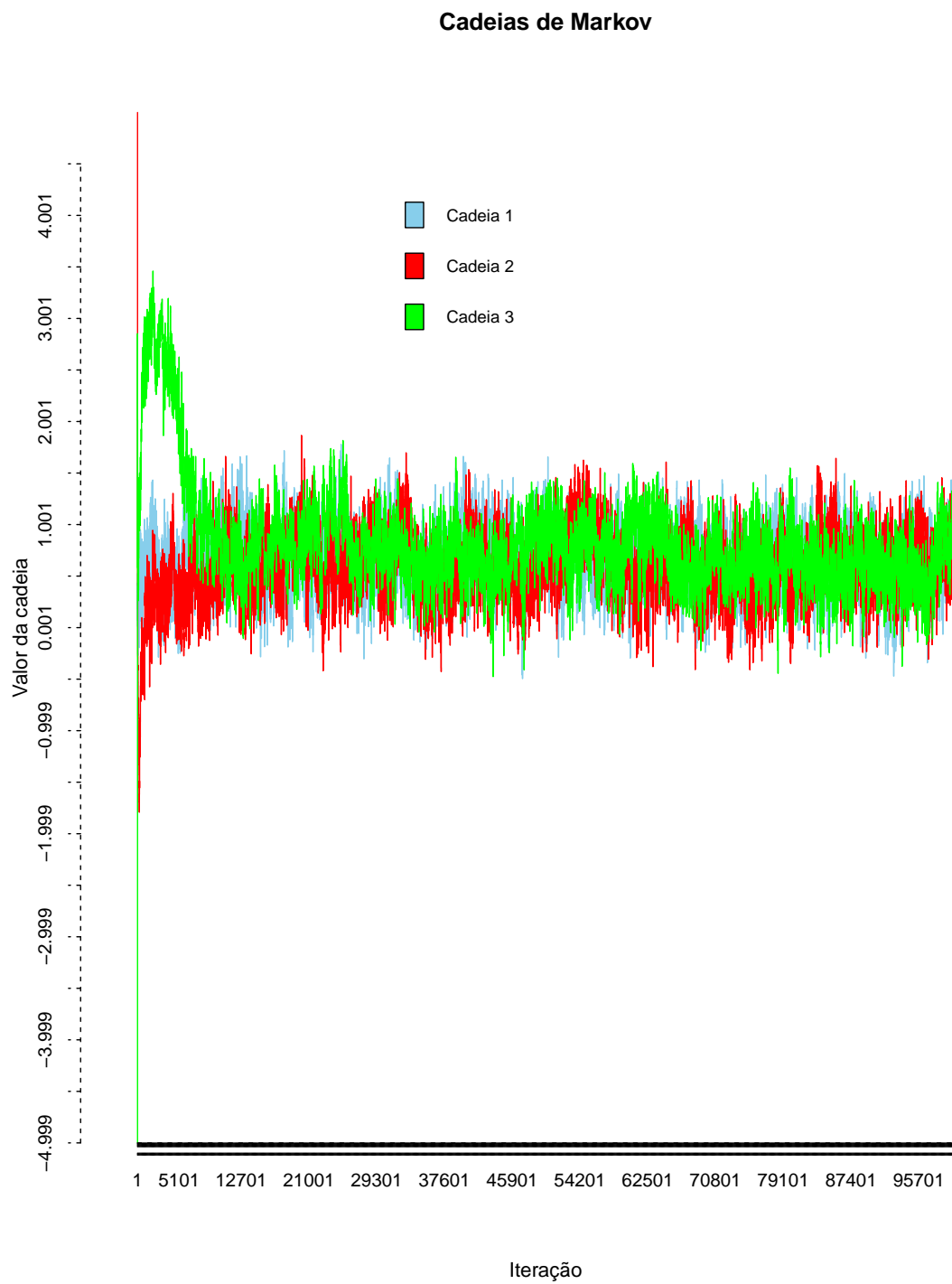


Figura 2.1: Valores das cadeias de Markov.

A Figura 2.1 pode ser usada para para ilustrar o método utilizado neste trabalho para verificar empiricamente a convergência de cadeias de Markov. Ele consistiu em usar três cadeias de Markov com a distribuição alvo como suas distribuições limite. Em seguida, checou-se se os valores delas começaram a se agrupar na mesma região da reta. Depois que se considerou que elas tinham se agrupado, os valores anteriores à convergência foram descartados. Na Figura 2.1, o eixo horizontal mostra os números das iterações do método MCMC, o que significa que não houve convergência durante as 10000 primeiras iterações, mas que a partir desse ponto houve. Pode-se ver, também, que o processo teve 100000 iterações e que se usou um espaçamento entre os valores das cadeias.

Sempre foi escolhido um espaçamento que implicasse que se teria 1000 valores para cada cadeia, após a convergência. Esses valores foram, então, colhidos e usados como uma amostra para a posteriori do parâmetro ao qual é referente. Esse método foi repetidamente usado para verificar a convergência das cadeias de Markov ao longo do trabalho, e todos os resultados apresentandos no presente trabalho são baseados em amostras compostas de valores de cadeias que convergiram.

Um exemplo de amostra da posteriori de um parâmetro θ indexada de acordo com o a -ésimo valor da amostra é dado em (2.9),

$$(\theta^{(1)}, \dots, \theta^{(n)}), \quad (2.9)$$

onde n representa o tamanho da amostra da posteriori.

As amostras das distribuições *a posteriori* obtidas dessa forma são usadas para chegar à distribuição preditiva dos jogos futuros, que é a distribuição que permite a previsão dos resultados desses jogos.

2.4 Previsões

Um dos objetivos deste trabalho é prever resultados de partidas de futebol. O instrumento que será usado para fazer previsões por meio dos modelos aqui apresentados é a distribuição preditiva dos dados. Esse é um instrumento fornecido pela inferência bayesiana e permite que sejam feitas previsões a partir do conhecimento que se tem sobre os parâmetros e os dados.

Assim, um dos objetivos do trabalho é prever observações futuras, denominadas como Y , a partir do conjunto de dados x , definido na seção 2.1. Para que se possa prever Y é

necessário defini-lo, tal como se vê em (2.10),

$$Y = (Y_1, \dots, Y_z), \quad (2.10)$$

onde y é o valor observado de Y .

Para que se possa definir matematicamente o que é a distribuição preditiva, são necessárias algumas definições de um modelo básico, antes. Primeiramente, é preciso definir a distribuição de probabilidade de y , como se vê em (2.11),

$$Y_j \sim f_j(y_j|\theta), j = 1, \dots, z. \quad (2.11)$$

Finalmente, é preciso apontar que, no modelo definido, as observações futuras, y , são independentes de x , quando ambos são condicionados ao parâmetro θ . Portanto, há independência condicional entre as observações futuras e passadas.

Dadas estas definições, é possível definir a distribuição preditiva, para a qual $f(y|x)$ representa sua função de densidade quando os dados são contínuos ou sua função de probabilidade, quando eles são discretos. Essa função é definida em (2.12),

$$f(y|x) = \int_{-\infty}^{\infty} f(y, \theta|x) d\theta, \quad (2.12)$$

que é proveniente de uma definição básica da Teoria das Probabilidades, a de que a integral de uma densidade conjunta em um de seus componentes é igual à marginal de seu outro componente. No caso, a função de densidade (ou de probabilidade) da distribuição preditiva de y dado x é igual à integral da densidade conjunta das observações futuras e do parâmetro, condicionados nas observações passadas.

Essa densidade conjunta pode ser decomposta, como mostra (2.13),

$$f(y, \theta|x) = f(y|\theta, x)\xi(\theta|x), \quad (2.13)$$

que vem de outra definição básica da Probabilidade, que determina que a função de densidade conjunta de duas variáveis pode ser decomposta como o produto entre a densidade condicional das duas variáveis - $f(y|\theta, x)$ representa a densidade de y condicionada em θ e x - e a densidade marginal da variável que não foi condicionada, θ , cuja marginal é representada por $\xi(\theta|x)$, a posteriori do parâmetro.

Em seguida, usa-se a independência condicional entre x e y para chegar à equação

(2.14),

$$f(y|\theta, x)f(\theta|x) = f(y|\theta)f(\theta|x), \quad (2.14)$$

na qual a densidade condicional mostrada em (2.13) é igualada a uma condicional, $f(y|\theta)$, que não depende de x , devido a essa independência.

A partir das equações (2.13) e (2.14), chega-se em (2.15),

$$f(y, \theta|x) = f(y|\theta)\xi(\theta|x), \quad (2.15)$$

que implica que a densidade conjunta de θ e de y , condicionada em x , é igual ao produto da densidade de y condicionado em θ e da densidade marginal de θ condicionada em x (ou seja, a posteriori de θ , avaliada em x).

Como visto em (2.15), a densidade conjunta que está sendo integrada em (2.12) pode ser decomposta, como mostra (2.16),

$$\int_{-\infty}^{\infty} f(y, \theta|x)d\theta = \int_{-\infty}^{\infty} f(y|\theta)\xi(\theta|x)d\theta. \quad (2.16)$$

Portanto, é possível finalmente chegar à definição de densidade preditiva para o modelo definido nesta seção, como mostra (2.17),

$$f(y|x) = \int_{-\infty}^{\infty} f(y|\theta)\xi(\theta|x)d\theta. \quad (2.17)$$

Para que se possa fazer previsões, é necessária uma amostra das observações futuras, Y . Como não se conhece sua densidade marginal, a preditiva, é preciso integrar a conjunta, como visto em (2.17). Porém, não é possível integrar a conjunta, visto que ela pode ser decomposta em um produto que envolve a posteriori de θ , que não é passível de ser integrada, como discutido nas seções 2.2 e 2.3.

Logo, é preciso integrar a conjunta amostralmente, fazendo uso da amostra aproximada da posteriori, ilustrada em (2.9). Essa amostra tem distribuição de probabilidade conhecida, tal como mostra

$$\theta^{(a)} \sim \xi(\theta|x), \quad (2.18)$$

que define que cada componente da amostra aproximada da posteriori de θ é uma variável aleatória cuja distribuição de probabilidade tem como sua função de densidade a posteriori do parâmetro, avaliada em x .

Esta amostra da posteriori pode ser usada então para gerar uma amostra para a preditiva. Como se vê em (2.16), a densidade conjunta de y e do parâmetro pode ser

decomposta em um produto com duas parcelas. A segunda parcela é a da posteriori, enquanto a primeira é a função de densidade de y , quando condicionado no parâmetro. Como definido em (2.11), a relação entre o parâmetro e as observações futuras é conhecida, o que implica que a primeira parcela do produto é conhecida.

A segunda, por seu lado, depende da amostra da posteriori. O processo de integração amostral é concluído, portanto, com o uso da amostra da posteriori e da definição (2.11) para a geração de uma amostra da preditiva, a partir da definição dada em (2.19),

$$Y(a)|\theta^{(a)} \sim f(y|\theta^{(a)}), \quad (2.19)$$

onde se define que cada componente da amostra da posteriori é usado como o parâmetro que define a distribuição do correspondente componente da amostra de Y .

Usando (2.19), é possível chegar à amostra da preditiva, a partir da qual são feitas previsões não só sobre resultados individuais, mas sobre uniões de diversos eventos relacionados a partidas individuais. E as probabilidades de cada evento de interesse são as somas das probabilidades de cada evento relacionado a partidas individuais.

Usando exemplos relacionados ao objeto de estudo deste trabalho, resultados de futebol, pode-se dizer que a distribuição preditiva é usada para dar respostas a perguntas como: "Qual é a probabilidade de o Flamengo marcar cinco gols no Vasco, levando em conta que o Flamengo não marcou no último jogo entre as equipes?". E ela também pode ser usada para se descobrir probabilidade de um time vencer dez jogos seguidos, ou chegar a 70 pontos, ou ser o campeão.

2.5 Qualidade de previsões

Apenas fazer previsões não é suficiente. É preciso ter como avaliar sua qualidade, para que, caso haja problemas nelas - ou caso elas simplesmente não estejam tão boas quanto o desejado -, seja possível identificá-los. Há diversas formas de efetuar essa avaliação.

Para esse projeto, escolheu-se efetuarla através do emprego de verossimilhanças preditivas. Uma verossimilhança preditiva é análoga à verossimilhança de um modelo. Definida de forma simples, a verossimilhança preditiva de um modelo é uma medida numérica de quão verossímeis foram as previsões desse modelo.

Em outras palavras, o uso de verossimilhanças preditivas indica que os modelos estão sendo comparados com base em quão perto suas previsões chegaram do que realmente aconteceu. As previsões de um modelo são comparadas com o que foi de fato observado,

e são considerados melhores os modelos que julgaram mais prováveis os resultados efetivamente ocorridos. Destarte, um modelo que tem maior verossimilhança preditiva é considerado melhor.

Para ajudar na compreensão do conceito de verossimilhança preditiva, um exemplo simples se faz necessário. Um dado foi lançado duas vezes, e nas duas vezes saiu o número seis. Dois indivíduos tinham tentado prever os resultados dos dados. Ambos acreditaram que a probabilidade de o primeiro resultado ser igual a seis era igual a 16%. No entanto, no segundo lançamento um dos dois acreditou que a probabilidade de o seis se repetir era de 5%, enquanto o outro continuou a acreditar que era de 16%. A verossimilhança preditiva para o primeiro indivíduo foi de 16% (0,16) vezes 5% (0,05), que é igual a 0,8% (0,008), enquanto a do segundo foi de 2,56% (0,0256). Olhando para as verossimilhanças preditivas, o segundo claramente fez as melhores previsões.

Tendo sido explicado o que é uma verossimilhança preditiva conceitualmente, é necessário definir matematicamente essa medida de qualidade de previsões. A definição está em (2.20),

$$f(y|x, M) = \int_{-\infty}^{\infty} \prod_{j=1}^z f_j(y_j, \theta_M|x) d\theta_M, \quad (2.20)$$

onde $f(y|x, M)$ representa a verossimilhança preditiva do modelo M para as observações futuras y, θ_M representa o parâmetro do mesmo modelo M e M representa o modelo que está sendo avaliado de acordo com a qualidade de suas previsões. Até o momento, não havia sido usado qualquer índice relacionado ao modelo avaliado, porque apenas um modelo foi definido, até esta seção. No entanto, em capítulos seguintes serão avaliados diferentes modelos, o que torna necessária uma indexação quanto ao modelo.

Como mostrado na seção 2.4, $f(y_j, \theta_M|x)$ pode ser decomposto, de forma que a verossimilhança preditiva de um modelo M possa ser mostrada em (2.21),

$$f(y|x, M) = \int_{-\infty}^{\infty} \prod_{j=1}^z f_j(y_j|\theta_M) \xi(\theta_M|x) d\theta_M. \quad (2.21)$$

Cada modelo avaliado pode ter parâmetros diferentes, de forma que o θ usado em (2.20) e (2.21) pode não ter a mesma definição para todos os modelos. Isso implica que cada modelo M terá uma posteriori $\xi(\theta|x)$ diferente, o que significa que $f(y_j|\theta)$ também será diferente para cada modelo.

Portanto, de acordo com (2.21), cada modelo M terá sua própria função de probabilidade para y, dados x e tudo que se sabe sobre os parâmetros desse modelo. Em termos

matemáticos, cada modelo M terá uma definição diferente para $f(y_j|\theta_M, M)$.

Como visto anteriormente, cada modelo também terá uma posteriori $\xi(\theta|x)$ diferente. Logo, o resultado de (2.20), $f(y|x, M)$, será diferente para cada modelo M , ainda que os dados sejam os mesmos. Isso acontece porque cada definição de modelo traz também uma definição do conjunto de parâmetros do mesmo, θ , e a função de densidade ou de probabilidade dos dados.

Isso implica que cada modelo terá verossimilhanças preditivas diferentes para o mesmo conjunto de dados, devido à forma como cada modelo foi definido. Cada modelo considera que Y depende de fatores diferentes e cada um deles define a influência que esses fatores têm diferentemente. Finalmente, em modelos probabilísticos há sempre um componente aleatório envolvido nas previsões. Levando em conta essas considerações, faz sentido que cada modelo tenha uma verossimilhança preditiva diferente.

Capítulo 3

Modelo estático

3.1 Definição do modelo

Para o modelo estático escolhido, há uma série de definições que devem ser feitas. Primeiramente, deve-se definir a forma que as observações passadas, X , terão nele, para que haja consistência na notação usada neste trabalho. Dessa forma, é preciso dizer que os números de gols feitos em partidas passadas, por cada time, serão as observações passadas. Ou seja,

$$X = \{X^1, \dots, X^T\}, \quad (3.1)$$

onde T é o número de rodadas de um campeonato. Para os conjuntos de dados estudados neste trabalho, $T=38$.

Já X^t é a coleção do número de gols feitos por todos os times que jogaram na t -ésima rodada, tal como definido em (3.2),

$$X^t = \{X_{ij}^t, \quad \forall \quad i = 1, \dots, N \quad e \quad \forall \quad j = 1, \dots, N, \quad i \neq j\}, \quad t = 1, \dots, T, \quad (3.2)$$

onde X_{ij}^t é a variável aleatória que representa o número de gols que o i -ésimo time faz no j -ésimo time, indexado de acordo com a rodada. Ela é definida em (3.3),

$$X_{ij}^t | \theta \sim \text{Poisson}(\lambda_{ij}), \quad i = 1, \dots, N, \quad j = 1, \dots, N \quad e \quad i \neq j, \quad (3.3)$$

onde θ representa o conjunto de parâmetros do modelo, e será definido mais a frente. Para os conjuntos de dados estudados neste trabalho, $N=20$. Os diferentes X_{ij}^t são independentes entre si quando condicionadas em θ e λ_{ij} é a média da distribuição dos

dados e uma função dos parâmetros, como se vê em (3.4),

$$\log(\lambda_{ij}) = At_i - De_j + Ca_i + Int, \quad i = 1, \dots, N, \quad j = 1, \dots, N \quad e \quad i \neq j. \quad (3.4)$$

Os fatores de qualidade que serão os parâmetros do modelo estão agrupados em vetores paramétricos (At , Ca e De) na equação (3.5),

$$\begin{aligned} At &= (At_1, \dots, At_N), \\ Ca &= (Ca_1, \dots, Ca_N), \\ De &= (De_1, \dots, De_N), \end{aligned} \quad (3.5)$$

onde At_i representa o fator de ataque do i -ésimo time, enquanto seus fatores campo e de defesa são representados, respectivamente, por Ca_i e De_i . Dessa forma, At representa o conjunto de fatores de ataque dos times do campeonato estudado, enquanto Ca e De representam o mesmo para os fatores campo e de defesa, respectivamente.

Todos os parâmetros do modelo podem ser representados como um vetor paramétrico, que será chamado de θ , de forma a manter consistência com a notação matemática usada no capítulo 2. Assim, θ representa o conjunto de parâmetros do modelo estático, e é definido em (3.6),

$$\theta = (At_1, \dots, At_N, Ca_1, \dots, Ca_N, De_1, \dots, De_N, Int)', \quad (3.6)$$

que, por conter todos os parâmetros, define o modelo, junto com a relação estabelecida em (3.3).

As equações (3.1) a (3.6) traduzem princípios básicos do modelo estático. Um deles é de que o número de gols que um time faz é aleatório. Escolheu-se representar essa aleatoriedade por meio de uma distribuição de probabilidade adequada e a distribuição de probabilidade considerada adequada foi uma de Poisson, como mostra (3.3). Essa distribuição modela variáveis que assumem valores inteiros (zero, um, dois, etc.) e o parâmetro que a define completamente é sua média.

Ela foi considerada apropriada porque números de gols são números inteiros (é impossível marcar 0,5 gols). Além disso, há outra forma de definir o parâmetro de uma Poisson, e que faz sentido para o modelo: a média de uma distribuição de Poisson pode ser definida, também, como uma taxa. A unidade de medida da taxa de uma Poisson é da seguinte forma: evento por unidade de exposição. Esse evento pode ser qualquer

contagem, como o número de gols que se faz. A unidade de exposição também pode representar tempo, área, pessoas, etc.

No caso de uma partida de futebol, o número de gols que um time pode fazer é restrito, pois uma partida acaba após o tempo regulamentar ter sido cumprido. Da mesma forma, um jogador só pode marcar gols enquanto estiver em campo, e nem todos os jogadores ficam a partida inteira em campo, devido às substituições que técnicos podem fazer.

Logo, o número de gols que podem ocorrer numa partida depende do tempo que a partida dura. Portanto, a partida seria a unidade de exposição, e o evento seria o número de gols feitos por cada time. Por esses motivos, considerou-se a distribuição Poisson para modelar o resultado das partidas de futebol, na maneira dada em (3.3).

Um segundo princípio é o de que o parâmetro que define a distribuição depende de fatores relacionados à qualidade dos times envolvidos na partida. Esses fatores são medidas da qualidade do ataque, do mando de campo e da defesa de cada time. Adicionalmente, há também o intercepto, um fator global, referente à liga da qual os times fazem parte, englobando fatores como qualidade dos árbitros da competição, qualidade do futebol jogado no país, tendências dos técnicos, etc.

Outro princípio básico é de que as observações utilizadas para a estimação dos parâmetros do modelo serão os resultados de partidas passadas, enquanto os objetos de previsões são os resultados de partidas futuras. Essas previsões serão feitas a partir das estimativas para os parâmetros. No entanto, para que o processo de estimação funcione corretamente, é preciso incluir uma restrição que torne o modelo identificável, pois a falta de identificabilidade gera problemas. A restrição imposta para tornar o modelo identificável exige que a soma de todos os níveis dentro de cada fator (ataque, campo e defesa) seja nula.

A equação (3.4) também define a forma como cada parâmetro influencia o número de gols feitos numa partida. Enquanto os fatores de ataque, campo e global (intercepto) são positivos, o de defesa é negativo. Portanto, quanto maior o fator de ataque ou de campo de um time, maior é o número de gols que se espera dele, e o mesmo pode ser dito de uma liga que tem um fator global alto.

Por outro lado, quanto melhor for a defesa de um time, maior é o fator negativo que influencia o número de gols feitos por seu adversário, o que significa que as chances do adversário marcar gols diminuem. Dessa forma, olhando do ponto de vista de apenas um time, quanto maior forem seus fatores campo, de ataque e de defesa, mais jogos ele ganhará.

Recapitulando, as variáveis usadas nesse modelo são assim definidas:

- θ -> Vetor paramétrico, onde estão todos os parâmetros associados às equipes da competição e o intercepto.
- X_{ij}^t -> Variável aleatória que representa o número de gols que o i-ésimo time faz no j-ésimo time, indexado de acordo com a rodada. Os diferentes X_{ij}^t são condicionalmente independentes.
- x_{ij}^t -> É o valor observado de X_{ij} .
- λ_{ij} -> Representa a média que define a distribuição de X_{ij} . Logo, pode ser interpretado como a média de gols que se espera que o i-ésimo time faça no j-ésimo time.
- At_i -> Fator de ataque do i-ésimo time.
- Ca_i -> Fator campo do i-ésimo time.
- De_i -> Fator de defesa do i-ésimo time.
- X^t -> É a coleção do número de gols feitos por todos os times que jogaram na t-ésima rodada.
- x^t -> É o valor observado de X^t .
- $X^{\{r;s\}}$ -> Coleção do número de gols feitos por todos os times que jogaram entre a r-ésima e a s-ésima rodadas.

$$X^{\{r;s\}} = \{X^t, \quad \forall \quad t \geq r \quad e \quad \forall \quad t \leq s\}, \quad r \geq 1 \quad , \quad s \leq T \quad e \quad r < s.$$

- $x^{\{r;s\}}$ -> É o valor observado de $X^{\{r;s\}}$.

A essas variáveis, uma pode ser acrescentada, a coleção dos números de gols feitos por todos os times que jogaram entre duas rodadas distintas. Ela é definida neste trabalho por (3.7),

$$X^{\{r;s\}} = \{X^t, \quad \forall \quad t \geq r \quad e \quad \forall \quad t \leq s\}, \quad r \geq 1 \quad , \quad s \leq T \quad e \quad r < s, \quad (3.7)$$

onde $X^{\{r;s\}}$ representa os números de gols feitos entre as rodadas r e s. Essa variável será muito importante para fazer e comparar previsões, porque cada pesquisador pode decidir usar diferentes valores de r e s, para seu modelo, ou para testar uma hipótese, e o valor observado da variável será representado como $x^{\{r;s\}}$.

3.2 Inferência no modelo

Neste trabalho, as distribuições *a priori*, ou prioris, escolhidas para os componentes do vetor paramétrico estão representadas nas equações (3.8) a (3.11),

$$At_i \sim N(\mu_{At}, \sigma_{At}^2), \quad i = 1, \dots, N, \quad (3.8)$$

$$Ca_j \sim N(\mu_{Ca}, \sigma_{Ca}^2), \quad j = 1, \dots, N \quad e \quad (3.9)$$

$$De_k \sim N(\mu_{De}, \sigma_{De}^2), \quad k = 1, \dots, N, \quad (3.10)$$

$$Int \sim N(\mu_{Int}, \sigma_{Int}^2), \quad (3.11)$$

onde $N(\mu, \sigma^2)$ indica que se fala de uma distribuição Normal cujos parâmetros são μ (sua média) e σ^2 (sua variância) e que tem sua função de densidade apresentada em (3.12),

$$f(x|\mu, \sigma^2) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}, \quad (3.12)$$

onde x representa o valor observado de X.

Já uma representação gráfica da distribuição Normal está na Figura 3.1. Nela, a curva $f(x)$, que é igual à função de densidade apresentada em (3.12), representa a densidade de uma Normal, enquanto foram marcados no eixo horizontal do gráfico pontos em que o desvio padrão σ (ou múltiplos desvios padrões) está sendo adicionado à média μ , ou

subtraído dela.

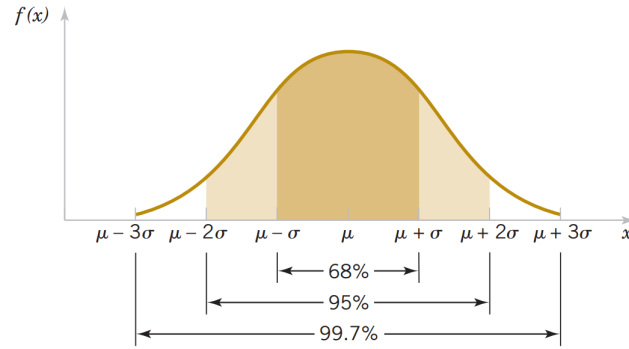


Figura 3.1: Distribuição normal com intervalos de probabilidade marcados.

As partes coloridas da Figura 3.1 representam a probabilidade da variável aleatória X ter seu valor observado, x , dentro de cada intervalo. Portanto, a probabilidade de estar a uma distância de até um desvio padrão da média é de 68%, enquanto a probabilidade de estar até dois desvios dessa média é de 95% e a de estar até 3 é de 99,7%.

A priori dos parâmetros deve ser escolhida tendo isso em mente que o número de gols que um time marca numa partida de futebol muito raramente excede cinco, e que o maior número de gols já marcado por um time é 149 (numa situação em que os perdedores repetidamente marcaram gols contra, em protesto contra a arbitragem de seu campeonato, o de Madagascar, em 2002).

Para as prioris dos parâmetros, foram escolhidas distribuições normais com médias zero e variâncias 10.000 (desvio padrão 100). Se uma variável aleatória segue uma distribuição normal, cerca de 99,7% de seus valores estarão no intervalo definido pela adição e subtração de três desvios padrões à sua média, como visto na Figura 3.1. A equação (3.13),

$$\mu_{At} = \mu_{Ca} = \mu_{De} = 0, \quad (3.13)$$

mostra o valor escolhido para a média. Em paralelo, a equação (3.14),

$$\sigma_{At}^2 = \sigma_{Ca}^2 = \sigma_{De}^2 = 10000, \quad (3.14)$$

mostra a variância escolhida.

Ou seja, está se usando o conhecimento prévio sobre os times para afirmar que os valores de seus fatores têm 99,7% de probabilidade de estar entre -300 e 300, sem que

se tenha observado qualquer resultado. Isso configura as prioris usadas como prioris vagas, ou não informativas. Portanto, toda a informação que se terá *a posteriori* sobre os parâmetros será proveniente dos dados, de forma semelhante ao que ocorre na inferência frequentista.

Também é importante apontar que escolheu-se prioris independentes entre si. Ou seja, não há expectativa prévia de dependência entre quaisquer componentes do vetor paramétrico.

Isso significa que a priori para o vetor paramétrico θ tem a forma mostrada em (3.15),

$$\theta \sim N_{3N+1}(0, 10000I_{3N+1}), \quad (3.15)$$

onde $N_{3N+1}(0, 10000I)$ indica que a priori de θ é uma normal multivariada (mais precisamente, com $3N+1$ dimensões, devido a θ ter $3N+1$ componentes), cuja média é um vetor nulo e cuja matriz de covariância é o produto entre 10000 e uma matriz identidade de dimensão $3N+1$, pois os parâmetros são independentes entre si, *a priori*.

Um vetor aleatório Z que segue uma normal multivariada com k dimensões tem a densidade presente em (3.16),

$$f(z|\mu, \Sigma) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z - \mu)'\Sigma^{-1}(z - \mu)\right\}, \quad (3.16)$$

onde z é o vetor com os valores observados para o vetor z , μ é o vetor com as médias para as k dimensões de Z e Σ é a matriz de covariância do vetor aleatório.

Tendo isso definido, é possível chegar à definição analítica da priori de θ , mostrada em (3.17),

$$\xi(\theta) = \det(2\pi 10000I_{3N+1})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\theta - 0)'(10000I_{3N+1})^{-1}(\theta - 0)\right\}. \quad (3.17)$$

Como o determinante de uma matriz identidade é igual a 1, pode-se reescrever a definição de $\xi(\theta)$, a priori de θ , conforme definido em (3.18),

$$\xi(\theta) = (20000\pi)^{-\frac{3N+1}{2}} \exp\left\{-\frac{1}{20000}\theta'\theta\right\}. \quad (3.18)$$

Destarte, as equações (3.3) a (3.6) definem o modelo usado, enquanto as equações (3.8) a (3.18) definem as prioris utilizadas. Portanto, uma das partes fundamentais da inferência baesiana, a da inclusão de conhecimento prévio sobre os parâmetros via uma priori, está completa, e a forma tomada pela priori neste trabalho está definida em

(3.18). Outra parte fundamental, a da inclusão do conhecimento adquirido via os dados não está completa, ainda, pois não foram analisados, mas a forma como eles afetarão o conhecimento que se tem sobre os parâmetros está definida, pois as equações (3.3) a (3.6) definem a forma que a função de verossimilhança do modelo terá, como será visto adiante.

É importante notar, também, que as prioris são uma tradução do conhecimento prévio que um pesquisador específico tem sobre os parâmetros. Dois pesquisadores podem usar o mesmo modelo, o que implicaria que a mesma função de verossimilhança está sendo usada, mas usar prioris diferentes. Dessa forma, as posteriores seriam diferentes, entre os modelos, e as conclusões tiradas sobre os parâmetros também o seriam.

Quanto à informação trazida pelos dados, a função de verossimilhança do modelo é responsável por traduzi-la. Para os campeonatos que serão estudados neste trabalho, a função de verossimilhança do modelo estático terá a forma dada em (3.19):

$$l(\theta; x^{\{r;s\}}) = \prod_{t=r}^s \prod_{i=1}^N \prod_{j=1, j \neq i}^N \frac{(\lambda_{ij})^{x_{ij}^t} \exp(-\lambda_{ij})}{x_{ij}^t!}, \quad (3.19)$$

onde $l(\theta; x^{\{r;s\}})$ representa a verossimilhança do modelo, seguindo o que foi definido em (2.1), com $x^{\{r;s\}}$ representando o conjunto de dados analisado. Como serão feitas previsões para diferentes rodadas, possivelmente usando diferentes conjuntos de dados, os valores de r e s variarão a cada rodada, e para cada modelo.

A função de verossimilhança usada no processo de estimação de máxima verossimilhança dos parâmetros do modelo. Ela tem essa forma devido à independência condicional entre os dados, e às equações (3.3) a (3.6), como foi dito no começo do capítulo 3.

Trabalhos com vetores paramétricos de altas dimensões, como esse, encontram problemas na estimação de seus parâmetros, porque o processo de maximizar uma função com muitos parâmetros é muito trabalhoso. Outro problema encontrado é o de falta de identificabilidade, comum a modelos como esse, que usam conjuntos de fatores como parâmetros.

Um modelo não identificável é um modelo em que o valor máximo para sua função de verossimilhança pode ser alcançado por diferentes conjuntos de argumentos, o que não leva a uma única EMV. Para tornar o modelo identificável, deu-se uma restrição aos valores dos parâmetros: a soma dos fatores de todos os times deve ser igual a zero. Ou seja, os fatores ataque de todas as equipes de uma competição devem ter soma zero e o

mesmo se aplica aos fatores defesa e campo, como pode ser visto na equação (3.20),

$$\sum_{i=1}^N At_i = 0, \quad \sum_{i=1}^N Ca_i = 0, \quad \sum_{i=1}^N De_i = 0. \quad (3.20)$$

Como mostra a seção 2.3, pode-se utilizar métodos numéricos para gerar uma amostra aproximada da posteriori de um parâmetro. Usando a definição do vetor paramétrico θ do modelo estático, dada em (3.6), e a notação usada em (2.9), vê-se que cada componente da amostra da posteriori do vetor paramétrico teria a forma dada em (3.21),

$$\theta^{(a)} = (At_1^{(a)}, \dots, At_N^{(a)}, Ca_1^{(a)}, \dots, Ca_N^{(a)}, De_1^{(a)}, \dots, De_N^{(a)}, Int^{(a)}), \quad a = 1, \dots, n, \quad (3.21)$$

E elas também são usadas para estimar os parâmetros do modelo. Como definido na seção 2.2, as EBs pontuais usadas no trabalho têm a forma dada em (2.5), enquanto as EBs intervalares têm a forma dada em (2.7). No entanto, essas fórmulas dependem da posteriori de θ , que não é possível ser descoberta analiticamente. Por isso, é necessário ter definições para essas estimativas que usem a amostra aproximada para a posteriori do parâmetro.

A EB pontual para um parâmetro passa a ser a média da amostra aproximada da posteriori desse parâmetro, conforme a equação (3.22),

$$\hat{\theta} = \frac{1}{n} \sum_{a=1}^n \theta^{(a)}, \quad (3.22)$$

onde $\hat{\theta}$ é a EB pontual para θ e cada $\theta^{(a)}$ representa um valor da amostra da posteriori de θ .

Ou seja, caso se deseje estimar o fator de defesa do Fluminense, é usada a média amostral da amostra aproximada de sua posteriori. Caso a amostra tivesse apenas dois componentes, de valores 0 e 1, a EB do fator de defesa do parâmetro seria 0,5.

Paralelamente, a EB intervalar para um parâmetro passa a ser o intervalo entre os quantis amostrais de 2,5% e de 97,5% da amostra da posteriori desse parâmetro. O quantil de uma amostra é o valor dela que divide os valores da mesma na proporção definida pelo valor do quantil, aproximadamente. O quantil 2,5% de uma amostra, portanto, é o valor que é maior que 2,5% dos valores de uma amostra, como pode-se ver em (3.23),

$$I = n * \frac{k}{100}, \quad (3.23)$$

onde é definido o índice I que determina o quantil $k\%$ de uma amostra. Dessa forma, se uma amostra for ordenada de maneira crescente, o I -ésimo valor dessa ordem é o quantil $k\%$ de uma amostra.

Como discutido na seção 2.2, as EMVs do modelo estático foram usadas como parâmetro de comparação para as EBs do modelo. As Figuras 3.2 a 3.4 contêm as EMVs pontuais e intervalares para cada parâmetro lado-a-lado com as EBs pontuais e intervalares para os mesmos parâmetros. Para cada par, a combinação de estimativas pontual e intervalar da esquerda é referente à inferência frequentista, enquanto a combinação da direita vem da inferência bayesianas.

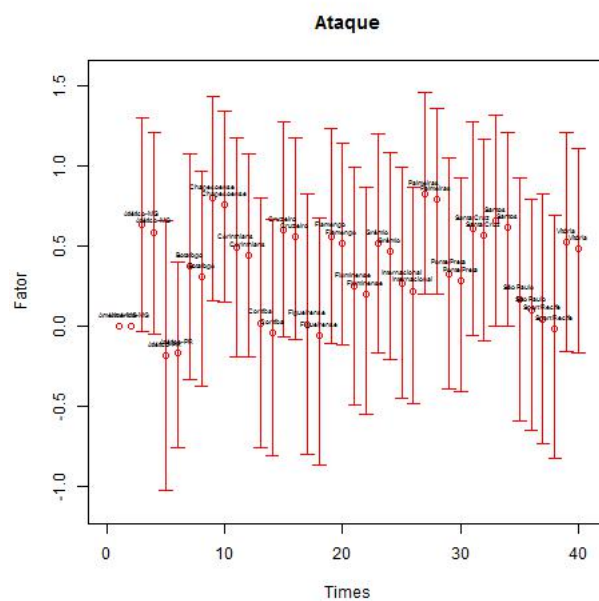


Figura 3.2: Estimativas frequentistas e bayesianas para os fatores de ataque do Campeonato brasileiro de 2016.

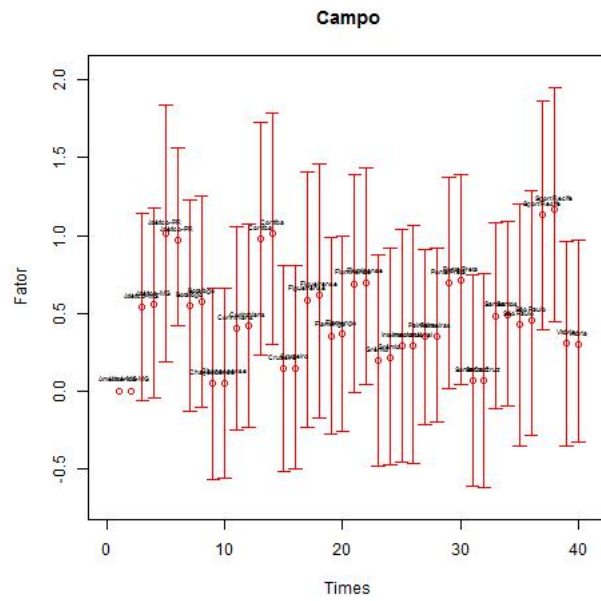


Figura 3.3: Estimativas frequentistas e bayesianas para os fatores de ataque do Campeonato brasileiro de 2016.

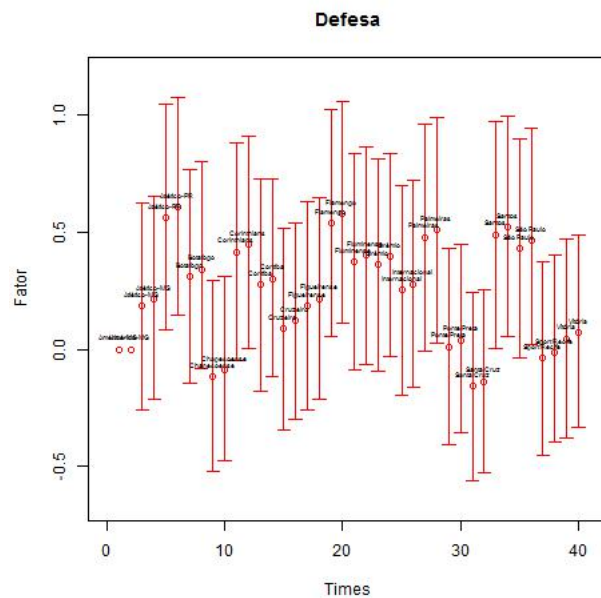


Figura 3.4: Estimativas frequentistas e bayesianas para os fatores de defesa do Campeonato brasileiro de 2016.

Conforme mostram as Figuras 3.2 a 3.4, as estimativas frequentistas e bayesianas convergiram para os mesmos resultados. O que era de se esperar, levando em conta que as priors escolhidas para os parâmetros eram não informativas.

3.3 Ajuste do modelo

No presente trabalho serão avaliados apenas conjuntos de dados relacionados a futebol, três deles. Os dois primeiros são provenientes das edições de 2016 e 2017 do Campeonato Brasileiro de Futebol (Campeonato brasileiro), e o terceiro é proveniente da temporada 2016/2017 do Campeonato Inglês de Futebol (Campeonato inglês).

Como dito anteriormente, as funções de verossimilhanças para dados desses campeonatos terão sempre a forma apresentada em (3.19), e as únicas mudanças serão devidas a mudanças no conjunto de dados analisado.

Esse conjunto mudou ao final de todas as rodadas, pois quando elas acabavam os números de gols feitos nelas eram incluídos. Ou seja, as estimativas dos parâmetros ao final de uma rodada, assim como as previsões para os jogos das próximas rodadas estavam baseadas numa verossimilhança que continha todos os resultados até aquela rodada. Na rodada seguinte, as estimativas e previsões mudariam, pois o conjunto de dados mudaria, devido à inclusão dos resultados dessa nova rodada.

Um exemplo desse contínuo ajuste do modelo está na Figura 3.5, onde estão representadas as EBs dos fatores de ataque dos times do Campeonato inglês. No gráfico, encontram-se não só essas estimativas pontuais, mas, também, estimativas intervalares. No caso, intervalos de credibilidade aproximados de 95%.

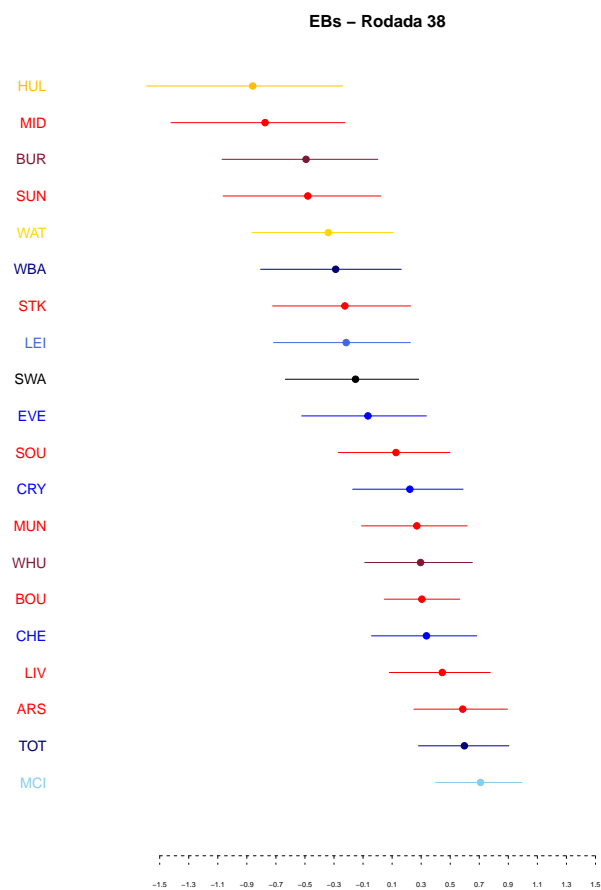


Figura 3.5: Estimativas do fator ataque para o Campeonato inglês 2016/17.

Em figuras deste trabalho que apresentam estimativas para os parâmetros do modelo, como a Figura 3.5, os times de cada competição estão representados por siglas. Essas siglas, para cada campeonato, estarão na seção A.1 do Apêndice A.

Recorreu-se a técnicas de MCMC para adquirir amostras aproximadas das distribuições *a posteriori* de cada parâmetro a cada rodada, baseado nos resultados até aquela rodada, inclusive.

A partir dessas amostras pôde-se gerar, a cada rodada, estimativas pontuais e intervalares para os parâmetros, além de uma amostra da preditiva. Por meio dessa amostra da preditiva foi possível gerar probabilidades preditivas aproximadas para os jogos depois dessa rodada.

Um exemplo de previsões feitas via o modelo usando os dados de 2016 está na Tabela 3.1, onde estão representadas as probabilidades de vitória dos mandantes (V), que são os times que aparecem à esquerda do x, de empate (E) e de vitória dos visitantes (D).

Tabela 3.1: Previsões para a rodada 35 do Campeonato brasileiro de 2016.

Jogo	V	E	D
AME x CAP	22,3	44	33,8
CAM x FIG	70,1	18,8	11
CHA x SAN	26,1	26,5	47,5
CTB x FLU	52,8	27,5	19,7
FLA x COR	46,9	30	23,1
GRE x INT	49,2	28,9	22
PAL x SPT	79,7	13,6	6,7
STA x BOT	32,2	27,1	40,7
SPO x PTE	51,7	27,7	20,6
VIT x CRU	43,8	23,9	32,3

3.4 Previsões no modelo

O modelo estático é um caso especial do modelo definido na seção 2.4. Por isso, a função de densidade ou de probabilidade da distribuição preditiva desse modelo é encontrada de forma semelhante ao que foi feito na seção 2.4.

Uma mudança é que as observações futuras não são mais representadas por y . Agora, usa-se a nomenclatura definida no modelo estático. Desta forma, as observações futuras são representadas por X_{ij}^t , com t variando entre valores de rodadas que ainda não foram observadas. Portanto, se foram observados dados até a rodada s , as observações futuras começarão a partir da rodada $s+1$.

Assim, a definição de densidade preditiva usada para o modelo estático é mostrada em (3.24),

$$f(x^{\{s+1:t\}}|x^{\{r;s\}}) = \int_{-\infty}^{\infty} f(x^{\{s+1:t\}}|\theta)\xi(\theta|x^{\{r;s\}})d\theta. \quad (3.24)$$

Essa densidade preditiva é encontrada a partir da posteriori para o conjunto de dados estudados, como pode-se ver em (3.24). Logo, também não é possível encontrá-la analiticamente. Para gerar uma amostra da preditiva, então, é necessário usar a amostra para a posteriori, ilustrada em (3.21), que foi gerada com base em (3.25),

$$\theta^{(a)} \sim f(\theta|x^{\{r;s\}}), \quad (3.25)$$

que define que cada componente da amostra aproximada da posteriori de θ é uma variável aleatória cuja distribuição de probabilidade tem como sua função de densidade a posteriori do parâmetro, avaliada com base nos dados entre as rodadas r e s .

A amostra da posteriori é usada na equação (3.26),

$$X_{ij}^{t(a)}|\theta^{(a)} \sim Poisson(\lambda_{ij}^{(a)}), \quad t \geq s+1, \quad (3.26)$$

que provém da equação (3.3), em que é definida a relação probabilística entre o vetor paramétrico e os números de gols feitos, e onde $\lambda_{ij}^{(a)}$ representa o a -ésimo componente da amostra de λ_{ij} , definido em (3.27),

$$\log(\lambda_{ij}^{(a)}) = At_i^{(a)} - De_j^{(a)} + Ca_i^{(a)} + Int^{(a)}, \quad i = 1, \dots, N, \quad j = 1, \dots, N \quad e \quad i \neq j, \quad (3.27)$$

onde $At_i^{(a)}$, $Ca_i^{(a)}$, $De_j^{(a)}$ e $Int^{(a)}$ representam os a -ésimos componentes das amostras dos fatores de qualidade, como mostrado em (3.21).

Em (3.26), define-se que cada componente da amostra da posteriori é usado como o vetor paramétrico que define a distribuição de dados observados a partir da rodada $s+1$.

Usando (3.26), é possível chegar à amostra da preditiva, pois cada componente da amostra de θ gera seus próprios $X_{ij}^{t(a)}$, e os n $X_{ij}^{t(a)}$ podem ser coletados. Dessa forma, a amostra da preditiva dos dados a partir da rodada $s+1$ pode ser gerada através da amostra da posteriori do vetor paramétrico.

Para o presente trabalho, todas as previsões avaliadas serão previsões de um-passo-a-frente. Ou seja, as previsões coletadas para todos os modelos serão previsões feitas por cada modelo após cada rodada, para a rodada seguinte. Dessa forma, as preditivas que serão avaliadas serão da forma $f(x^{s+1}|x^{\{r;s\}})$, com cada modelo definindo seu próprio r .

3.5 Qualidade de previsões no modelo

Adaptando a definição de verossimilhança preditiva dada em (2.20) ao objeto de estudo presente trabalho, tem-se a equação (3.28),

$$f(x^{s+1}|x^{\{r;s\}}, M) = \int_{-\infty}^{\infty} \prod_{m=1}^n f_m(x_m^{s+1}|\theta_M) \xi(\theta^s|x^{\{r;s\}}) d\theta, \quad (3.28)$$

onde M representa o modelo que está sendo avaliado de acordo com a qualidade de suas previsões, n representa o número de jogos jogados na rodada $s+1$, $f(x^{s+1}|x^{\{r;s\}}, M)$ representa a verossimilhança preditiva do modelo M para a rodada $s+1$ e a definição de $f(x_m^{s+1}|\theta)$ é dada em (3.29),

$$f_m(x_m^{s+1}|\theta_M) = \int_{-\infty}^{\infty} \prod_{m=1}^n f_m(x_m^{s+1}|\theta_M) \xi(\theta_M|x^{\{r;s\}}) d\theta, \quad (3.29)$$

onde x_m^{s+1} representa os gols feitos na m -ésima partida da rodada $s+1$. Casos times que tenham jogado nessa partida tenham sido o a -ésimo e o b -ésimo times, a definição de x_m^{s+1} é dada em (3.30),

$$x_m^{s+1} = (x_{ab}^{s+1}, x_{ba}^{s+1}). \quad (3.30)$$

Além disso, cada modelo avaliado neste trabalho tem parâmetros diferentes, de forma que o θ usado em (3.28) e (3.29) pode não ter a mesma definição do modelo estático, dada em (3.5). Isso implica que cada modelo M terá uma posteriori $\xi(\theta_M|x^{\{r;s\}})$ diferente.

Adicionalmente, os criadores de cada modelo definiram o valor de r , ou seja, a rodada a partir da qual serão incluídos resultados por rodada, na posteriori do conjunto de parâ-

metros de cada modelo. Eles também terão a relação entre os dados e seus parâmetros definida de forma diferente. As definições do modelo estático estão em (3.3) e (3.4), mas um modelo M qualquer pode ter determinado essa relação com equações diferentes, e provavelmente determinou. Isso significa que cada modelo também terá um $f(x_m^{s+1}|\theta)$ diferente.

Portanto, de acordo com (3.29), cada modelo M terá sua própria função de probabilidade para os jogos futuros, dados os resultados e tudo que se sabe sobre os parâmetros desse modelo. Em termos matemáticos, cada modelo M terá uma definição diferente para $f(x_m^{s+1,m}|\theta_M, x^{\{r;s\}})$.

Como visto anteriormente, cada modelo também terá uma posteriori $\xi(\theta_M|x^{\{r;s\}})$ diferente. Logo, o resultado de (3.28), $f(x^{s+1}|x^{\{r;s\}}, M)$, será diferente para cada modelo M , ainda que os dados sejam os mesmos. Isso acontece porque cada definição de modelo traz também uma definição do conjunto de parâmetros do mesmo, θ , e a função de densidade dos dados, $f(x^r|\theta_M)$ diferentemente.

A partir do presente modelo, é possível gerar a densidade conjunta preditiva para θ e para as variáveis aleatórias que se quer prever. Logo, é possível calcular exatamente a verossimilhança preditiva conjunta para esse modelo. Isso significa, também, que se pode fazer previsões específicas sobre o placar de um jogo, como pode ser visto na Tabela 3.2, na qual estão contidas as probabilidades preditivas aproximadas de ocorrerem certos placares no jogo entre América-MG e Flamengo, em Belo Horizonte.

Tabela 3.2: Probabilidades de placares específicos para AME x FLA.

Placar	P	Placar	P	Placar	P
0 x 0	14.4	1 x 0	6.8	0 x 1	19.6
1 x 1	9.5	2 x 1	2.5	1 x 2	7.6
2 x 2	2.0	2 x 0	1.9	0 x 2	15.3

Todavia, as probabilidades de cada conjunto de resultados de uma rodada ser observado são muito baixas, o que significa que o evento observado é muito raro. Essa raridade leva a grandes erros de Monte Carlo, devido à grande variância decorrente das baixas probabilidades. Portanto, a precisão que se tem em previsões feitas a partir das amostras para os resultados conjuntos é muito pequena.

Além disso, os modelos feitos por outros pesquisadores que foram escolhidos para comparação disponibilizam apenas as densidades preditivas marginais para cada variável

aleatória. Não há, portanto, informações sobre os parâmetros dos mesmos, o que impede o cálculo exato de sua verossimilhança preditiva conjunta.

Por esses motivos, empregou-se uma aproximação. Em vez de se usar a definição de verossimilhança preditiva conjunta dada em (3.28), usou-se a seguinte aproximação,

$$f(x^{s+1}|x^{\{r;s\}}, M) \approx \prod_{m=1}^n f(x_m^{s+1}|x^{\{r;s\}}, M), \quad (3.31)$$

onde presume-se independência entre os jogos. Desta forma, nela a verossimilhança preditiva conjunta de um modelo está sendo aproximada pelo produto das densidades preditivas marginais de cada modelo.

Isto é, como não se sabe a forma como cada pesquisador definiu os parâmetros de seu modelo, nem como foi definida a influência que eles têm sobre os resultados, não se pode usar a equação (3.28) para os modelos que não estão definidos neste trabalho.

Contudo, apesar de não se saber quais são seus parâmetros e como eles afetam os resultados, sabe-se o que foi previsto por cada modelo. Todos os criadores desses modelos disponibilizaram suas previsões para cada partida: probabilidades preditivas de vitória do mandante, empate e vitória do visitante.

Portanto, para cada jogo foram coletadas as previsões de cada modelo, e a verossimilhança preditiva de cada jogo foi definida como a probabilidade dada para o resultado que ocorreu. Em outras palavras, se o modelo A previu que o Flamengo tinha 40% de chance de vencer o Vasco, e isso realmente ocorreu, a verossimilhança preditiva do modelo A para o jogo entre Flamengo e Vasco foi de 40%.

Se a rodada em que Flamengo vs. Vasco foi jogado também foi a rodada em que Fluminense vs. Botafogo foi jogado, a verossimilhança preditiva do modelo A para essa rodada é o produto entre as verossimilhanças preditivas para cada jogo. Caso Fluminense e Botafogo tenham empatado, e a probabilidade dada pelo modelo de isso ocorrer tenha sido de 50%, então, a verossimilhança preditiva aproximada de A para essa rodada foi de 40% (0,4) vezes 50% (0,5), que é igual a 20% (0,2).

Essa aproximação está representada matematicamente em (3.31), onde M varia de acordo com o modelo que está sendo considerado, e tanto r quanto s podem ser diferentes, para cada modelo.

Como dito anteriormente, o modelo considerado melhor é, em geral, o que apresenta a maior verossimilhança preditiva. Por conseguinte, comparou-se as verossimilhanças preditivas conjuntas de cada modelo para suas coleções de previsões um passo a frente para

cada conjunto de dados. Ou seja, ao final de cada rodada foram coletadas as previsões de cada modelo para a rodada seguinte. Em seguida, foi calculada a verossimilhança preditiva de cada modelo para cada rodada. Finalmente, foram calculadas as verossimilhanças preditivas para todo o conjunto de rodadas escolhido.

Na Tabela 3.3 há um exemplo disso. Nela, estão as verossimilhanças preditivas de quatro modelos, além do estático, para o Campeonato brasileiro de 2016, a partir da sétima rodada, a primeira rodada em que houve convergência para as cadeias de Markov. Assim, cada um dos cinco modelos tem nove valores associados: uma verossimilhança preditiva para cada uma das oito rodadas e uma para o conjunto das oito.

Os modelos que estão sendo comparados com o modelo estático do presente trabalho (identificado como UFRJ) são: o modelo do site Chance de Gol (CdG) - apresentado em Arruda -, o modelo do Departamento de Matemática da Universidade Federal de Minas Gerais (UFMG) - apresentado em Departamento de Matemática da UFMG - e dois modelos criados para este trabalho. O modelo nulo é assim chamado porque todas suas previsões são iguais: as probabilidades preditivas são iguais a um terço para todas as opções de resultado, em todas as partidas.

Já o modelo simples tem probabilidades preditivas iguais às médias do campeonato até o momento em que elas foram feitas. Ou seja, caso os mandantes tenham ganho cerca de 53% dos jogos no campeonato até a rodada 30, as probabilidades preditivas do modelo simples para a vitória de um mandante na rodada 31 serão todas iguais a 53%. As probabilidades de vitória do visitante e de empate são calculadas de forma análoga.

Na Tabela 3.3 os valores estão padronizados em relação ao do modelo nulo, bem como nas outras tabelas mostradas no Apêndice A e que apresentarão verossimilhanças preditivas para conjuntos de dados. Ou seja, as verossimilhanças preditivas de cada modelo foram divididas pela verossimilhança preditiva do modelo nulo, para que a interpretação dos valores ficasse mais fácil.

Tabela 3.3: Verossimilhanças preditivas para o Campeonato brasileiro de 2016.

Rodadas	UFRJ	CdG	UFMG	Nulo	Simples
7 a 10	0,02	14,66	0	1	23,17
11 a 20	0,29	$2, 13.10^4$	0	1	$6, 11.10^3$
21 a 30	$1, 14.10^5$	$1, 59.10^5$	0	1	$2, 25.10^4$
31 a 38	13609,47	277,43	11,45	1	137,4
Total	$7, 84.10^6$	$1, 38.10^{13}$	11,45	1	$4, 37.10^{11}$

Dessa forma, espera-se que um modelo tenha uma verossimilhança preditiva padronizada maior que um, pois isso significa que seu desempenho foi melhor que o do modelo nulo. Isso é esperado porque o modelo nulo é o mais simples que existe, e sua definição se baseia na ideia de que a probabilidade de um time vencer outro não depende das qualidades inerentes dos times.

De modo semelhante, seria desejável que um modelo tivesse desempenho melhor que o do modelo simples, pois esse modelo também não define o resultado de uma partida como uma variável que depende da qualidade dos times envolvidos, mas como o uma variável que depende apenas da definição de quem é o mandante e quem é o visitante, na partida.

No Apêndice A estão as Figuras e Tabelas referentes a cada conjunto. Mas a Tabela 3.3 já funciona como um bom resumo. Em todos os conjuntos, o modelo estático teve desempenho muito ruim no começo de cada campeonato e muito bom no final que o do modelo nulo. No total, ele foi muito melhor que o modelo nulo, mas pior que outros modelos. Porém, há considerações a serem feitas.

Elas são sobre o tempo de que o modelo precisa para que suas previsões estabilizem. Ou seja, a quantidade de rodadas para que a variabilidade das previsões do modelo pareça ter diminuído significativamente. Parece razoável dizer que as previsões para o Campeonato brasileiro de 2016 estabilizaram a partir da vigésima terceira rodada, enquanto as do campeonato seguinte (2017) parecem ter ficado mais estáveis a partir da vigésima quarta rodada. Finalmente, os dados apontam para a vigésima oitava rodada como um limite razoável para as previsões do Campeonato inglês.

No Apêndice A, é discutido o desempenho do modelo estático nos três campeonatos estudados, nas subseções A.3.3, A.4.3 e A.5.3. Lá se mostrará a diferença de qualidade entre as previsões feitas antes e depois dos pontos de estabilização escolhidos para cada campeonato, além de se discutir um possível motivo para essa diferença qualitativa

3.6 Comentários

O primeiro comentário a se fazer é uma crítica ao modelo a partir do qual foram feitas as previsões, que é estático, apesar das muitas razões para se acreditar que o tempo influencia a qualidade de um time. Em times de futebol, por exemplo, alguns jogadores se cansam e contudem, outros são comprados e vendidos, e ainda há outros cuja habilidade muda ao longo do tempo.

Considerando isto, pode-se pensar que a melhor forma de avaliar a qualidade de um

time é uma que leva em conta a passagem do tempo. Mas, além desses questionamentos teóricos e intuitivos, pode-se, também, defender que a influência do tempo deve ser levada em conta, com base nos dados.

As Figuras 3.6 a 3.8, por exemplo, apresentam evidências a favor de considerar os parâmetros do modelo como dinâmicos. Nelas, estão as evoluções das estimativas de máxima verossimilhança dos parâmetros, ao longo das rodadas.

Ou seja, para que as Figuras fossem geradas, a cada rodada do Campeonato brasileiro de 2016 foram coletadas as EMVs para os parâmetros do modelo. Em seguida, essas estimativas foram dispostas de acordo com a rodada em que foram feitas, nas Figuras.

Logo, na Figura 3.6 estão as EMVs para os fatores de ataque de cada um dos vinte times que disputaram o campeonato. Cada linha do gráfico representa o quanto a EMV do fator desse time mudou ao longo do campeonato. O mesmo vale para as Figuras 3.7 e 3.8, que são análogas à 3.6, com a Figura 3.7 contendo as estimativas dos fatores campo e a Figura 3.8 contendo as dos fatores de defesa.

Como as estimativas feitas após uma rodada dependem de todas as rodadas até o momento da estimação, as estimativas exibem uma dependência temporal forte. Ainda assim, é possível ver o quanto as estimativas variam com o tempo, pois todas as linhas se movimentam bastante no eixo vertical.

Visto que as três figuras mostram algo semelhante, a movimentação constante das EMVs ao longo das rodadas, isso foi visto como uma forte evidência de que os fatores de qualidade de um time variam ao longo do tempo.

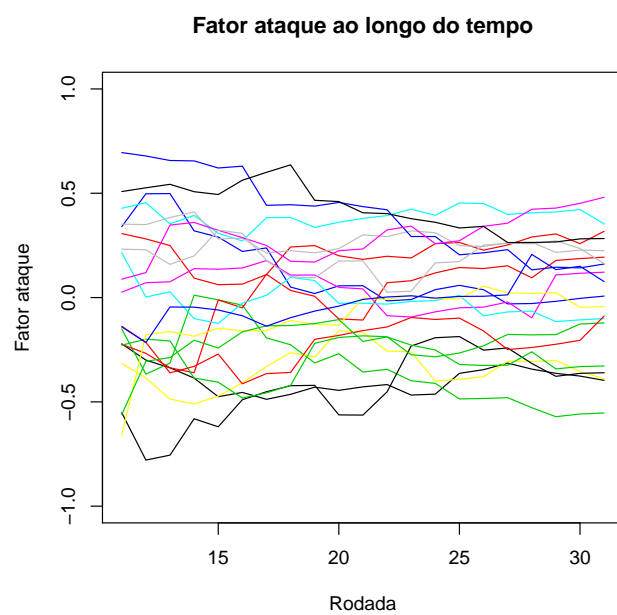


Figura 3.6: Estimativas dos fatores ataque a cada rodada, para o Campeonato brasileiro de 2016.

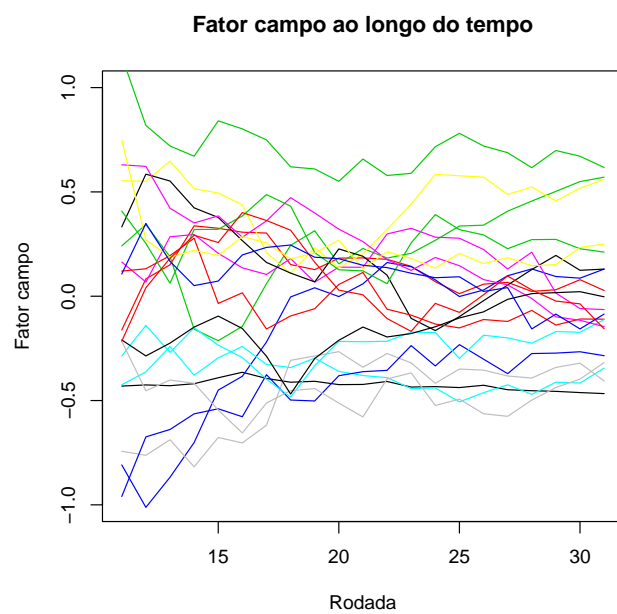


Figura 3.7: Estimativas dos fatores campo a cada rodada, para o Campeonato brasileiro de 2016.

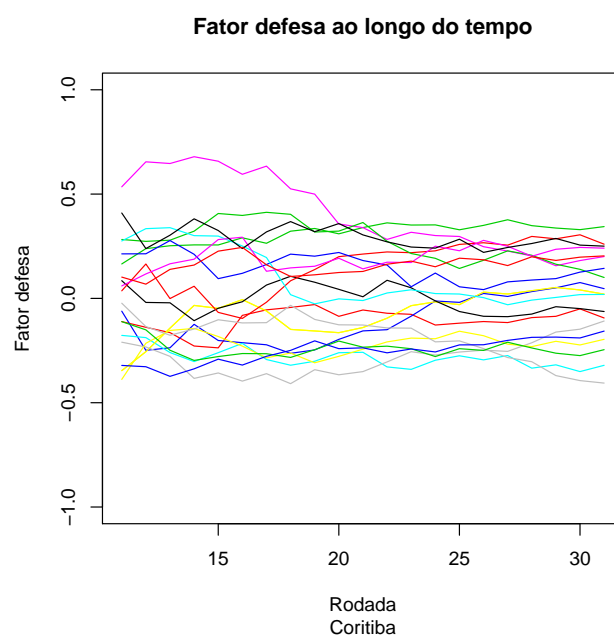


Figura 3.8: Estimativas dos fatores defesa a cada rodada, para o Campeonato brasileiro de 2016.

Um exemplo simples do que é representado nos gráficos pode ser dado. A linha referente ao Flamengo, por exemplo, contém as estimativas da qualidade do time após cada rodada. Nesse exemplo, após 25 jogos o Flamengo tinha um ataque considerado forte, com seu fator de ataque sendo 0,5. No entanto, ele não marcou gols nos dez jogos seguintes. Após 35 rodadas, a estimativa de seu fator de ataque caiu para 0,2, pois a estimativa leva em conta não só seus dez jogos sem gols, mas os primeiros 25 jogos, em que o time marcou muitos gols.

Portanto, a estimativa do fator de ataque do time após 35 rodadas não parece ser muito verossímil. Ela subestima o forte ataque que a equipe apresentou nas 25 primeiras rodadas, e superestima seu ataque nas dez seguintes. No entanto, se as estimativas de cada rodada forem coletadas, a queda de 0,5 para 0,2 será gradual, e poderá ser observada na linha referente ao Flamengo, no gráfico dos fatores de ataque.

Se todas as linhas de todos os gráficos apresentam esse comportamento, em que o valor da estimativa aumenta ou diminui gradualmente, pode-se acreditar que a qualidade de um time varia, com o tempo. E é exatamente isso que pode ser visto nas Figuras 3.6 a 3.8.

Os próximos comentários são sobre a qualidade das previsões. No caso do Campeonato brasileiro de 2016, a qualidade parece ter sido alta, especialmente a das previsões para as oito últimas rodadas, como se pôde ver na Tabela 3. No campeonato do ano seguinte a qualidade foi muito menor, com a verossimilhança preditiva padronizada do modelo estático para as últimas 16 rodadas ficando muito próxima de um, como se pode ver na Tabela A.3 do Apêndice A.

No Apêndice A também se encontra a tabela com os valores das verossimilhanças preditivas padronizadas dos modelos avaliados para as rodadas do Campeonato inglês, a Tabela A.4. Nas comparações usando dados do Campeonato brasileiro de 2017 há a adição de um novo modelo, o do site *FiveThirtyEight*, criado pelo economista americano Nate Silver, e associado à *Entertainment and Sports Programming Network* (ESPN) americana. A definição desse modelo foi apresentada em Boyce, e o modelo é representado por 538, nas Tabelas que o incluem.

Nesse site são compartilhadas previsões feitas pelo modelo usado lá. Diferentemente do modelo apresentado no presente trabalho, o modelo do *FiveThirtyEight* não usa apenas o número de gols feitos por um time como observação para seu modelo. Os criadores do modelo usam diversas estatísticas relacionadas ao desempenho dos times em suas partidas, tanto ofensivas quanto defensivas. Eles unem todas essas informações para estimar os

parâmetros de seu modelo, que são então usados para gerar previsões para as rodadas de um campeonato.

No caso do Campeonato inglês, os criadores do modelo da UFMG não se propuseram a fazer previsões sobre o campeonato inglês de futebol. Portanto, não há verossimilhanças preditivas desse modelo para a competição, e não há como esse modelo ser comparado com seus competidores, para esse conjunto de dados.

Continuando os comentários sobre a qualidade das previsões, tanto o modelo simples quanto os modelos do Chance de Gol e do *FiveThirtyEight* tiveram desempenho melhor que o do modelo estático, em suas previsões. Por outro lado, mesmo esses modelos não tiveram bom desempenho. Preocupantemente, o modelo simples, que considera que apenas o mando de campo de uma partida é suficiente para explicar o resultado da partida, teve o melhor desempenho de todos os modelos, que levam em conta mais fatores.

Isso parece apontar para uma imprevisibilidade imensa no Campeonato brasileiro de 2017, que afetou todos os modelos, aparentemente. E análises subjetivas de comentaristas brasileiros de futebol também definiram o torneio como mais imprevisível que o habitual.

Finalmente, para o Campeonato inglês as previsões do modelo estático voltaram a ter excelente desempenho, apesar de não terem sido tão boas quanto às do *FiveThirtyEight*. No entanto, ambos os modelos tiveram qualidade suficientemente semelhante. E o desempenho do modelo estático foi ligeiramente melhor que o do Chance de Gol, mas semelhante, em grande parte. Finalmente, os três modelos se saíram muito melhor que o modelo simples.

Juntando os comentários sobre as qualidades das previsões dos modelos para cada conjunto de dados, é possível tirar algumas conclusões sobre os modelos. A primeira é que o modelo estático teve desempenho melhor que o do nulo em todos os dados, muito melhor em dois deles. Isso aponta fortemente para a validade do modelo para prever resultados de futebol.

A segunda é que o desempenho dele também foi muito melhor que o do modelo simples nos mesmos dois conjuntos de dados, apesar de ter sido muito pior no terceiro. Uma análise conjunta dos três campeonatos mostra que o modelo estático teve desempenho muito superior ao do simples.

Falando sobre os modelos da UFMG e do Chance de Gol, ambos parecem ter sido vencidos pelo modelo estático, no quesito de verossimilhança preditiva. No caso da UFMG, seu modelo teve desempenho muito abaixo que o do estático, em ambos os campeonatos brasileiros.

Já o modelo do Chance de Gol teve desempenho muito pior que o do estático no Campeonato brasileiro de 2016, e muito melhor na edição seguinte, além de desempenho ligeiramente pior, no Campeonato inglês. Visto que o criador do modelo também é estatístico e que os comentários que ele faz sobre a definição de seu modelo sugerem que ele tem definição semelhante ao do estático, parece razoável que os modelos tenham tido desempenhos semelhantes.

O modelo do *FiveThirtyEight*, por outro lado, teve desempenho muito melhor que o do modelo estático, tanto no Campeonato brasileiro de 2017 quanto no Campeonato inglês. Levando em conta que o modelo do *FiveThirtyEight* parece ser muito mais complexo que o estático, é razoável que tenha havido uma diferença grande entre eles.

Na Tabela 4 estão as verossimilhanças preditivas, padronizadas e não padronizadas, de todos os modelos avaliados para o conjunto completo de dados, que contém os resultados das três competições avaliadas. Nela estão também o número de jogos para os quais os modelos geraram previsões, para que se tenha em mente as diferenças entre os tamanhos de amostra. Finalmente, os valores das verossimilhanças preditivas foram padronizados em relação ao número de jogos avaliados sob cada modelo.

Essa padronização foi feita mediante a seguinte fórmula,

$$VPP = VP^{\frac{1}{n}}, \quad (3.32)$$

onde VP representa o valor da verossimilhança preditiva de um modelo, para o conjunto de todos os dados, n representa o número de jogos nesse conjunto e VPP representa o valor da verossimilhança preditiva padronizada desse modelo. Tanto VP quanto n variam para cada modelo. Portanto, quanto maior for o valor dessa estatística, melhor foi o desempenho do modelo.

Para interpretar a Tabela 3.4 é necessária uma legenda, então. Na primeira coluna (identificada como VP) estão as verossimilhanças preditivas totais para cada modelo. Na segunda (VP^N) estão esses valores padronizados em relação ao modelo nulo (logo, valores maiores que um indicam que o modelo em questão teve desempenho melhor que o do nulo). Na terceira (Jogos) estão os números de jogos avaliados para cada modelo. Por fim, na quarta (VPP) estão as verossimilhanças padronizadas de acordo com (3.32).

Tabela 3.4: Medidas de qualidade dos modelos.

Modelo	VP	VP^N	Jogos	VPP
UFRJ	$6, 27.10^{-181}$	$5, 14.10^{19}$	419	0,3715
538	$1, 14.10^{-112}$	$1, 28.10^{12}$	260	0,3706
UFMG	$7, 06.10^{-111}$	0,13	229	0,3304
CdG	$3, 37.10^{-181}$	$2, 76.10^{19}$	419	0,3709
Nulo	$1, 22.10^{-200}$	1	419	0,3333
Simples	$1, 59.10^{-189}$	$1, 3.10^{11}$	419	0,3543

É importante apontar que só foram incluídos nos conjuntos de dados jogos que ocorreram depois dos pontos de estabilização das previsões do modelo estático.

A Tabela 3.4 reforça as conclusões feitas acima. O modelo estático teve desempenho melhor que o nulo, o que o aponta como um modelo válido para previsões de resultados de futebol. Também teve desempenho muito melhor que o do modelo simples, o que aponta que se deve considerar mais que o mando de campo, em previsões. Seu desempenho foi melhor que o do Chance de Gol e o da UFMG, de forma geral. Por fim, um dado inesperado: seu desempenho foi melhor que o do *FiveThirtyEight*, quando se leva em conta os números de jogos que ambos previram (o modelo do *FiveThirtyEight* não foi usado por seus criadores para fazer previsões sobre o Campeonato brasileiro de 2016).

Considerando-se o exposto, pode-se afirmar que o modelo teve resultados bons, apesar de não conferir qualquer variabilidade temporal a seus parâmetros, e de precisar, em geral, de mais da metade de um campeonato de 38 rodadas para que suas previsões estabilizem.

Capítulo 4

Modelo dinâmico

4.1 Adaptação

Como visto anteriormente, o modelo analisado até o momento, o modelo estático, é um modelo que não permite que os fatores de qualidade dos times variem ao longo do tempo. Na estimação desses fatores, o modelo estático também dá o mesmo peso a todas as rodadas.

Ele é assim definido apesar das muitas razões para acreditar que o tempo influencia a qualidade de um time, e que rodadas mais recentes deveriam ter um peso maior na avaliação da qualidade atual de um time.

Por isso, julgou-se necessária a utilização de um modelo dinâmico. Felizmente, o modelo estático utilizado possui uma adaptação dinâmica, a qual já foi usada em múltiplos trabalhos, em especial o de Gamerman e Souza Junior (2004).

Nesse modelo dinâmico, as equações básicas definidas no capítulo 3 continuam a ser usadas, só que algumas alterações foram feitas. No modelo dinâmico, continua-se a julgar que a distribuição de probabilidade apropriada para o número de gols feitos por um time é a Poisson. Ou seja,

$$X_{ij}^t | \theta \sim \text{Poisson}(\lambda_{ij}^t), \quad i = 1, \dots, N, \quad j = 1, \dots, N, \quad i \neq j \quad e \quad t = 1, \dots, T, \quad (4.1)$$

onde θ representa o conjunto de parâmetros do modelo e será definido a seguir. É possível ver que a única mudança de (3.3) para (4.1) é que a taxa que define a Poisson agora é indexada de acordo com a rodada t do campeonato estudado.

A taxa definida em (4.1) difere da definida em (3.4) por apresentar o índice t , pois os parâmetros do modelo dinâmico são todos indexados com base na rodada de que se fala,

como mostra (4.2),

$$\log(\lambda_{ij}^t) = At_i^t + De_j^t + Ca_i^t + Int^t, \quad i = 1, \dots, N, \quad j = 1, \dots, N, \quad i \neq j \quad e \quad t = 1, \dots, T, \quad (4.2)$$

onde At_i^t representa o fator de ataque do i -ésimo time na rodada t , Ca_i^t representa o fator campo do i -ésimo time na mesma rodada, De_j^t representa o fator de defesa do j -ésimo time e Int^t representa o intercepto para a t -ésima rodada do campeonato estudado.

Enquanto no modelo estático havia apenas um vetor paramétrico θ , definido em (3.5), no modelo dinâmico há um vetor paramétrico para cada rodada, tal como em (4.3),

$$\theta^t = (At_1^t, \dots, At_N^t, Ca_1^t, \dots, Ca_N^t, De_1^t, \dots, De_N^t, Int^t), \quad t = 1, \dots, T, \quad (4.3)$$

de forma que cada vetor paramétrico contém todos os parâmetros para cada rodada.

Todos os vetores paramétricos de um campeonato podem ser colocados num mesmo conjunto, conforme mostra (4.4),

$$\theta = (\theta^1, \dots, \theta^T), \quad (4.4)$$

que define θ como o conjunto de todos os parâmetros do modelo para uma competição.

Lembra-se que, para o modelo estático, vale o que (4.5) apresenta,

$$\theta^1 = \theta^2 = \dots = \theta^T = \theta, \quad (4.5)$$

o que implica que a dimensão do vetor paramétrico para cada campeonato é muito menor para o modelo estático do que para o dinâmico.

A adaptação dinâmica citada anteriormente consiste em permitir a variação dos valores dos parâmetros ao longo do tempo, via as seguintes equações, apresentadas em Knorr-Held (2000),

$$At_i^{t+t_2} = At_i^t + \omega_{i,At}^{t+t_2}, \quad i = 1, \dots, N, \quad (4.6)$$

$$Ca_i^{t+t_2} = Ca_i^t + \omega_{i,Ca}^{t+t_2}, \quad i = 1, \dots, N, \quad (4.7)$$

$$De_i^{t+t_2} = De_i^t + \omega_{i,De}^{t+t_2}, \quad i = 1, \dots, N, \quad e \quad (4.8)$$

$$Int^{t+t_2} = Int^t + \omega_{Int}^{t+t_2}. \quad (4.9)$$

Nas equações (4.6) a (4.9) há a inclusão de termos, os ω , que representam perturbações periódicas ao valor dos parâmetros. Ou seja, os parâmetros de um time em uma rodada dependem de seus parâmetros nas rodadas anteriores e de uma perturbação, com cada tipo de fator tendo sua própria perturbação. Dessa forma, o ataque do i -ésimo time sofre uma perturbação, $\omega_{i,At}^{t+t_2}$, em cada ponto de atualização de seu valor e o mesmo vale para os fatores campo (afetado por $\omega_{i,Ca}^{t+t_2}$) e defesa ($\omega_{i,De}^{t+t_2}$) do time, assim como para o intercepto do torneio ($\omega_{Int}^{t+t_2}$).

Pode-se usar um exemplo simples para facilitar a compreensão desse conceito. De acordo com o que está definido nas equações (4.6) a (4.9), o ataque do Flamengo na rodada 31 depende não só da qualidade que o time demonstrou durante todo o campeonato, até a rodada anterior. Depende, também, de quaisquer mudanças no time que foram causadas pela passagem do tempo.

A volta de lesão de um atacante importante, por exemplo, deve aumentar a qualidade de seu ataque. E esse aumento seria traduzido matematicamente como a adição de um termo com valor positivo ao fator de ataque da equipe. Portanto, na equação (4.6) o ω_{At} relacionado ao Flamengo teria um valor maior que zero.

Seguindo com o exemplo, há situações em que a qualidade de um time pode cair com o tempo, também. Caso o goleiro do Flamengo tenha se contundido, durante a semana que precede a partida do time, faz sentido presumir que a qualidade da defesa do time diminuirá. Isso seria representado matematicamente com a adição de um termo negativo ao fator de defesa do time. Assim, na equação (4.8) o ω_{De} relacionado ao Flamengo teria um valor menor que zero.

Para finalizar o exemplo, alguns valores serão usados. Ao final da trigésima rodada o ataque do Flamengo era considerado mediano, e o fator de ataque do time tinha sido estimado como zero. Já sua defesa era considerada forte, e seu fator de defesa fora estimado como igual a 0,5. Após a volta do atacante lesionado, seu ataque ficou mais forte, tendo sido adicionado ao seu fator ataque um valor de 0,25. Por outro lado, a contusão de seu goleiro levou ao enfraquecimento de sua defesa, com a adição de -0,25 a seu fator de defesa.

Portanto, os fatores de ataque e defesa desse fictício plantel do Flamengo seriam ambos iguais a 0,25. Isso provavelmente significa que a probabilidade de o time vencer sua próxima partida não mudou muito. No modelo estático, as estimativas dos parâmetros não mudariam muito após a partida pois ela teria o mesmo peso que as anteriores. No modelo dinâmico, essas grandes mudanças seriam representadas na variação dos valores

dos parâmetros, entre as rodadas.

4.2 Inferência no modelo dinâmico

No contexto da periodicidade na atualização dos valores dos parâmetros, é necessário definir t_2 , usado nas equações (4.6) a (4.9). Esse termo representa o número de rodadas entre os pontos de atualização dos valores dos parâmetros. Portanto, se $t_2=1$, os valores do parâmetro atualizam a cada rodada.

Contudo, por preocupações computacionais, não foi possível usar um modelo dinâmico em que os parâmetros fossem atualizados a cada rodada. Para este trabalho, foram testados modelos dinâmicos com múltiplos valores para t_2 , e os valores testados foram

$$(5, 8, 10, 13, 19).$$

Cada um dos modelos dinâmicos foi construído a partir da estrutura apresentada entre as equações (4.1) e (4.9), e foi definido com um dos possíveis valores para t_2 . Eles foram comparados entre si, então, e o modelo que foi considerado o melhor foi um em que o valor usado para t_2 era cinco (mais explicações sobre os modelos comparados estão na seção A.2 do Apêndice A). Portanto, o modelo dinâmico utilizado e discutido neste trabalho atualiza os valores de seus parâmetros a cada cinco rodadas. Isso implica que as definições básicas do modelo devem ser atualizadas, o que significa que as equações (4.1) e (4.9) serão atualizadas.

A primeira delas, (4.1), passa a ser definida como em (4.10),

$$X_{ij}^t | \theta \sim \text{Poisson}(\lambda_{ij}^{\lceil \frac{t}{5} \rceil}), \quad i = 1, \dots, N, \quad j = 1, \dots, N, \quad i \neq j \quad e \quad t = 1, \dots, T, \quad (4.10)$$

onde $\lceil \frac{t}{5} \rceil$ indica que o valor que indexa a taxa da distribuição Poisson é o menor valor inteiro maior ou igual a $\frac{t}{5}$, como pode-se ver em (4.11),

$$X[k] = \min\{n \in \mathbb{Z} | n \geq k\}. \quad (4.11)$$

De forma análoga, a equação (4.2) passa ser definida como em (4.12),

$$\log(\lambda_{ij}^{\lceil \frac{t}{5} \rceil}) = A t_i^{\lceil \frac{t}{5} \rceil} + D e_j^{\lceil \frac{t}{5} \rceil} + C a_i^{\lceil \frac{t}{5} \rceil} + \text{Int}^{\lceil \frac{t}{5} \rceil}, \quad i = 1, \dots, N, \quad j = 1, \dots, N, \quad i \neq j \quad e \quad t = 1, \dots, T, \quad (4.12)$$

a equação (4.3) passa a ter a forma dada em (4.13),

$$\theta^{\lceil \frac{t}{5} \rceil} = (At_1^{\lceil \frac{t}{5} \rceil}, \dots, At_N^{\lceil \frac{t}{5} \rceil}, Ca_1^{\lceil \frac{t}{5} \rceil}, \dots, Ca_N^{\lceil \frac{t}{5} \rceil}, De_1^{\lceil \frac{t}{5} \rceil}, \dots, De_N^{\lceil \frac{t}{5} \rceil}, Int^{\lceil \frac{t}{5} \rceil}), \quad t = 1, \dots, T, \quad (4.13)$$

e a equação (4.4) tem agora a definição apresentada em (4.14),

$$\theta = (\theta^1, \dots, \theta^{\lceil \frac{T}{5} \rceil}), \quad (4.14)$$

onde $\lceil \frac{T}{5} \rceil$ é igual a oito, para os campeonatos estudados neste trabalho, que têm 38 rodadas.

Adicionalmente, as equações (4.6) a (4.9) passam a seguir as formas dadas de (4.15) a (4.18),

$$At_i^{\lceil \frac{t}{5} \rceil + 1} = At_i^{\lceil \frac{t}{5} \rceil} + \omega_{i, At}^{\lceil \frac{t}{5} \rceil + 1}, \quad i = 1, \dots, N, \quad (4.15)$$

$$Ca_i^{\lceil \frac{t}{5} \rceil + 1} = Ca_i^{\lceil \frac{t}{5} \rceil} + \omega_{i, Ca}^{\lceil \frac{t}{5} \rceil + 1}, \quad i = 1, \dots, N, \quad (4.16)$$

$$De_i^{\lceil \frac{t}{5} \rceil + 1} = De_i^{\lceil \frac{t}{5} \rceil} + \omega_{i, De}^{\lceil \frac{t}{5} \rceil + 1}, \quad i = 1, \dots, N, \quad \text{e} \quad (4.17)$$

$$Int^{\lceil \frac{t}{5} \rceil + 1} = Int^{\lceil \frac{t}{5} \rceil} + \omega_{Int}^{\lceil \frac{t}{5} \rceil + 1}. \quad (4.18)$$

Finalmente, algumas definições extras têm de ser apresentadas, para a definição do conjunto paramétrico que será usado no modelo dinâmico. Essas definições estão na equação (4.19),

$$\begin{aligned} At^k &= (At_1^k, \dots, At_N^k), \quad k = 1, \dots, \lceil \frac{T}{5} \rceil, \\ Ca^k &= (Ca_1^k, \dots, Ca_N^k), \quad k = 1, \dots, \lceil \frac{T}{5} \rceil, \\ De^k &= (De_1^k, \dots, De_N^k), \quad k = 1, \dots, \lceil \frac{T}{5} \rceil, \\ \omega_{At}^k &= (\omega_{1, At}^k, \dots, \omega_{N, At}^k), \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\ \omega_{Ca}^k &= (\omega_{1, Ca}^k, \dots, \omega_{N, Ca}^k), \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\ \omega_{De}^k &= (\omega_{1, De}^k, \dots, \omega_{N, De}^k), \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\ At &= (At^1, \dots, At^{\lceil \frac{T}{5} \rceil}), Ca = (Ca^1, \dots, Ca^{\lceil \frac{T}{5} \rceil}), De = (De^1, \dots, De^{\lceil \frac{T}{5} \rceil}), \\ \omega_{At} &= (\omega_{At}^2, \dots, \omega_{At}^{\lceil \frac{T}{5} \rceil}), \omega_{Ca} = (\omega_{Ca}^2, \dots, \omega_{Ca}^{\lceil \frac{T}{5} \rceil}), \omega_{De} = (\omega_{De}^2, \dots, \omega_{De}^{\lceil \frac{T}{5} \rceil}), \\ Int &= (Int^1, \dots, Int^{\lceil \frac{T}{5} \rceil}), \omega_{Int} = (\omega_{Int}^2, \dots, \omega_{Int}^{\lceil \frac{T}{5} \rceil}), \\ \theta &= (At, Ca, De, Int, \omega_{At}, \omega_{Ca}, \omega_{De}, \omega_{Int}) \end{aligned} \quad (4.19)$$

As prioris usadas neste trabalho para modelos dinâmicos gerados a partir da estrutura definida entre as equações (4.1) e (4.9) estão apresentadas na equação (4.20),

$$\begin{aligned}
At_i^1 &\sim N(0, 10000), \quad i = 1, \dots, N, \\
Ca_i^1 &\sim N(0, 10000), \quad i = 1, \dots, N, \\
De_i^1 &\sim N(0, 10000), \quad i = 1, \dots, N, \\
Int^1 &\sim N(0, 10000), \\
\omega_{i,At}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \left\lceil \frac{T}{t_2} \right\rceil, \\
\omega_{i,Ca}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \left\lceil \frac{T}{t_2} \right\rceil, \\
\omega_{i,De}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \left\lceil \frac{T}{t_2} \right\rceil, \\
\omega_{Int}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \left\lceil \frac{T}{t_2} \right\rceil,
\end{aligned} \tag{4.20}$$

portanto, as prioris para os fatores de qualidade têm forma semelhante à apresentada nas equações (3.8) a (3.11), pelo menos para os primeiros componentes dos vetores At, Ca e De. Pode-se ver, também, em (4.20) que as perturbações periódicas aos valores dos parâmetros são perturbações normais com média 0 e variância σ_ω^2 .

Como mostrado na seção 4.1, foram testados diversos modelos dinâmicos, com diferentes valores para t_2 . Todavia, os modelos testados não diferiam apenas no valor para t_2 , mas, também, nos valores para σ_ω^2 . Em alguns modelos, lidou-se com essa variância como se ela fosse uma constante, em outros, ela foi vista como uma variável aleatória, e recebeu sua própria priori. Mais comentários sobre os modelos comparados estão na seção A.2.

Ao final das comparações, como apontado em 4.1, decidiu-se por um modelo em que t_2 é igual a cinco e em que a variância das perturbações normais, σ_ω^2 , foi considerada uma variável aleatória, e não uma constante. Assim, foi necessário atualizar a definição de θ^t a partir do que foi definido em (4.19), como pode-se ver em (4.21),

$$\theta^t = (At^t, Ca^t, De^t, Int^t, \omega_{At}^t, \omega_{Ca}^t, \omega_{De}^t, \omega_{Int}^t, [\sigma_\omega^2]^t), \tag{4.21}$$

enquanto continua a valer a equação (4.14).

As prioris para todos os parâmetros do modelo, dadas em (4.20), tiveram de ser

atualizadas para a forma que se vê em 4.22,

$$\begin{aligned}
At_i^1 &\sim N(0, 10000), \quad i = 1, \dots, N, \\
Ca_i^1 &\sim N(0, 10000), \quad i = 1, \dots, N, \\
De_i^1 &\sim N(0, 10000), \quad i = 1, \dots, N, \\
Int^1 &\sim N(0, 10000), \\
\omega_{i,At}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\omega_{i,Ca}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\omega_{i,De}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\omega_{Int}^k &\sim N(0, \sigma_\omega^2), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\frac{1}{\sigma_\omega^2} &\sim Gama(0, 1; 0, 1),
\end{aligned} \tag{4.22}$$

onde cada uma das prioris é independente entre si.

Destarte, as equações (4.10) a (4.18) e (4.21) definem o modelo dinâmico usado, enquanto as equações em (4.22) definem as prioris utilizadas para os parâmetros desse modelo. Assim a inclusão de conhecimento prévio sobre os parâmetros via uma priori está completa. Já a inclusão do conhecimento adquirido via os dados será feita por meio da função de verossimilhança do modelo dinâmico, apresentada em (4.23),

$$l(\theta; x) = \prod_{t=1}^T \prod_{i=1}^N \prod_{j=1, j \neq i}^N \frac{(\lambda_{ij}^{\lceil \frac{t}{5} \rceil})^{x_{ij}^t} \exp(-\lambda_{ij}^{\lceil \frac{t}{5} \rceil})}{x_{ij}^t!}, \tag{4.23}$$

que difere um pouco da verossimilhança do modelo estático, apresentada em (3.19). A verossimilhança do modelo dinâmico tem essa forma devido à independência condicional dos dados e à definição do modelo dinâmica, feita entre as equações (4.10) a (4.18), junto com a equação (4.21)

O modelo dinâmico, assim como o estático, não é identificável. A restrição aplicada aos parâmetros para que ele se torne identificável é semelhante à aplicada ao modelo estático, descrita na seção 3.2 e definida em (3.20). Mas ela é diferente, pois agora a soma dentro de cada tipo de fator deve ser nula em todos os pontos de atualização dos valores

dos parâmetros, conforme define (4.24),

$$\sum_{i=1}^N At_i^k = 0, \quad \sum_{i=1}^N Ca_i^k = 0, \quad \sum_{i=1}^N De_i^k = 0, \quad k = 1, \dots, \lceil \frac{T}{5} \rceil. \quad (4.24)$$

Finalmente, pode-se usar exemplos dos conjuntos de dados usados neste trabalho. Na Figura 4.1 estão representadas as estimativas do modelo estático para a qualidade dos ataques dos times do Campeonato brasileiro de 2017. Na linha do Grêmio (identificada como GRE), pode-se ver que o time, ao final do campeonato teve seu ataque estimado como o melhor da competição.

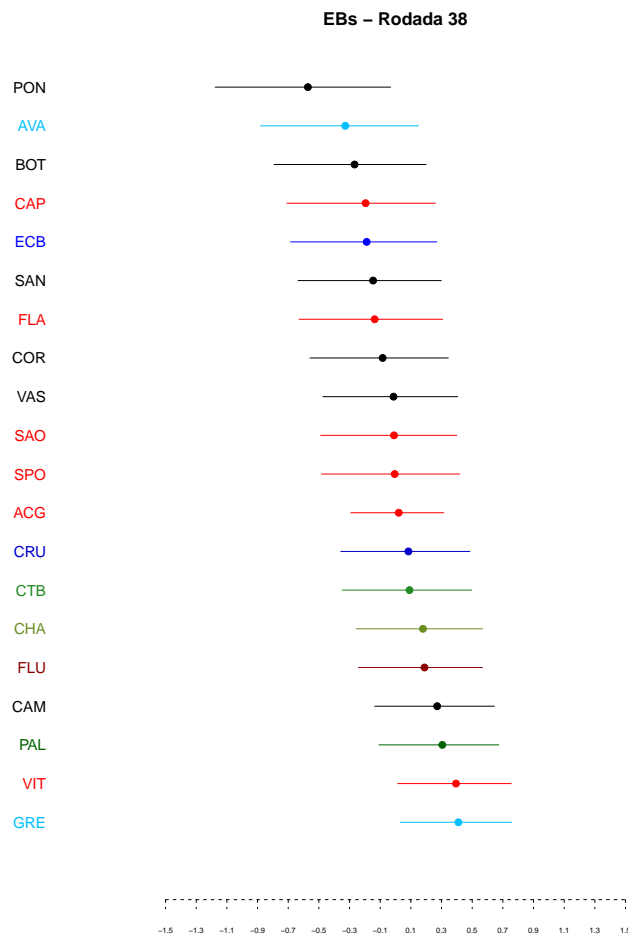


Figura 4.1: Estimativas do fator ataque para o Campeonato brasileiro de 2017.

No entanto, quando se olha para a Figura 4.2, na qual estão as estimativas do modelo dinâmico para o ataque da equipe ao longo do tempo, é possível perceber que a qualidade do ataque do time foi gradualmente diminuindo ao longo do tempo, o que significa que o modelo estático estava superestimando a capacidade da equipe de fazer gols, no final do torneio.

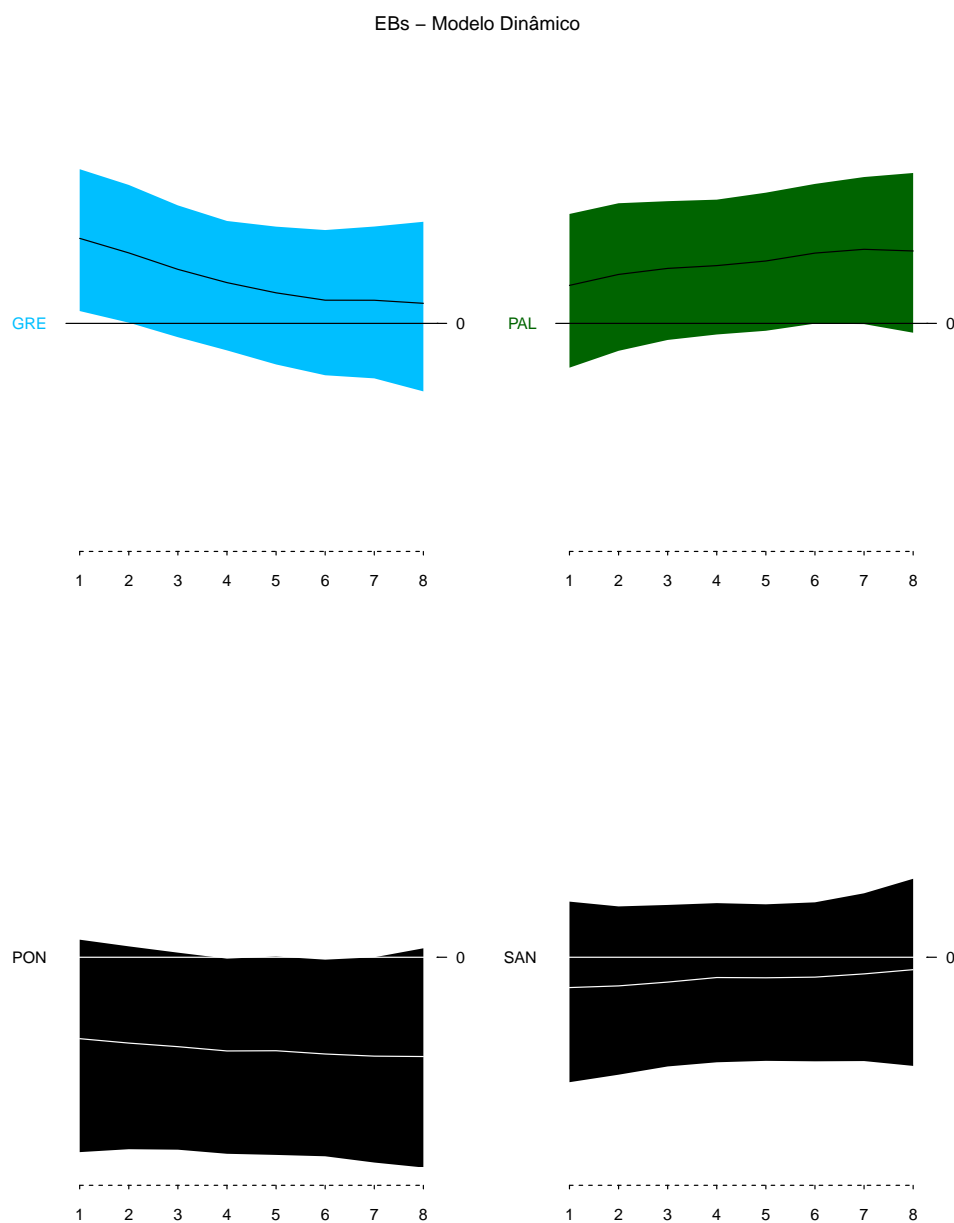


Figura 4.2: Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 4.

No Campeonato brasileiro de 2016 aconteceu o contrário com a defesa do Botafogo. Como se pode ver na Figura 4.3, onde encontram-se as estimativas do modelo estático para os fatores de defesa das equipes do campeonato, o Botafogo teve uma das melhores defesas do campeonato.

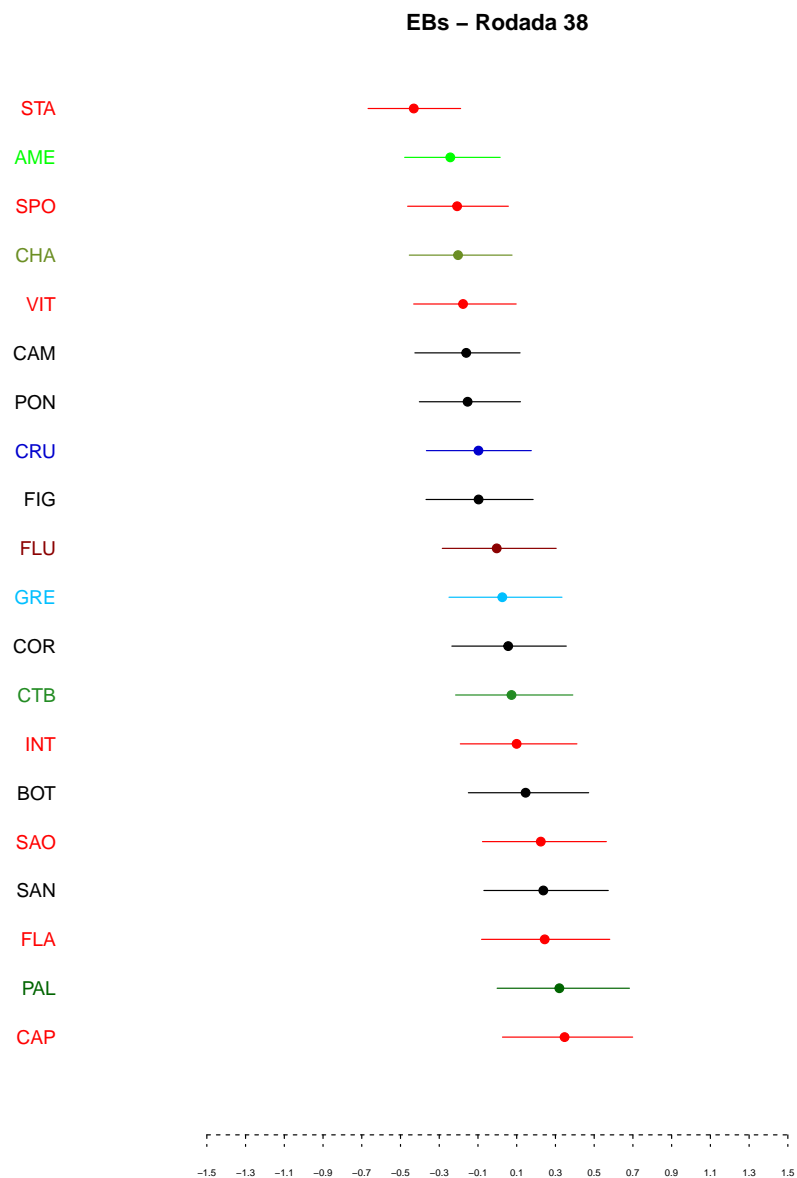


Figura 4.3: Estimativas do fator defesa para o Campeonato brasileiro de 2016.

Entretanto, a Figura 4.4 deixa claro que o time terminou com uma das melhores defesas da competição, mas que começou o torneio com uma defesa mediana. Isso significa que estimativas que não levam em conta a variação temporal dos parâmetros teriam superestimado a qualidade de sua defesa no começo do campeonato, mas a teriam subestimado ao final do mesmo.

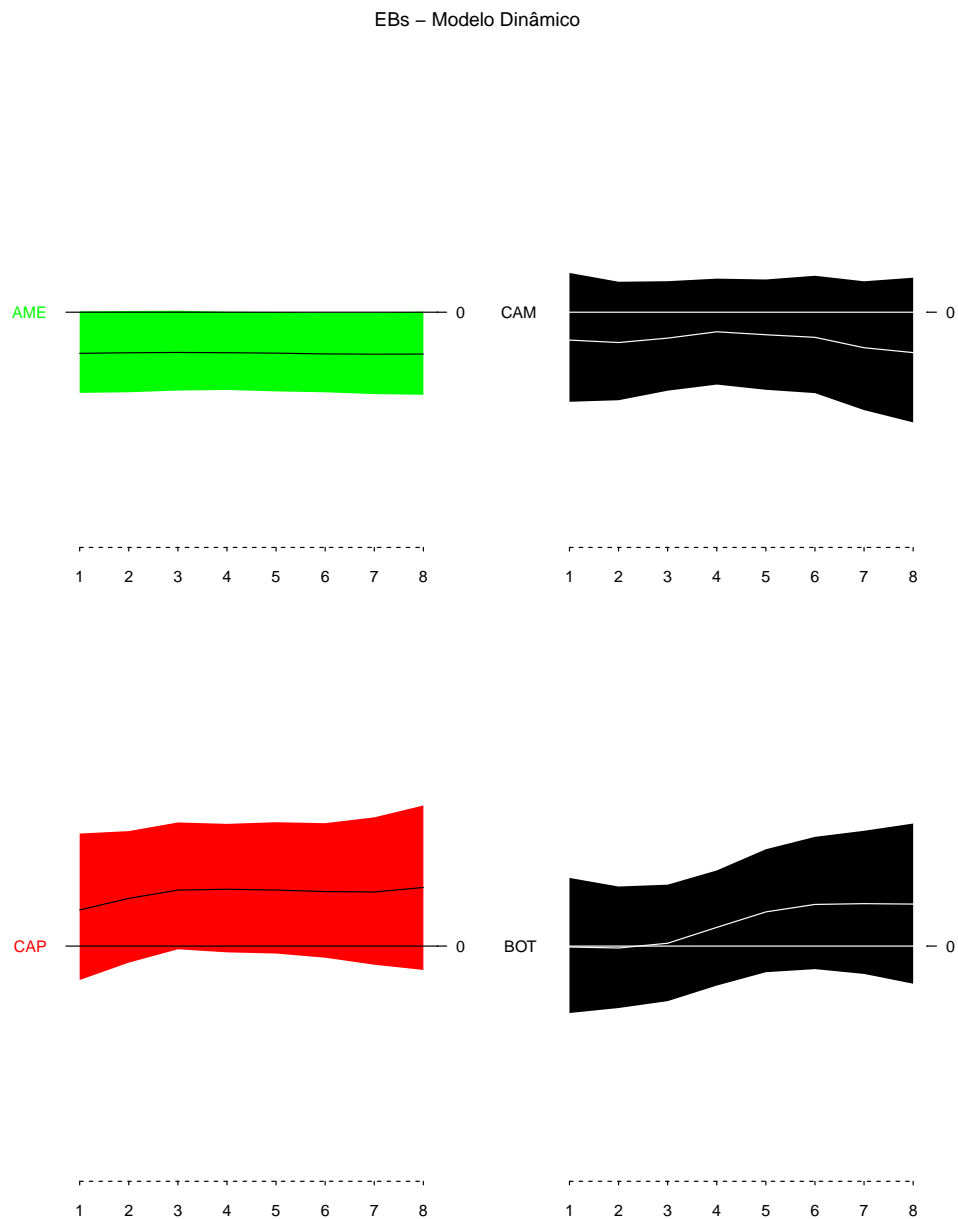


Figura 4.4: Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 1.

4.3 Previsões no modelo dinâmico

No tocante à inferência bayesiana, algumas coisas se mantêm, para o modelo dinâmico. Continuam a existir múltiplas posteriores, uma para cada conjunto de dados utilizado, e elas continuam sendo impossíveis de encontrar analiticamente. Continua-se empregando simulação via MCMC para se obter amostras dessas posteriores.

A amostra aproximada encontrada para a posteriori de θ^t é mostrada em (4.25),

$$\begin{aligned} \theta^{t;(a)} = (At^{t;(a)}, Ca^{t;(a)}, De^{t;(a)}, Int^{t;(a)}, \omega_{At}^{t;(a)}, \omega_{Ca}^{t;(a)}, \omega_{De}^{t;(a)}, \omega_{Int}^{t;(a)}, [\sigma_\omega^2]^{t;(a)}), \\ a = 1, \dots, n, \quad t = 1, \dots, \lceil \frac{T}{5} \rceil, \end{aligned} \quad (4.25)$$

e é análoga à amostra da posteriori para o vetor paramétrico do modelo estático, exibida em (3.21), diferente por causa da inclusão do índice t , que sinaliza o ponto de atualização dos valores dos parâmetros do qual a amostra faz parte, e por causa da definição de θ^t dada em (4.21).

Quando se quer fazer previsões para uma rodada que é a seguinte a uma de número múltiplo de cinco (por exemplo, previsões para a rodada 26 depois da rodada 25), é necessária a atualização dos valores dos parâmetros. Portanto, é necessária uma amostra para a posteriori de θ^{t+1} , caso a t -ésima rodada tenha acabado de ser completada.

A partir da amostra da posteriori de θ^t é possível gerar uma amostra aproximada para a posteriori de θ^{t+1} . Mas, para isso, é preciso estabelecer algumas relações, presentes em (4.26),

$$\begin{aligned} At^{k+1} &= At^k + \omega_{At}^{k+1}, \\ Ca^{k+1} &= Ca^k + \omega_{Ca}^{k+1}, \\ De^{k+1} &= De^k + \omega_{De}^{k+1}, \end{aligned} \quad (4.26)$$

que dependem das definições dadas em (4.19).

Para que se possa gerar a amostra da posteriori de θ^{t+1} , faz-se uso, também do que é

definido em (4.22), como se vê em (4.27),

$$\begin{aligned}
\omega_{At}^{t;(a)} &\sim N_N(0, [\sigma_\omega^2]^{t;(a)} I_N), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\omega_{Ca}^{t;(a)} &\sim N_N(0, [\sigma_\omega^2]^{t;(a)} I_N), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\omega_{De}^{t;(a)} &\sim N_N(0, [\sigma_\omega^2]^{t;(a)} I_N), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil, \\
\omega_{Int}^{t;(a)} &\sim N(0, [\sigma_\omega^2]^{t;(a)} I_N), \quad i = 1, \dots, N, \quad k = 2, \dots, \lceil \frac{T}{5} \rceil,
\end{aligned} \tag{4.27}$$

onde é estabelecida a distribuição de probabilidade dos elementos das amostras das posteriores das perturbações normais.

Finalmente, usando (4.26) e (4.27) pode-se gerar a amostra da posteriori de θ^{t+1} , mostrada em (4.28),

$$\begin{aligned}
At^{t+1;(a)} &= At^{t;(a)} + \omega_{At}^{t+1;(a)}, \\
Ca^{t+1;(a)} &= Ca^{t;(a)} + \omega_{Ca}^{t+1;(a)}, \\
De^{t+1;(a)} &= De^{t;(a)} + \omega_{De}^{t+1;(a)}, \\
Int^{t+1;(a)} &= Int^{t;(a)} + \omega_{Int}^{t+1;(a)}, \\
\theta^{t+1;(a)} &= (At^{t+1;(a)}, Ca^{t+1;(a)}, De^{t+1;(a)}, Int^{t+1;(a)}, \omega_{At}^{t+1;(a)}, \omega_{Ca}^{t+1;(a)}, \omega_{De}^{t+1;(a)}, \omega_{Int}^{t+1;(a)}, [\sigma_\omega^2]^{t;(a)}), \\
a &= 1, \dots, n, \quad t = 1, \dots, \lceil \frac{T}{5} \rceil,
\end{aligned} \tag{4.28}$$

Tendo as amostras das posteriores dos parâmetros, é possível gerar a preditiva do modelo dinâmico, usando cálculos análogos aos mostrados na seção 2.4, pois o modelo dinâmico é um caso especial do modelo apresentado na seção 2.4.

Portanto, é possível chegar à definição de densidade preditiva usada para o modelo dinâmico, como mostra (4.29),

$$f(x^{\{s+1;t\}} | x^{\{r;s\}}) = \int_{-\infty}^{\infty} f(x^{\{s+1;t\}} | \theta^s) \xi(\theta^s | x^{\{r;s\}}) d\theta^s. \tag{4.29}$$

4.4 Qualidade de previsões no modelo dinâmico

Para que se possa comparar o modelo dinâmico com os outros modelos, as verossimilhanças preditivas do modelo foram comparadas com as de outros modelos. Na Tabela 4.1 estão as verossimilhanças preditivas padronizadas para o Campeonato inglês, coletadas a partir da rodada em que se julgou ter ocorrido estabilização das verossimilhanças preditivas do modelo estático, por preocupações computacionais.

Tabela 4.1: Verossimilhanças preditivas para o Campeonato inglês 2016/2017.

Rodada	Dinâmico	Estático	538	CdG	Nulo	Simples
28	50,34	56,64	30,17	27,39	1	5,62
29	6,09	7,66	5,94	8,36	1	1,94
30	0,78	0,56	0,27	0,31	1	0,73
31	10,71	10,14	12,4	12,47	1	6,03
32	33,24	35	52,15	50,46	1	2,24
33	35,7	39,04	39,31	21,64	1	1,94
34	29,91	23,25	17,62	10,44	1	4,24
35	0,29	0,24	0,49	0,5	1	0,12
36	0,13	0,19	1,02	1,9	1	2,71
37	131,24	145,31	78,47	79,57	1	2,81
38	52,13	48,29	64,77	32,21	1	2,47
Total	$2,4 \cdot 10^{10}$	$2,52 \cdot 10^{10}$	$5,62 \cdot 10^{10}$	$2,47 \cdot 10^{10}$	1	1977,07

Pode-se ver nessa tabela que o desempenho do modelo dinâmico foi muito semelhante ao do estático, nesse conjunto de dados. Quando são levados em conta os outros dois campeonatos, o modelo dinâmico se destaca, como pode ser visto na Tabela 4.2.

A partir da tabela, é possível observar o quão bom foi o desempenho do modelo

Tabela 4.2: Medidas de qualidade dos modelos.

Modelo	VP	VP^N	Jogos	VPP
Dinâmico	$9,59 \cdot 10^{-181}$	$7,51 \cdot 10^{19}$	419	0,3718
Estático	$6,27 \cdot 10^{-181}$	$5,14 \cdot 10^{19}$	419	0,3715
538	$1,14 \cdot 10^{-112}$	$1,28 \cdot 10^{12}$	260	0,3706
UFMG	$7,06 \cdot 10^{-111}$	0,13	229	0,3304
CdG	$3,37 \cdot 10^{-181}$	$2,76 \cdot 10^{19}$	419	0,3709
Nulo	$1,22 \cdot 10^{-200}$	1	419	0,3333
Simples	$1,59 \cdot 10^{-189}$	$1,3 \cdot 10^{11}$	419	0,3543

dinâmico, comparado com os outros modelos, e mesmo com o estático. Tudo isso parece apontar para o modelo dinâmico como um bom modelo para previsão de resultados de futebol, até melhor que o estático. Portanto, os resultados parecem apontar como razoáveis as presunções de que a qualidade de um time varia com o tempo, e que rodadas diferentes devem ter influências diferentes na estimação dessa qualidade do time.

Capítulo 5

Conclusão

O modelo estático discutido neste trabalho teve excelente desempenho em suas previsões, de acordo com uma medida de qualidade de previsões, a verossimilhança preditiva. Seu desempenho foi comparado com diversos modelos, alguns simples e outros vindos de pesquisadores respeitados. Modelos como o do *FiveThirtyEight* e do Chance de Gol são frequentemente considerados modelos confiáveis pelo público que acompanha o esporte. Já os modelos nulo e simples oferecem níveis mínimos de qualidade que um modelo deve alcançar, e o modelo estático passou desses níveis com facilidade.

Apesar de ter definição estatística robusta, o modelo estático não é especialmente complexo. Isso só torna a qualidade de seus resultados mais impressionante. Principalmente porque modelos como o do *FiveThirtyEight* e do Chance de Gol são modelos consideravelmente mais complexos, usando estatísticas referentes ao desempenho das equipes envolvidas, além de arcabouços estatísticos que também parecem ser sólidos.

Todavia, é necessário ressaltar que o desempenho do *FiveThirtyEight* foi melhor que o do modelo estático, quando são descartados partidas do Campeonato brasileiro de 2016 e levadas em conta apenas partidas dos outros dois campeonatos. Isso não significa, necessariamente, que os resultados do modelo estático seriam piores caso fossem incluídos os 159 jogos do Campeonato brasileiro de 2016, afinal, o desempenho do modelo nesses dados foi excelente. Ainda assim, é bom se ter em mente essa ressalva, de que o modelo do *FiveThirtyEight* teve desempenho mais impressionante, em parte dos dados.

Além disso, é preciso apontar que o modelo estático, por ser razoavelmente simples, parte de algumas presunções teóricas que são questionáveis. Especificamente, as presunções de que a qualidade de um time não varia com o tempo e de que todas as rodadas de um campeonato devem ter peso igual, na estimação dos parâmetros do modelo, parecem pouco razoáveis, com base em todo o conhecimento que se tem sobre futebol, e como times se comportam ao longo do tempo.

O modelo dinâmico, por outro lado, não é definido com essas presunções em mente. Ele permite que os parâmetros variem com o tempo, e que cada rodada tenha influência de magnitude própria na estimação dos mesmos. Dessa forma, o modelo dinâmico permite que o pesquisador faça mais testes, em cima de seus dados, e tire mais conclusões.

Sob a ótica de qualidade de previsões, medida por verossimilhança preditiva, o modelo dinâmico teve desempenho melhor que o do estático. Quando são observados todos os jogos dos três campeonatos, há uma diferença significativa entre as qualidades de desempenho dos dois modelos.

Por outro lado, quando os modelos são comparados sob o ponto de vista do princípio da parcimônia, a resposta pode mudar. O modelo dinâmico tem resultados melhores, mas ele também é significativamente mais complexo que o estático. Dessa maneira, seria razoável concluir que o modelo estático é melhor, já que tem resultados muito semelhantes com muito menos gastos de tempo e computacionais, além de ter interpretação mais simples.

Entretanto, é necessário considerar algo que foi levantado no capítulo 1. Os esportes, de forma geral, mas o futebol, especialmente, envolvem imensas quantidades de dinheiro. Num mundo capitalista, isso é de extrema importância, e qualquer pequena vantagem pode trazer imensos lucros. Ou vitórias, caso se encare a situação pelo ponto de vista de alguém envolvido diretamente com um time. É razoável, portanto, considerar o modelo dinâmico como o melhor dos dois.

Uma crítica que pode ser feita a ambos modelos, não só ao dinâmico, é de que eles se baseiam apenas no número de gols feitos pelos times. Isso é um problema, e o porquê é exemplificado por trabalhos como Vizzoni (2015), no qual foi defendido que um gol pode ser dividido em dois momentos: o momento até o chute ser tentado pelo jogador de ataque, e o que aconteceu para que a bola entrasse no gol adversário.

De acordo com a teoria apresentada, o primeiro momento é uma marca melhor da habilidade de um time do que o segundo, uma vez que um jogador e seu time exercem muito mais controle sobre o que acontecerá ao fim da jogada até o chute, pois a bola está em sua posse. Após o chute, contudo, muitos são os fatores importantes para o que acontecerá com a bola: tais como a defesa e o goleiro adversários, o vento e o estado de conservação do campo.

Dessa forma, este trabalho aponta um caminho para trabalhos futuros. Pode-se usar modelos com o mesmo arcabouço, mas que deem espaço para outras estatísticas referentes ao desempenho dos times. Ou modelos que continuem a usar as equações (4.6) a (4.9)

para definir seus parâmetros, mas que definam as perturbações normais como tendo distribuições diferentes, de acordo com o fator.

Isso implicaria que a equação (3.20) deixaria de ser usada, pois cada ω_{At} teria variância diferente de cada ω_{Ca} ou ω_{De} . Conceitualmente, isso significaria que o modelo possibilitaria comportamentos diferentes para cada fator de uma equipe. Por exemplo, se um pesquisador acredita ter bons motivos para considerar que o ataque de um time é menos consistente que sua defesa, ao longo do tempo, ele definiria a variância da perturbação no fator de ataque como sendo maior que a da perturbação no fator de defesa. Isso tornaria fatores de ataque mais voláteis que fatores de defesa.

Poderiam ser incluídas perturbações inter-temporada ao modelo dinâmico, de forma semelhante à presente em Glickman e Stern (1998). Destarte, o modelo permitiria que os parâmetros variassem não só dentro de uma mesma temporada, mas entre uma temporada e outra. Ou seja, a qualidade do Flamengo em 2016 teria influência em sua qualidade em 2017, por exemplo. Isso implicaria a criação de novas perturbações aleatórias. Consequentemente, seria necessário pensar num valor para a variância dessa perturbação inter-temporada.

Finalmente, um trabalho futuro poderia usar estatísticas referentes ao desempenho dos jogadores de cada time, para ter uma medida mais robusta da qualidade do ataque, da defesa e do mando de campo das equipes. Assim, espera-se que as previsões de um modelo desse tipo fossem ainda melhores.

Porém, as críticas e as possíveis melhorias aos modelos não devem tirar atenção de sua qualidade. Ambos os modelos fizeram precisões que foram avaliadas como de boa qualidade, mesmo quando comparadas com as previsões de modelos criados por profissionais renomados no ambiente futebolístico.

Apêndice A

Nesse Apêndice estão figuras e tabelas, algumas das quais já foram citadas no texto. Sobre elas são feitas análises e comentários. Visto que são diversos gráficos e tabelas, e que alguns deles só devem ser interpretados em conjunto com outros elementos, não são feitos comentários sobre absolutamente todas as figuras e tabelas. No entanto, dentro de cada seção e subseção há espaço para comentários sobre o conjunto de imagens ou tabelas contido dentro dela.

O Apêndice está estruturado de forma que cada campeonato estudado tenha uma seção devotada a ele, exclusivamente. Entretanto, a primeira seção do Apêndice, a seção [A.1](#) contém as siglas utilizadas para identificar os times que participaram de cada competição, enquanto a seção [A.2](#) contém análises e comentários sobre o desempenho dos modelos dinâmicos que foram descartados, como foi discutido no capítulo 4.

A análise dos campeonatos em si começa na seção [A.3](#), e nesta seção são apresentados análises e comentários sobre os resultados dos modelos para o Campeonato brasileiro de 2016. Eles estão divididos de acordo com o modelo usado. Na primeira subseção, [A.3.1](#), são mostradas figuras relacionadas ao modelo estático e nela são discutidas as EBS pontuais e intervalares para os parâmetros do modelo estático.

Na segunda subseção, [A.3.2](#), o modelo em questão é o dinâmico, e discute-se as EBS dos parâmetros desse modelo. Finalmente, na terceira subseção [A.3.3](#), são mostradas as verossimilhanças preditivas para cada modelo, para esse campeonato.

Essa divisão se repete para os outros dois campeonatos estudados. Análises para o Campeonato brasileiro de 2017 encontram-se na seção [A.4](#), na qual há subseções para os modelos estático (subseção [A.4.1](#)) e dinâmico (subseção [A.4.2](#)), além de uma subseção para as verossimilhanças preditivas para os modelos avaliados, a subseção [A.4.3](#). Já as análises para o Campeonato inglês estão na seção [A.5](#), dividida em subseções [A.5.1](#), [A.5.2](#) e [A.5.3](#).

A.1 Siglas utilizadas

As siglas utilizadas para o Campeonato brasileiro de 2016 estão relacionadas a seguir:

- AME -> América-MG.
- CAM -> Atlético-MG.
- CAP -> Atlético-PR.
- BOT -> Botafogo.
- CHA -> Chapecoense.
- COR -> Corinthians.
- CTB -> Coritiba.
- CRU -> Cruzeiro.
- FIG -> Figueirense.
- FLA -> Flamengo.
- FLU -> Fluminense.
- GRE -> Grêmio.

- INT -> Internacional.
- PAL -> Palmeiras.
- PON -> Ponte Preta.
- STA -> Santa Cruz.
- SAN -> Santos.
- SAO -> São Paulo.
- SPO -> Sport Recife .
- VIT -> Vitória.

Quanto ao Campeonato brasileiro de 2017, as siglas utilizadas são:

- ACG -> Atlético-GO .
- CAM -> Atlético-MG .
- CAP -> Atlético-PR .
- AVA -> Avaí.
- ECB -> Bahia.
- BOT -> Botafogo.
- CHA -> Chapecoense .
- COR -> Corinthians .
- CTB -> Coritiba.
- CRU -> Cruzeiro.
- FLA -> Flamengo.
- FLU -> Fluminense.
- GRE -> Grêmio.

- PAL -> Palmeiras.
- PON -> Ponte Preta .
- SAN -> Santos.
- SAO -> São Paulo.
- SPO -> Sport Recife.
- VAS -> Vasco.
- VIT -> Vitória.

Para o Campeonato inglês, as siglas usadas estão a seguir:

- BOU -> AFC Bournemouth.
- ARS -> Arsenal FC.
- BUR -> Burnley FC.
- CHE -> Chelsea FC.
- CRY -> Crystal Palace.
- EVE -> Everton FC.
- HUL -> Hull City.
- LEI -> Leicester City.
- LIV -> Liverpool FC.
- MCI -> Manchester City.
- MUN -> Manchester United.
- MID -> Middlesbrough FC.
- SOU -> Southampton FC.

- STK -> Stoke City.
- SUN -> Sunderland AFC.
- SWA -> Swansea City.
- TOT -> Tottenham Hotspur.
- WAT -> Watford FC.
- WBA -> West Bromwich Albion.
- WHU -> West Ham United.

A.2 Comparação entre modelos dinâmicos

Como descrito no capítulo 4, foram testados diversos modelos que foram gerados por meio da estrutura de modelos dinâmicos, definida entre as equações (4.1) e (4.9). Eles diferiram de acordo com o valor usado para t_2 e de acordo com o tratamento conferido a σ_ω^2 . Alguns trataram a variância das perturbações normais como uma constante, enquanto outros a trataram como uma variável aleatória, cuja priori foi definida em (4.22).

Nas Figuras A.1 a A.3 estão as verossimilhanças preditivas de alguns dos modelos para cada um dos campeonatos estudados. Cada modelo em que foi conferida uma priori à variância das perturbações é identificado de acordo com o valor de t_2 usado na definição do modelo. Dessa forma, um modelo identificado como "D-5" é um modelo dinâmico com $t_2 = 5$ e priori para a variância. Da mesma maneira, modelos que trataram a variância como uma constante foram identificados também com o valor que deram a ela. Portanto, um modelo identificado como "D-5-0,05" é um modelo com $t_2 = 5$ e $\sigma_\omega^2 = 0,05$.

Uma análise das figuras mostra que o modelo "D-5" teve as melhores verossimilhanças preditivas do conjunto de modelos. Além disso, esse modelo foi considerado como tendo a definição estatística mais robusta, por permitir maior variabilidade a seus parâmetros. Assim, o modelo dinâmico definido neste trabalho é o modelo identificado como "D-5" nas Figuras 100 a 102.

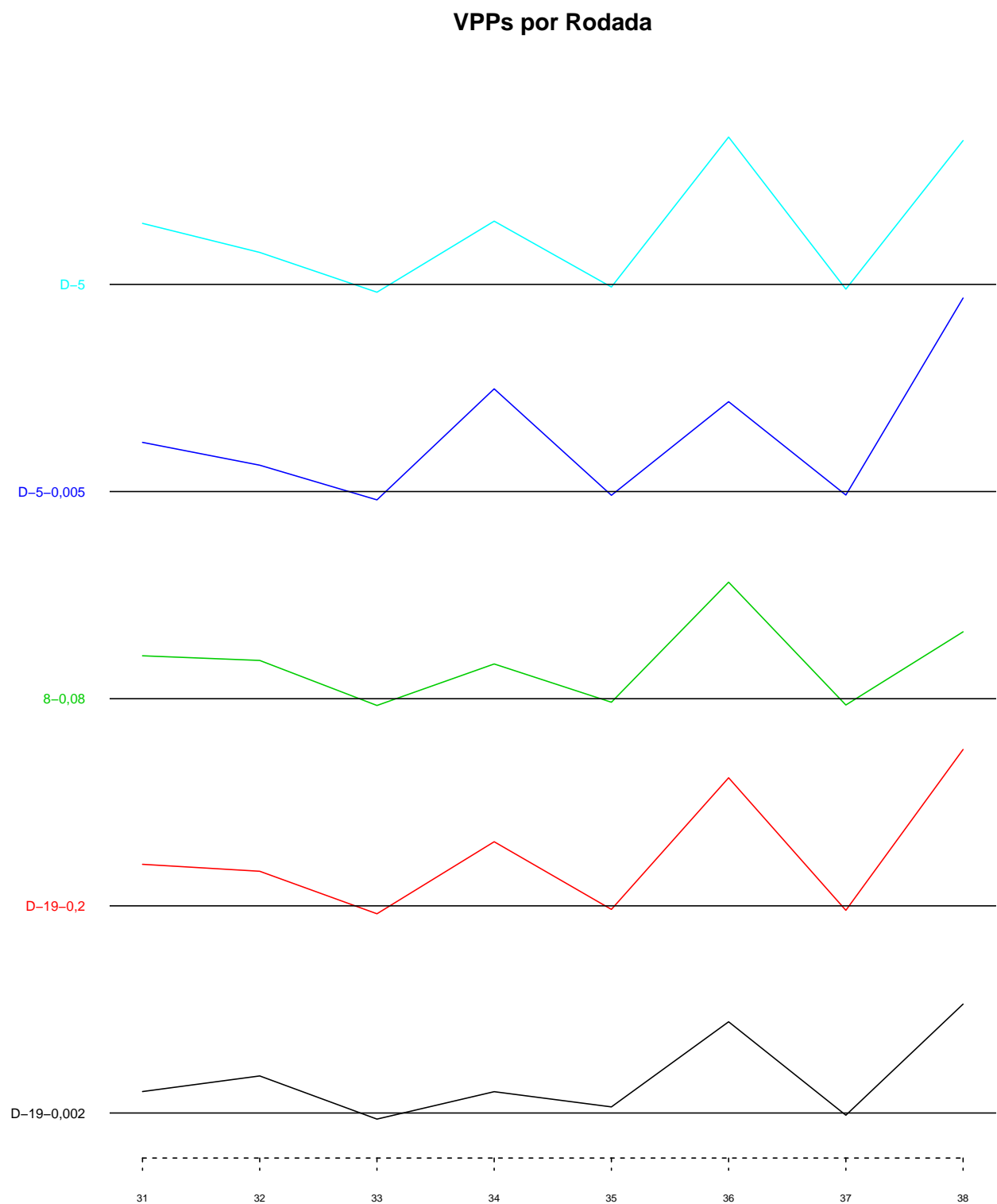


Figura A.1: Estimativas do fator ataque para o Campeonato brasileiro de 2016.

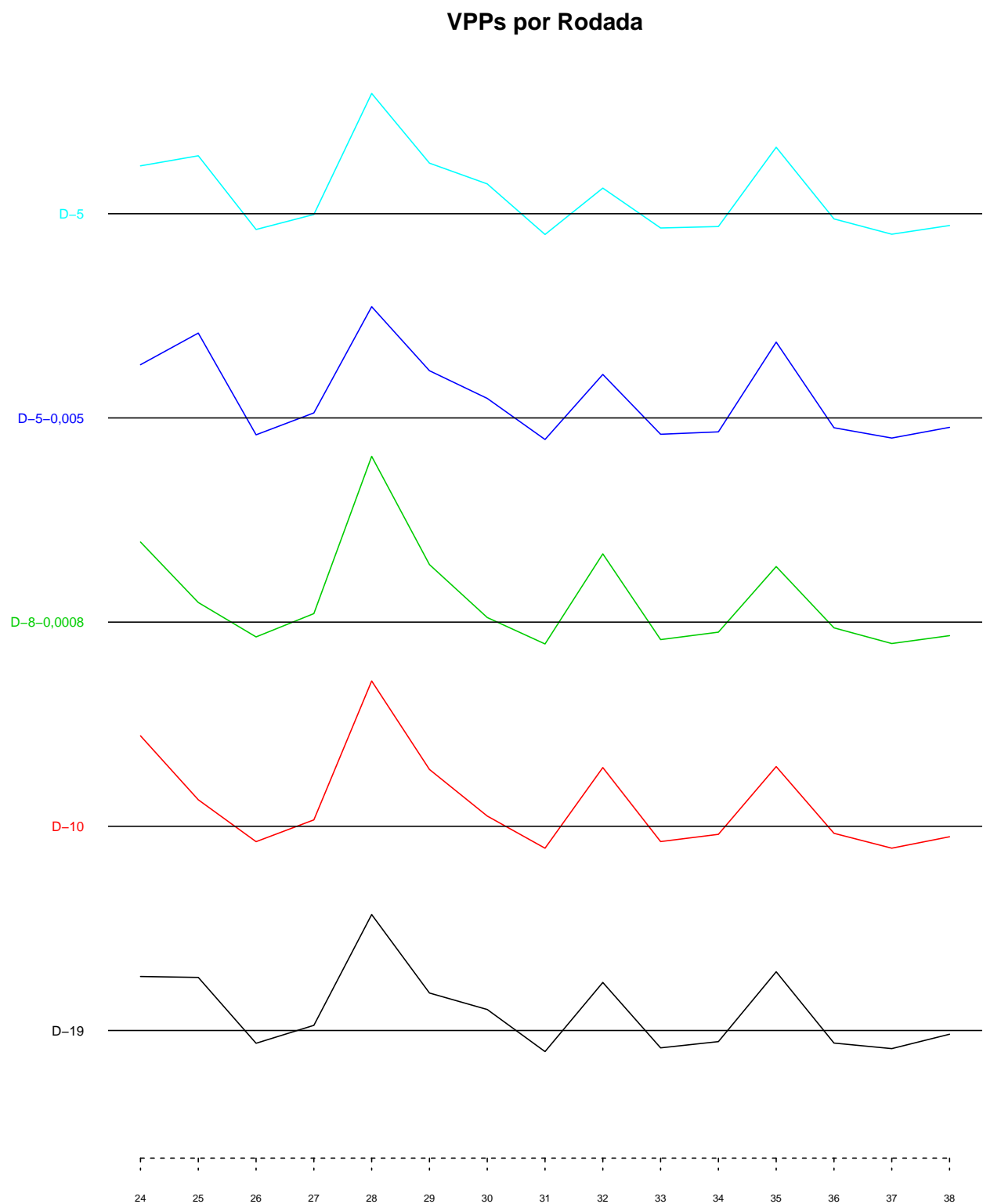


Figura A.2: Estimativas do fator campo para o Campeonato brasileiro de 2016.

VPPs por Rodada

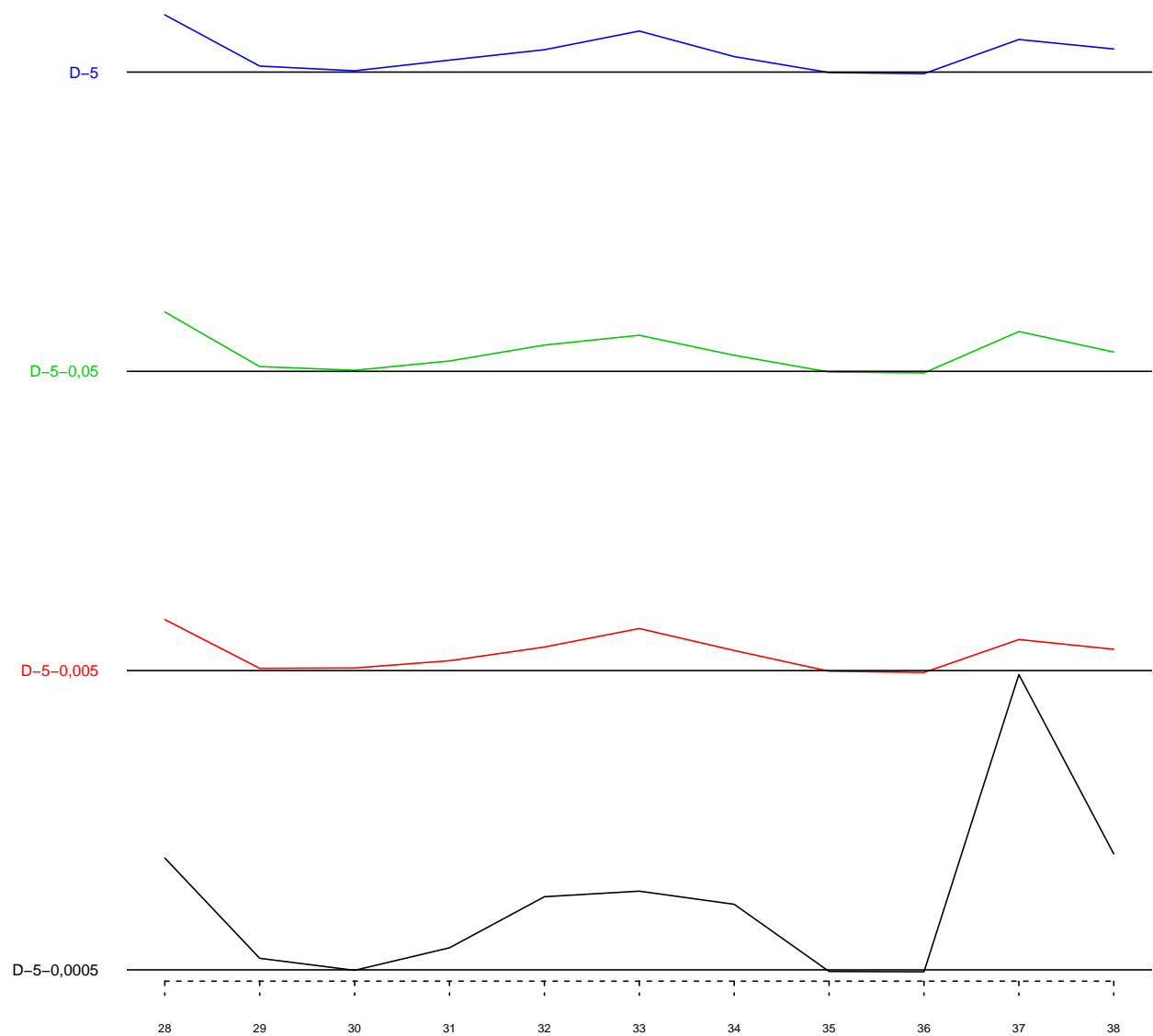


Figura A.3: Estimativas do fator defesa para o Campeonato brasileiro de 2016.

A.3 Campeonato brasileiro de 2016

Tal como foi escrito anteriormente, nesta seção do Apêndice estão comentários mais completos sobre as figuras e tabelas deste trabalho que estão relacionadas com os resultados dos modelos para o campeonato brasileiro de 2016. Aqui não só se explicará de forma mais completa o que cada figura representa, como serão mostradas figuras que não apareceram no texto, anteriormente.

A.3.1 Análises e resultados do Modelo estático

Nessa subseção são mostradas as estimativas para os parâmetros do modelo estático. Para cada fator de qualidade, as estimativas pontuais mostradas são estimativas de Bayes (EBs) com função de perda quadrática, enquanto as estimativas intervalares são intervalos de credibilidade aproximados de 95%.

Nas Figuras [A.4](#) a [A.6](#) estão as EBs para os fatores de ataque, campo e defesa de cada equipe. Na Figura [A.4](#), assim como nas subsequentes, os times são identificados de acordo com sua sigla, que pode ser encontrada na seção [A.1](#). Escolheu-se identificá-los dessa forma para que as Figuras não apresentassem informações demais, e se tornassem uma leitura complicada, para os leitores deste trabalho.

Todas as figuras estão ordenadas de maneira crescente. Ou seja, a primeira linha é referente ao time com a menor estimativa para seu fator, dentro de cada conjunto de fatores. Já a última linha é referente ao time com a maior estimativa. Portanto, o fato de o Figueirense ser o primeiro time a aparecer na Figura [A.4](#) significa que seu ataque foi estimado como o pior do campeonato brasileiro de 2016. Enquanto isso, a defesa do Atlético-PR foi estimada como a melhor do torneio, e, por isso, ele é o último time a aparecer na Figura [A.6](#).

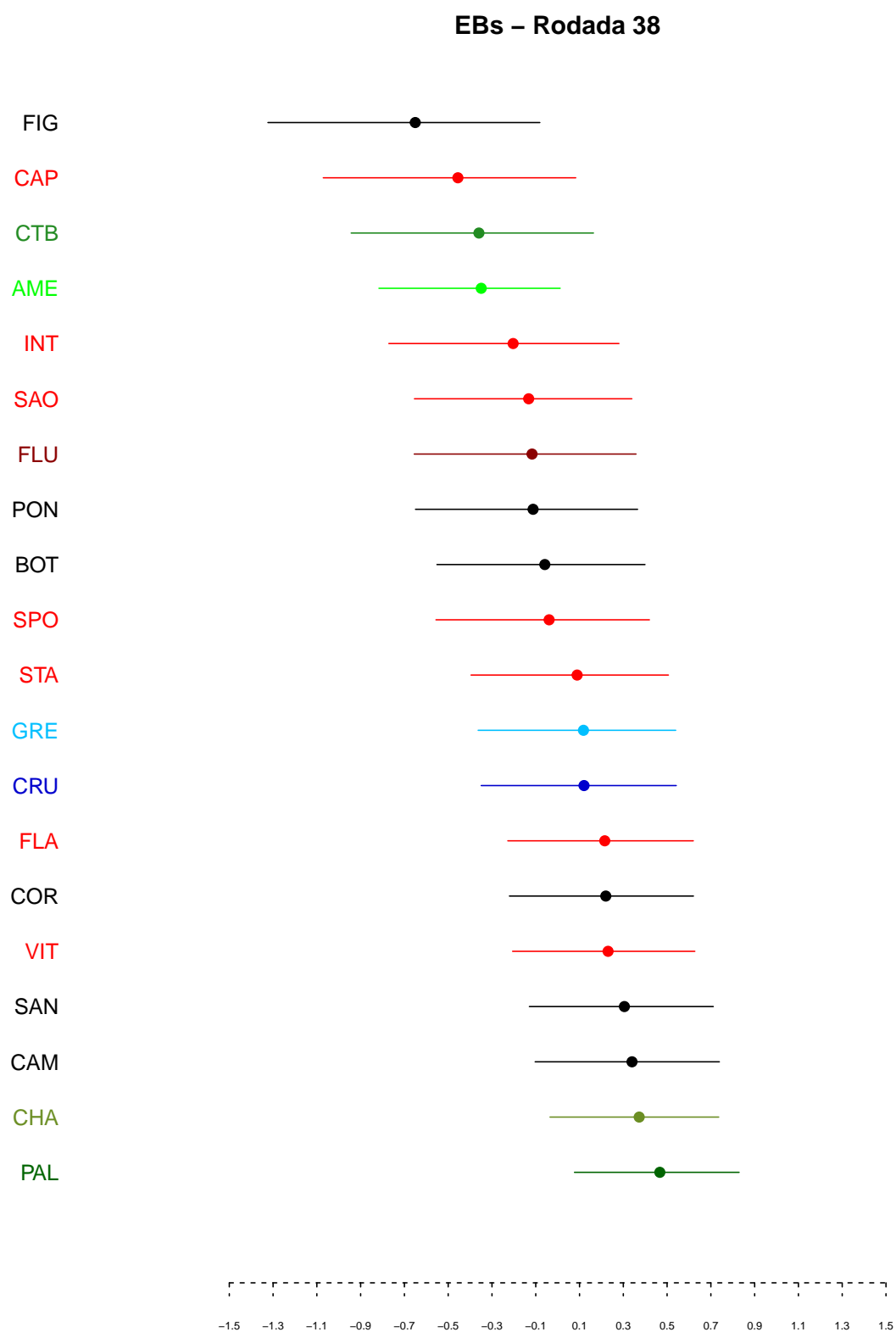


Figura A.4: Estimativas do fator ataque para o Campeonato brasileiro de 2016.

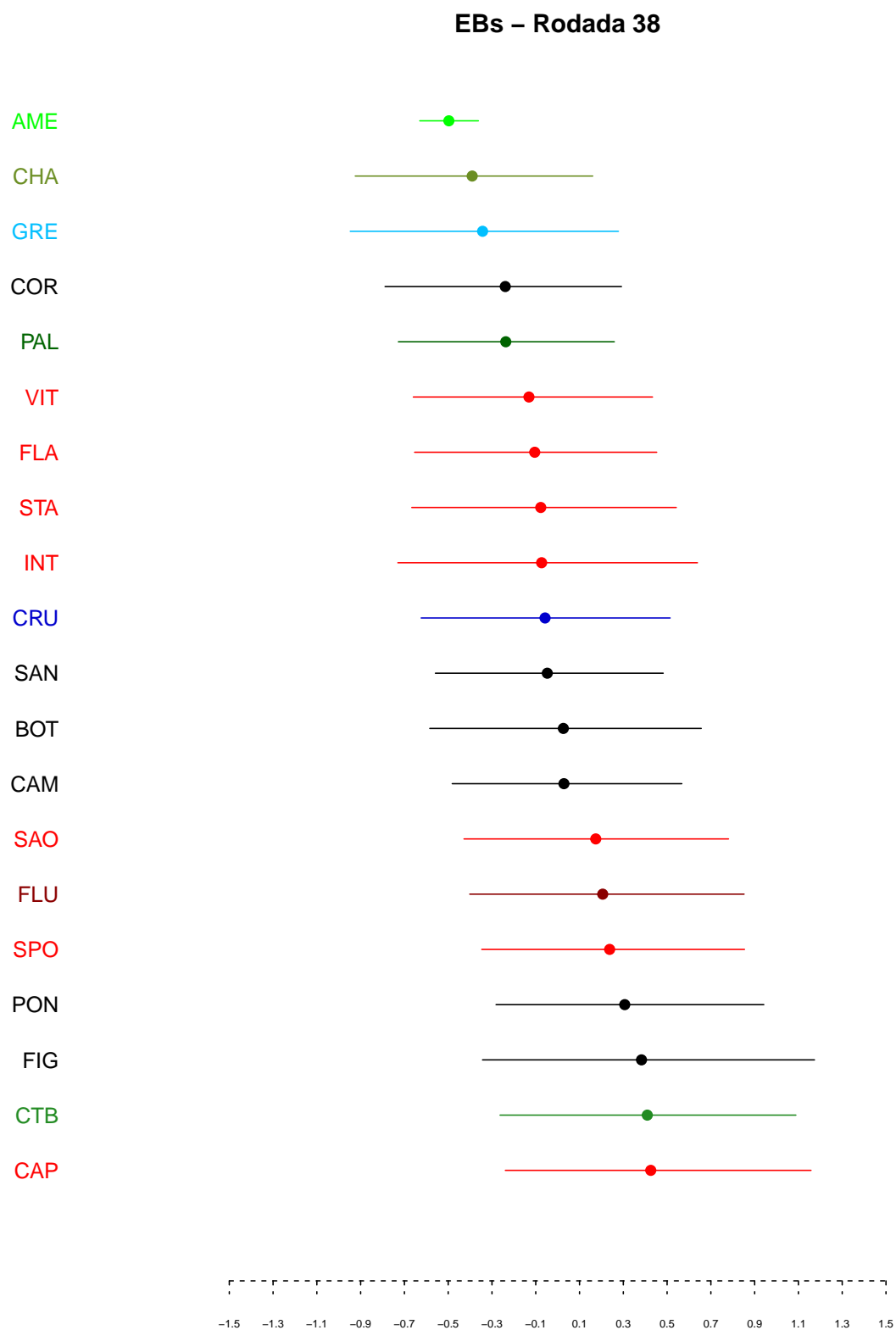


Figura A.5: Estimativas do fator campo para o Campeonato brasileiro de 2016.

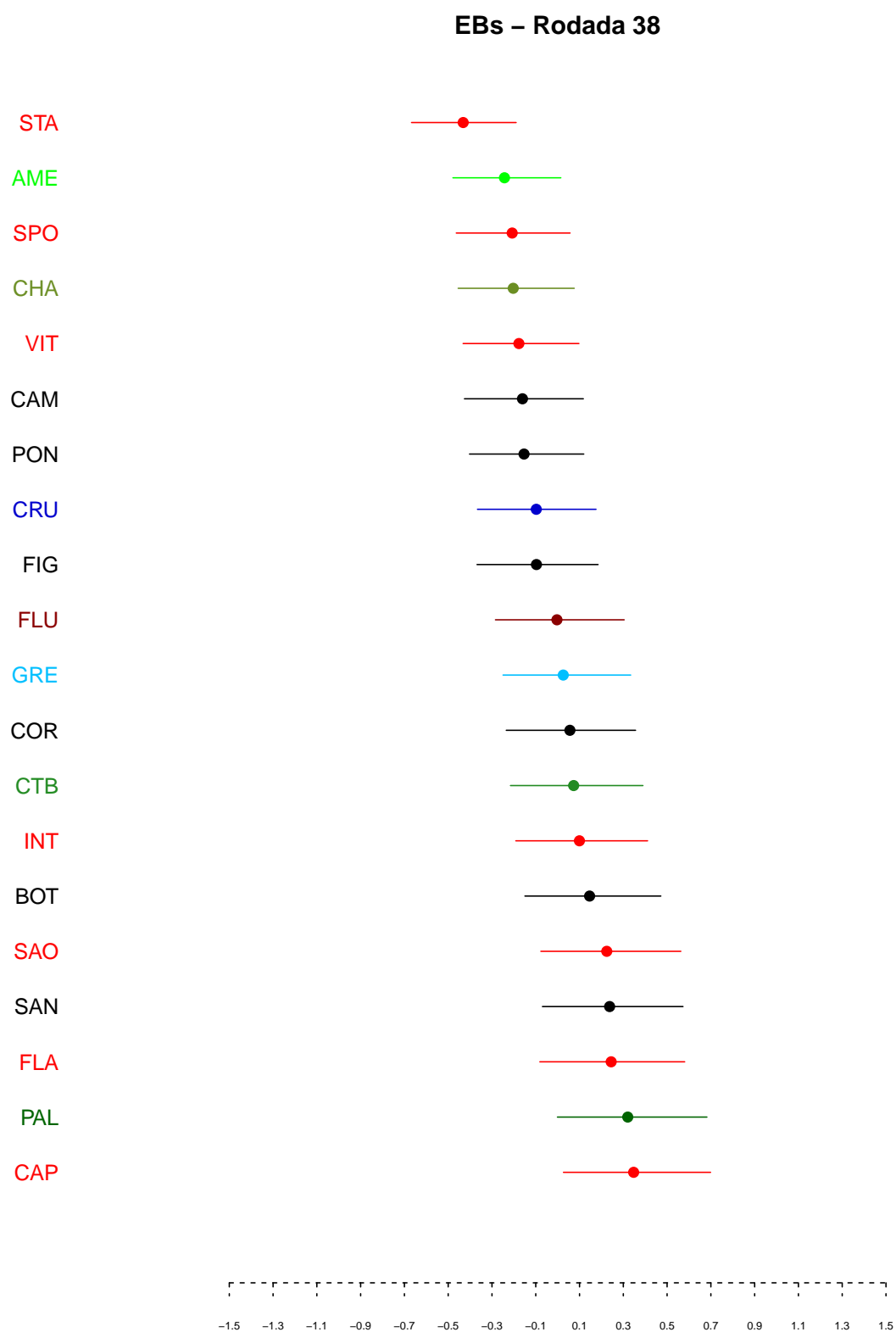


Figura A.6: Estimativas do fator defesa para o Campeonato brasileiro de 2016.

Observando-se as Figuras A.4 a A.6 e usando-se as siglas dadas na seção A.1, pode-se ver que o Atlético-PR tem ao mesmo tempo o melhor fator campo e o segundo pior fator ataque. Isso ocorreu porque o time marcou 27 de seus 38 gols em casa, enquanto sofreu apenas seis de seus 32 gols em seu próprio estádio. Levando em conta a forma como o modelo é definido, ele teve certa dificuldade de separar os efeitos dos fatores ataque e campo, por causa da diferença entre as campanhas do Atlético-PR em casa e fora de casa. Por isso, ele acabou considerando que o fator campo alicerçou toda a temporada do time.

Algo semelhante ocorreu com o Figueirense, que marcou 21 de seus 30 gols e sofreu 16 de seus 50 gols em casa, e com o Coritiba, que marcou 29 dos seus 41 gols e sofreu 17 dos seus 42 gols em casa. Dessa forma, os três times criaram dificuldades para o processo de estimação do modelo. Assim, eles talvez tenham seus ataques subestimados, com base na Figura A.4, e seus fatores campo superestimados, na Figura A.5.

O eventual campeão da competição, Palmeiras, e o eventual terceiro colocado, Flamengo, talvez tenham sofrido do problema oposto, de mandos de campo fracos. Suas campanhas foram muito equilibradas, sendo os únicos dois times a ter saldo de gol positivo (mais gols marcados do que sofridos), quando jogando fora de casa. Além disso, seus saldos de gol dentro de casa não foram tão impressionantes, principalmente o do Flamengo. Dessa forma, ambos os times apresentaram estimativas altas para seus fatores de ataque e defesa, mas baixas para seu fator campo.

É interessante notar, também, que o Palmeiras pode ser considerado como merecedor de seu título. A equipe teve o melhor ataque e a segunda melhor defesa da competição, apesar do quinto pior mando de campo. Seus competidores mais próximos, Flamengo e Santos, não se aproximaram muito da qualidade do adversário. O Flamengo teve apenas o sétimo melhor ataque, combinado com o sétimo pior mando de campo e a terceira melhor defesa. O Santos, por outro lado, teve o quarto melhor ataque, a quarta melhor defesa e o décimo melhor fator de campo.

Uma análise semelhante dos dois últimos colocados da competição, América-MG e Santa Cruz também leva a conclusões interessantes. O América teve o quarto pior ataque, o pior mando de campo e a segunda pior defesa da competição. Já o Santa Cruz teve o décimo melhor ataque, o oitavo pior mando de campo e a pior defesa do campeonato. Fica claro que o América realmente era um time fraco, nessa competição, enquanto o Santa Cruz foi rebaixado por causa de sua fraca defesa.

A.3.2 Análises e resultados do Modelo dinâmico

Nessa subseção, o modelo avaliado é o dinâmico, enquanto o conjunto de dados continua sendo o do Campeonato brasileiro de 2016. No caso do modelo dinâmico, as estimativas apresentadas continuam sendo EBs, mas foram calculadas estimativas para cada time a cada ponto de atualização dos valores dos parâmetros do modelo. As Figuras A.7 a A.21 mostram a coleção dessas estimativas, para o conjunto de dados.

As Figuras A.7 a A.11 contêm as estimativas pontuais e intervalares para os fatores de ataque das equipes do campeonato, estimadas a cada rodada. Já as Figuras A.12 a A.16 contêm as estimativas para os fatores campo, enquanto as estimativas para os fatores de defesa estão nas Figuras A.17 a A.21.

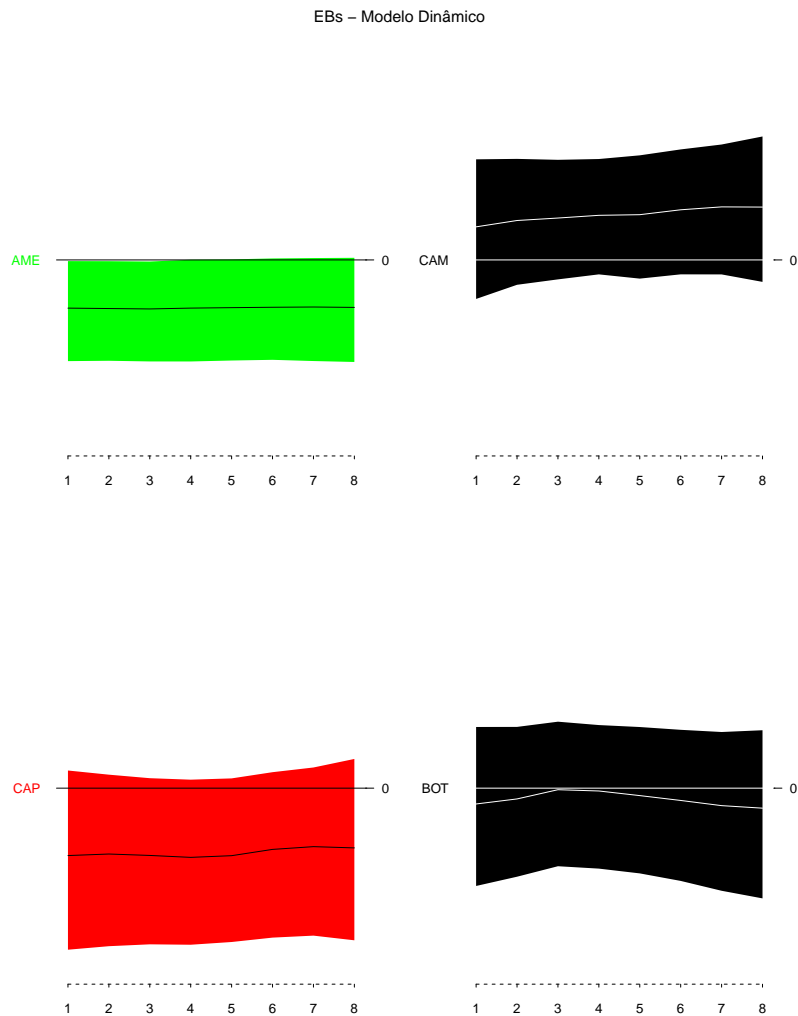


Figura A.7: Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 1.

EBs – Modelo Dinâmico

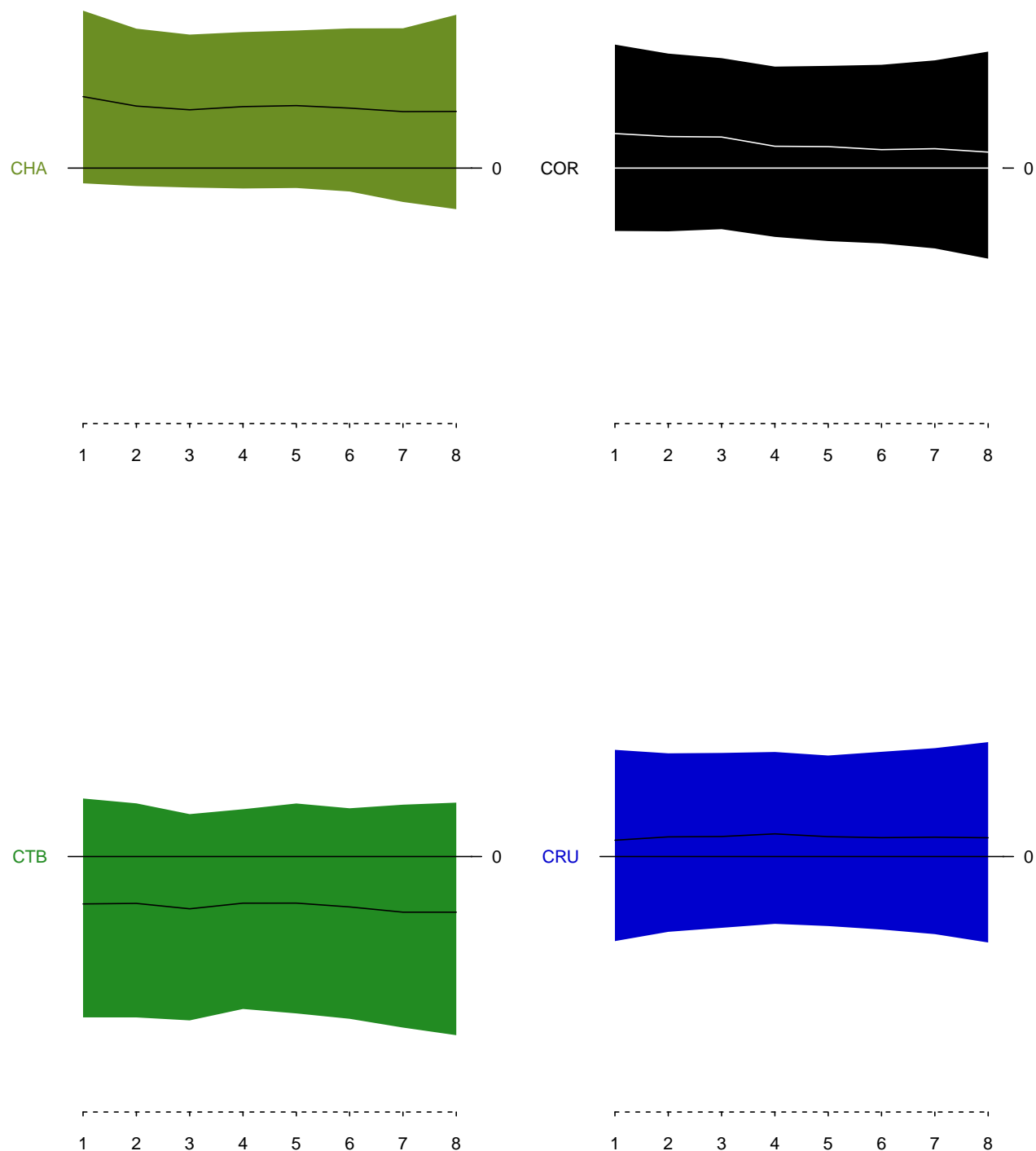


Figura A.8: Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 2.

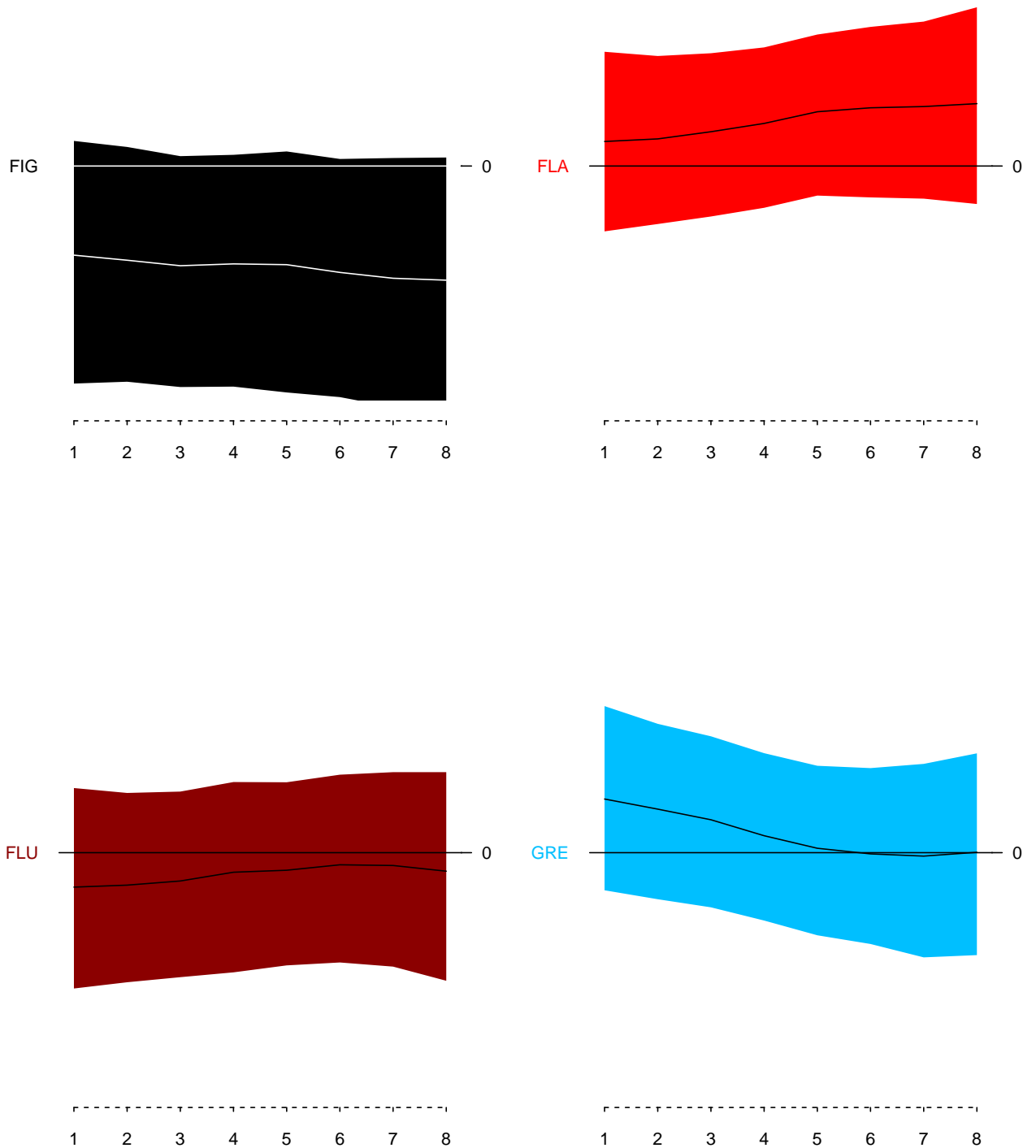


Figura A.9: Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 3.

EBs – Modelo Dinâmico

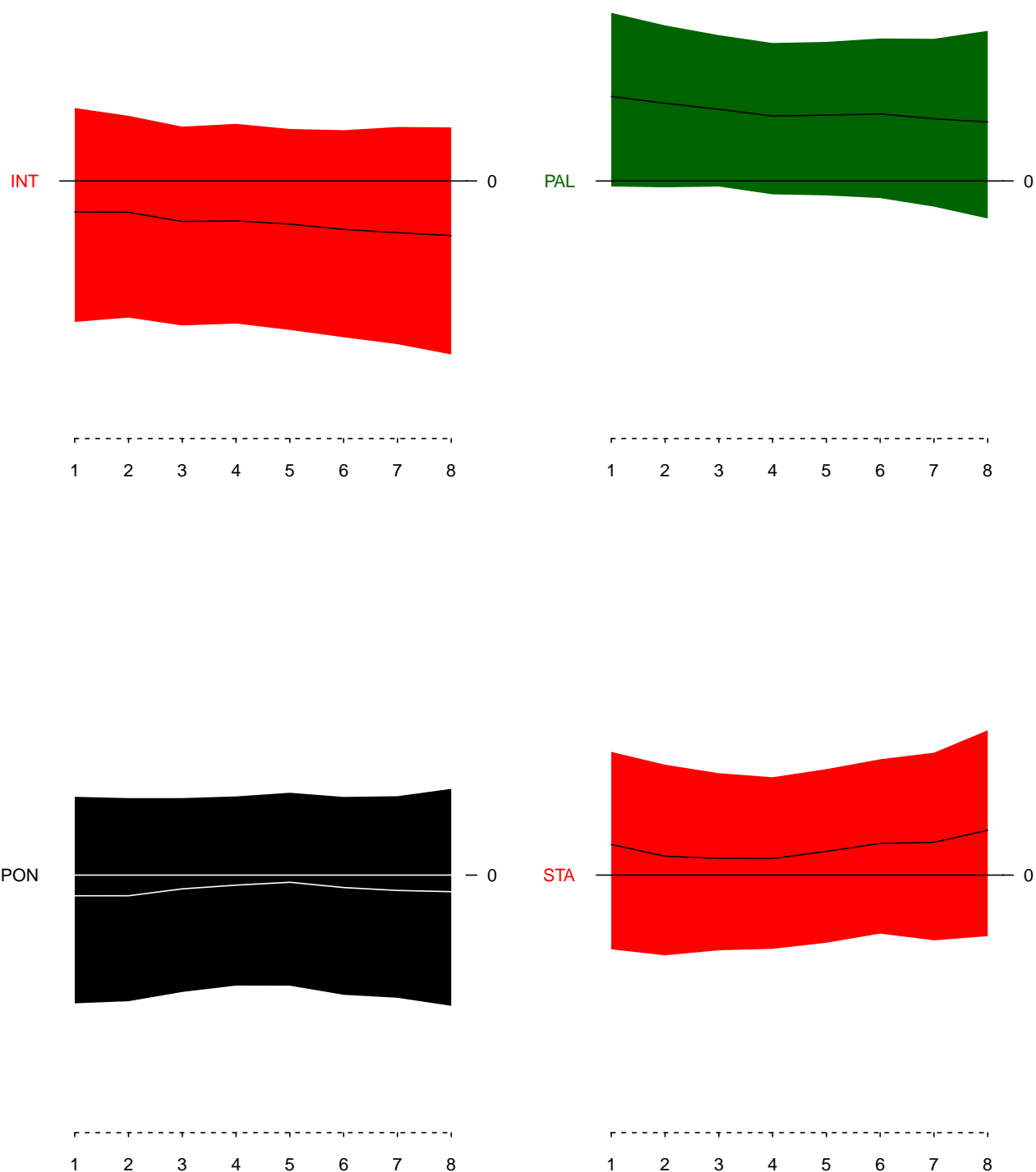


Figura A.10: Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 4.

EBs – Modelo Dinâmico

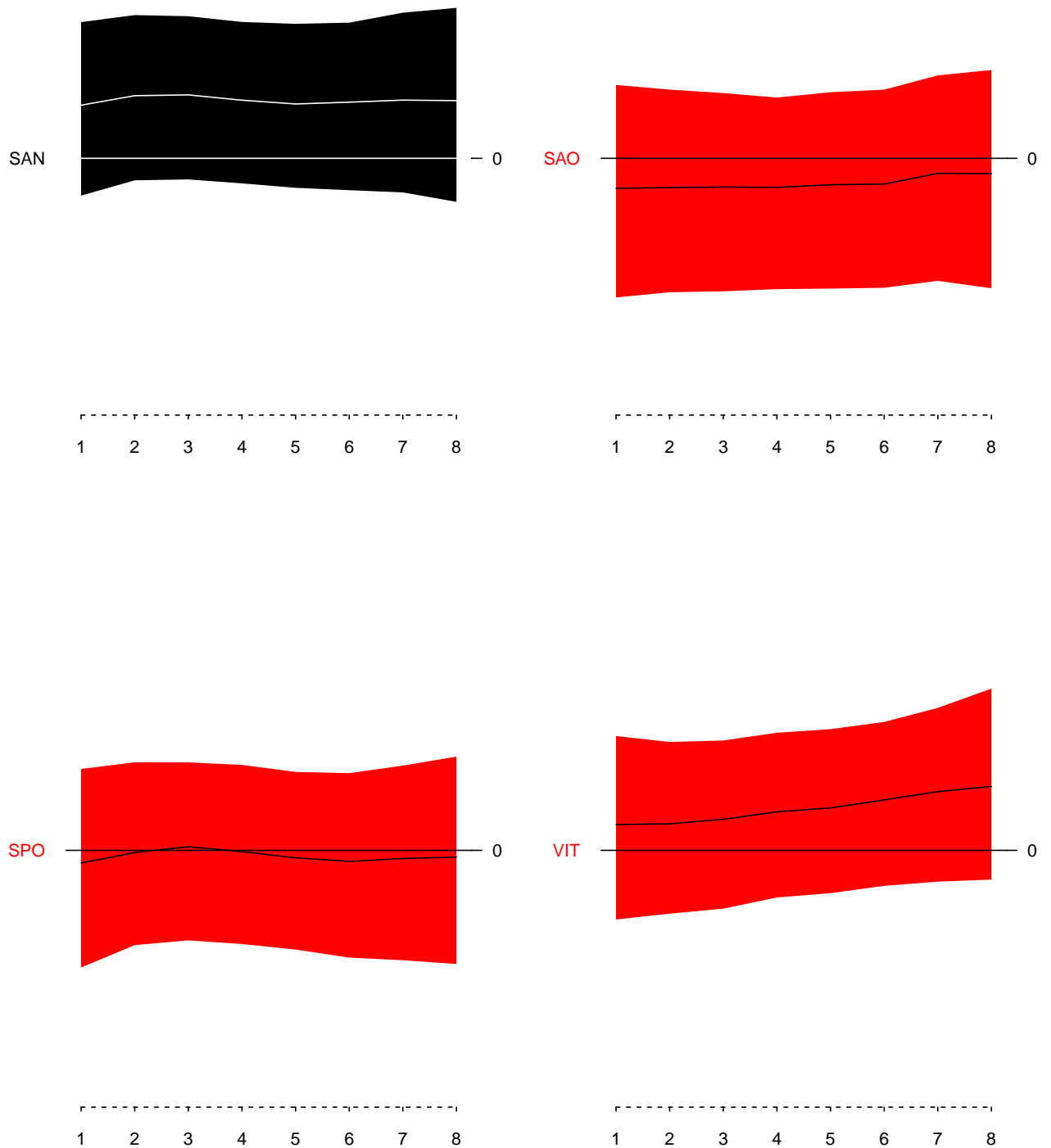


Figura A.11: Estimativas do fator ataque para o Campeonato brasileiro de 2016 - Parte 5.

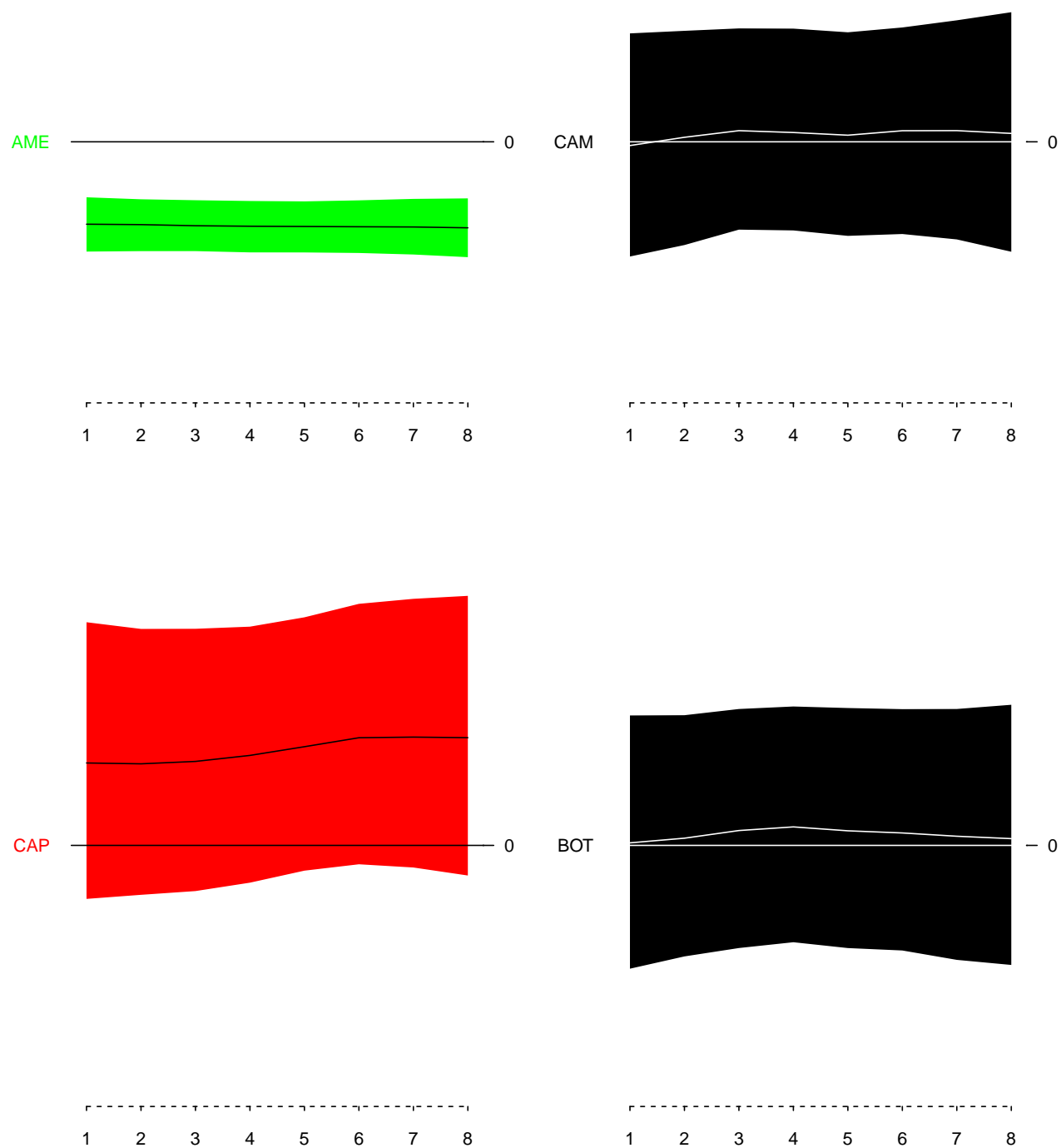


Figura A.12: Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 1.

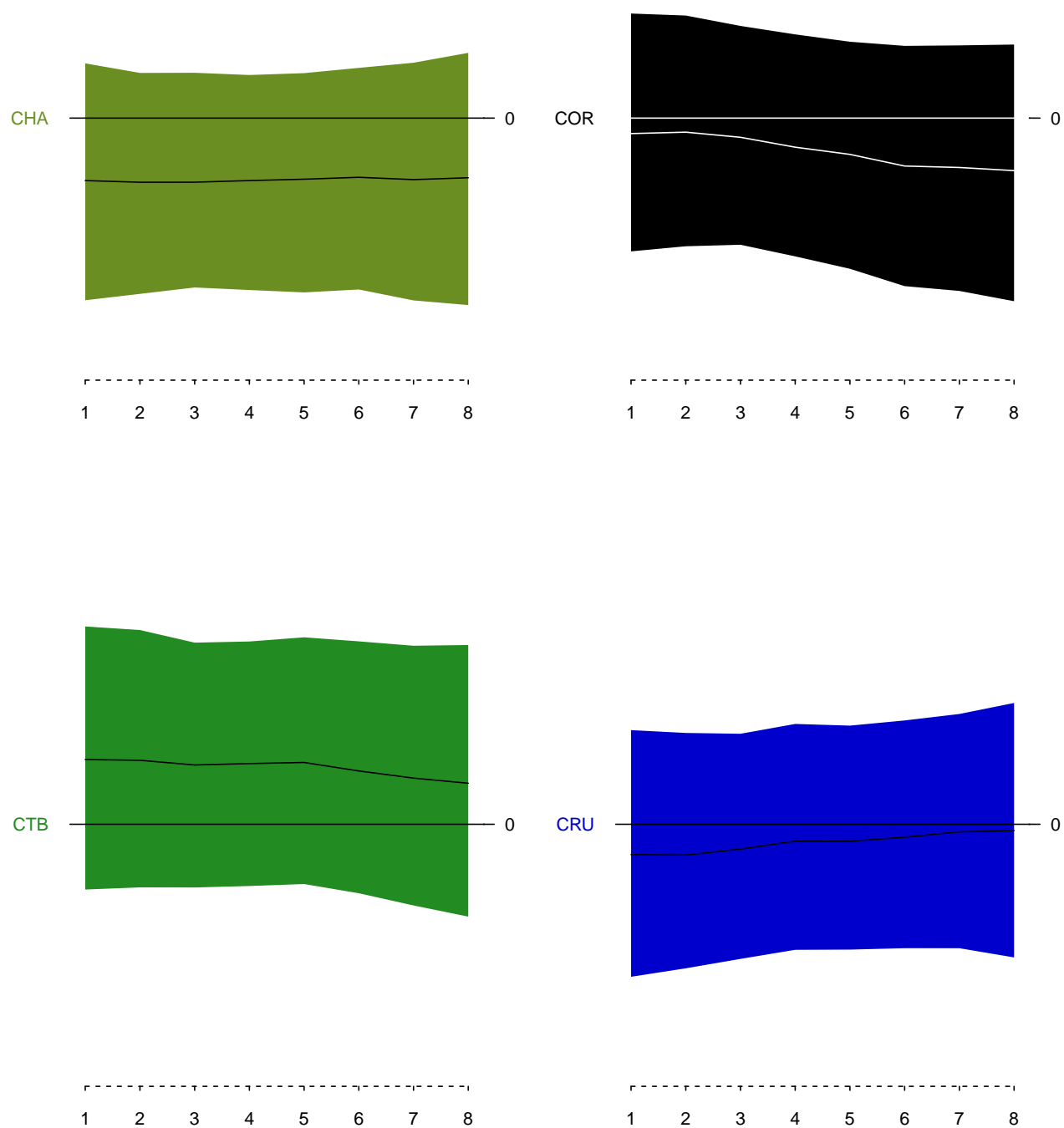


Figura A.13: Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 2.

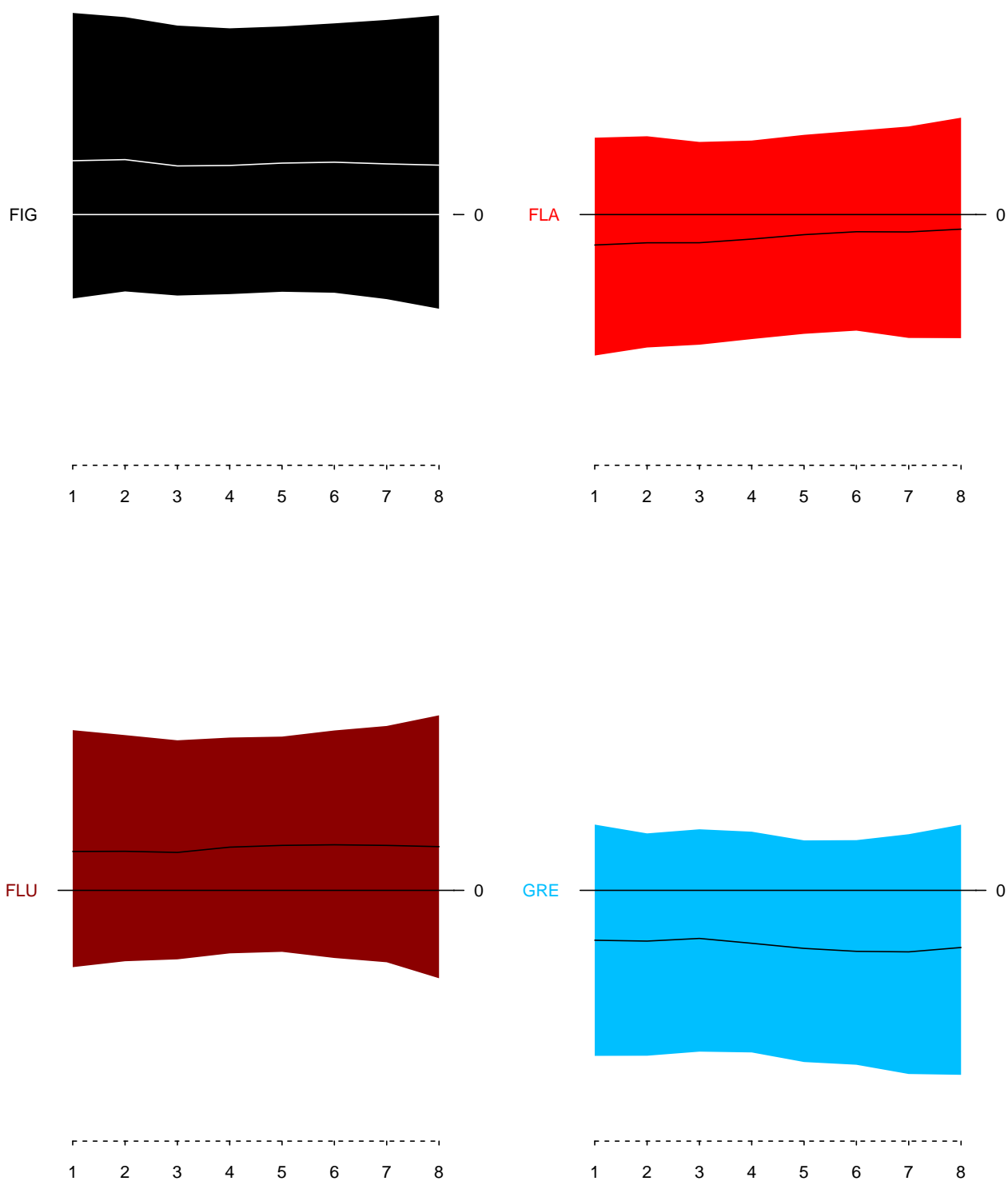


Figura A.14: Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 3.

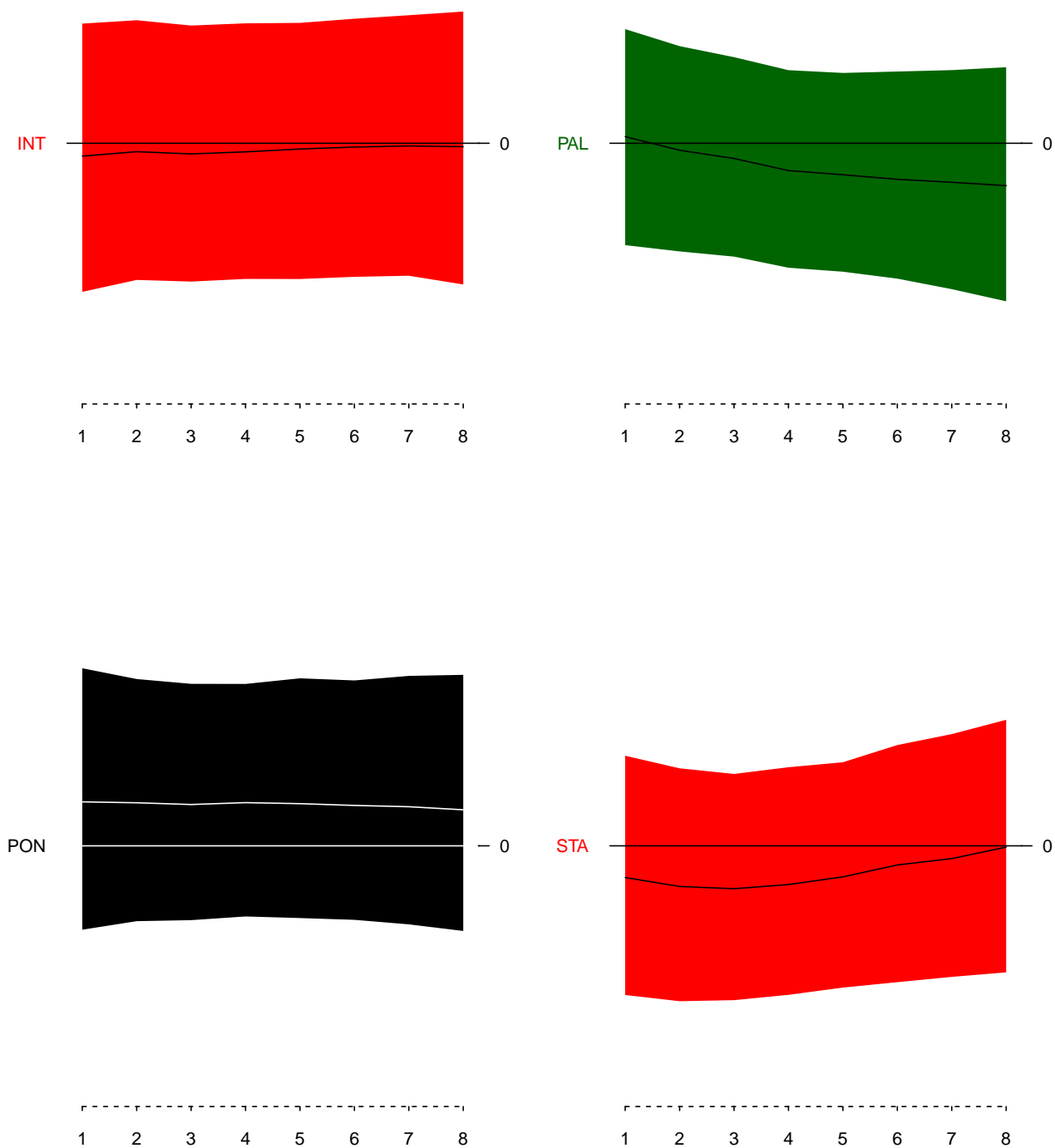


Figura A.15: Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 4.

EBs – Modelo Dinâmico

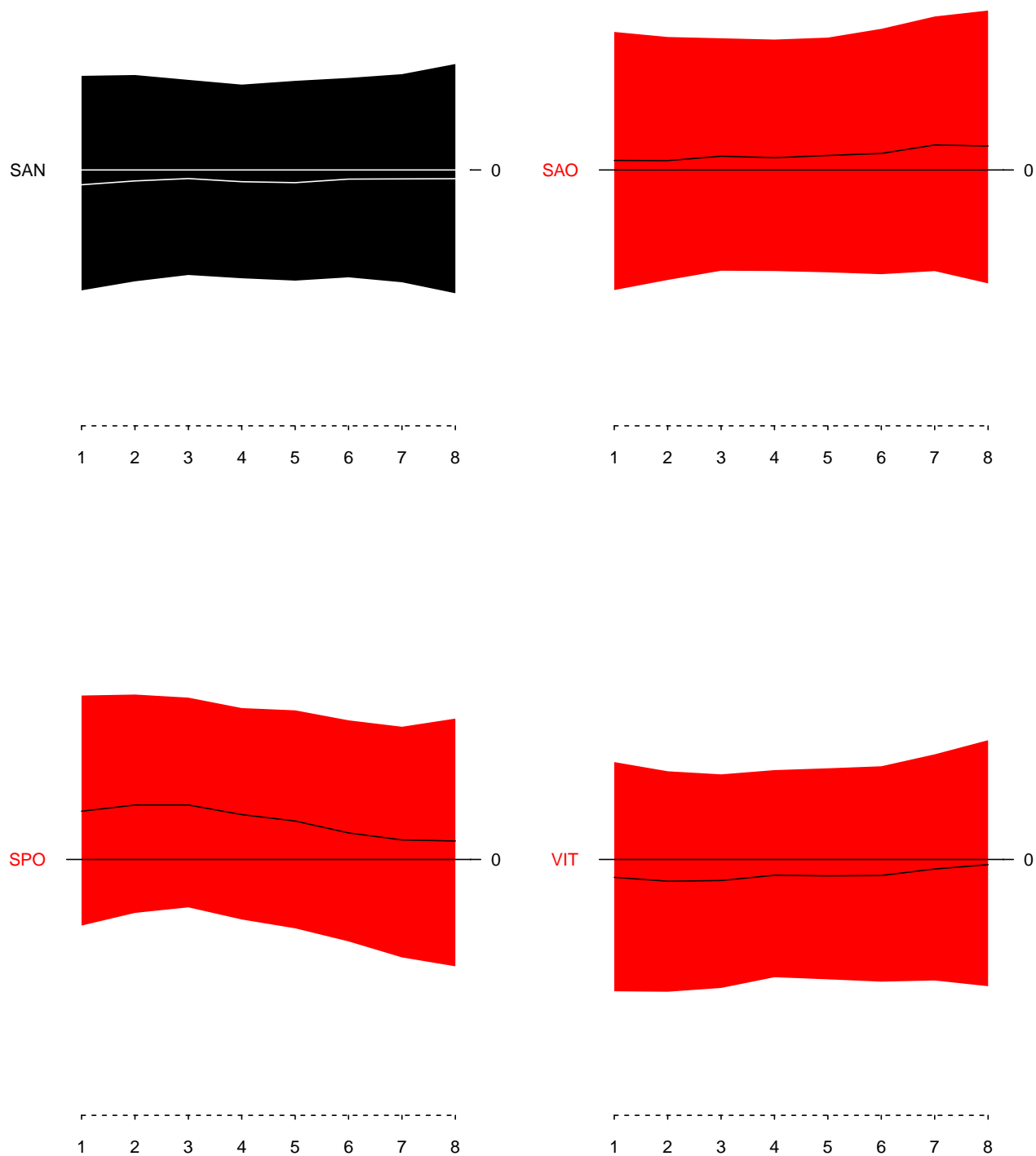


Figura A.16: Estimativas do fator campo para o Campeonato brasileiro de 2016 - Parte 5.

EBs – Modelo Dinâmico

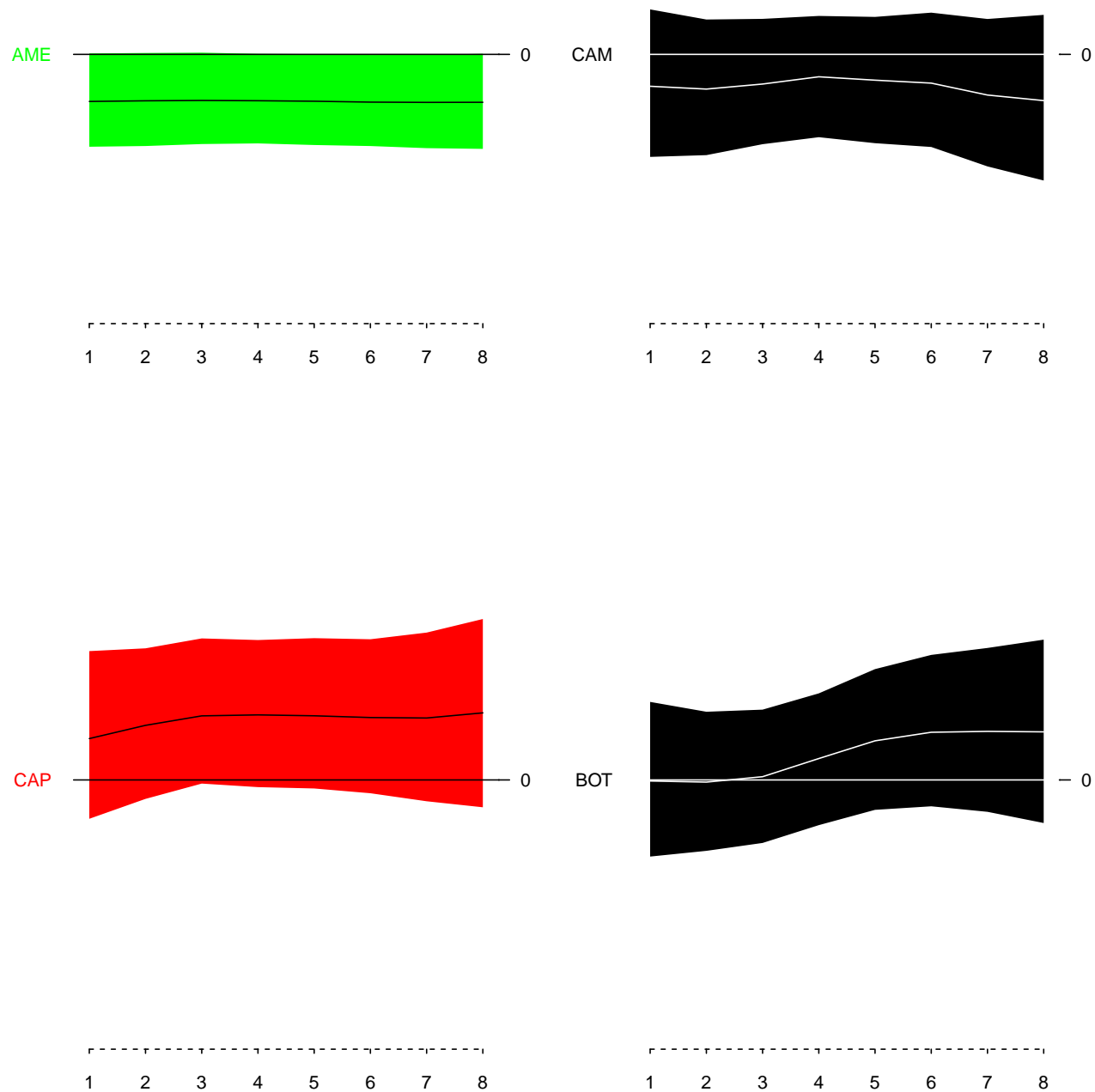


Figura A.17: Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 1.

EBs – Modelo Dinâmico

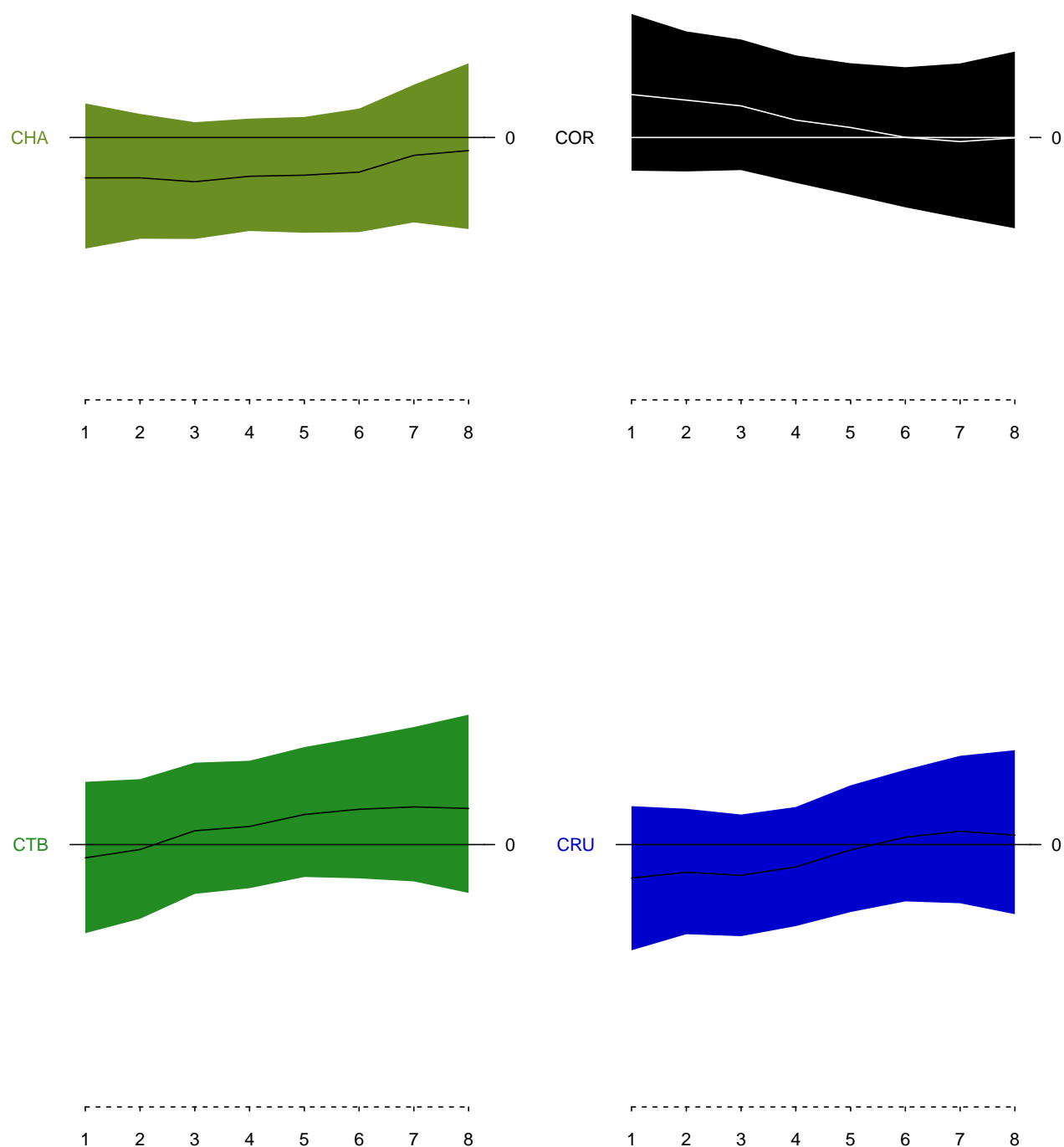


Figura A.18: Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 2.

EBs – Modelo Dinâmico

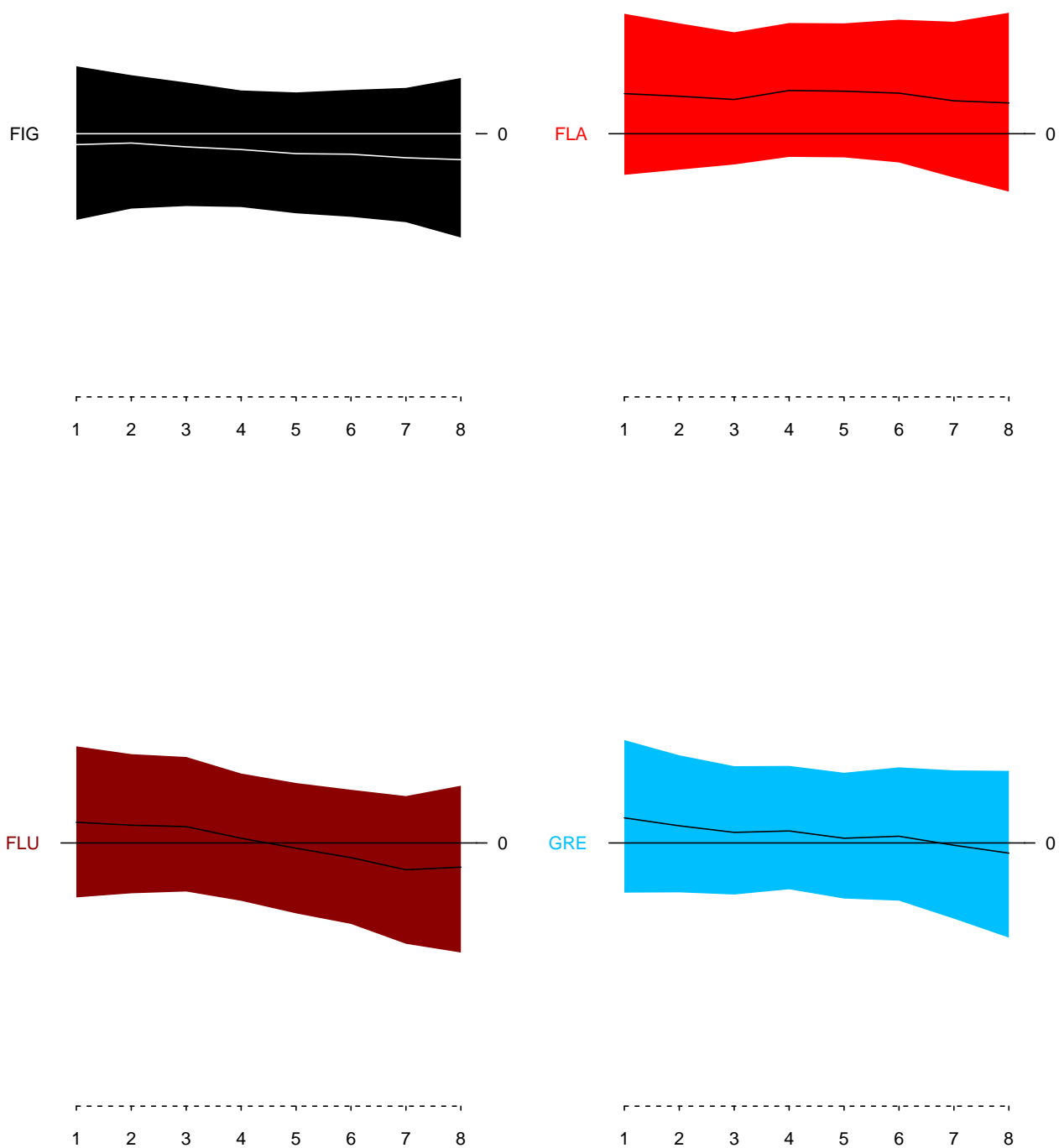


Figura A.19: Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 3.

EBs – Modelo Dinâmico

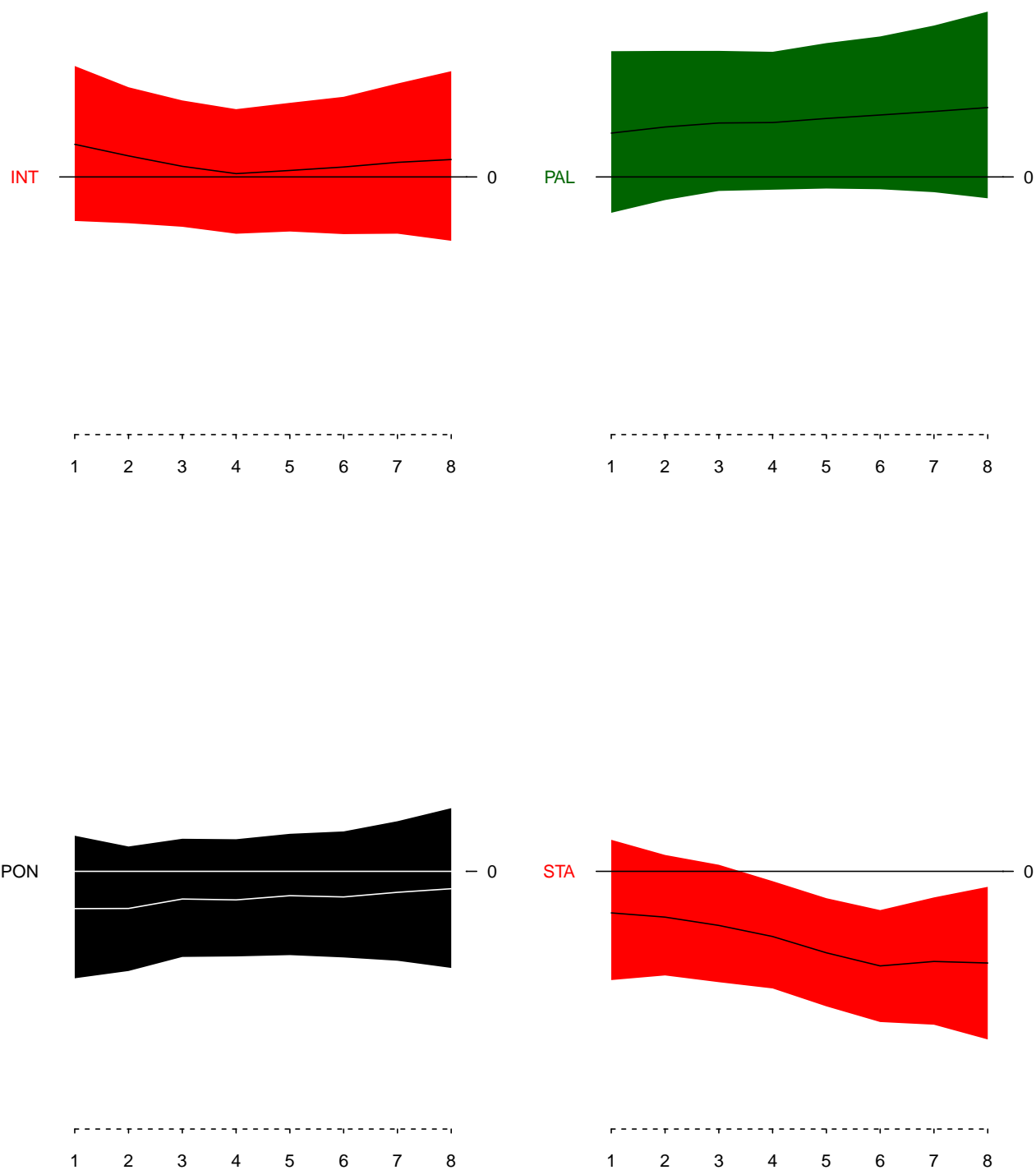


Figura A.20: Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 4.

EBs – Modelo Dinâmico

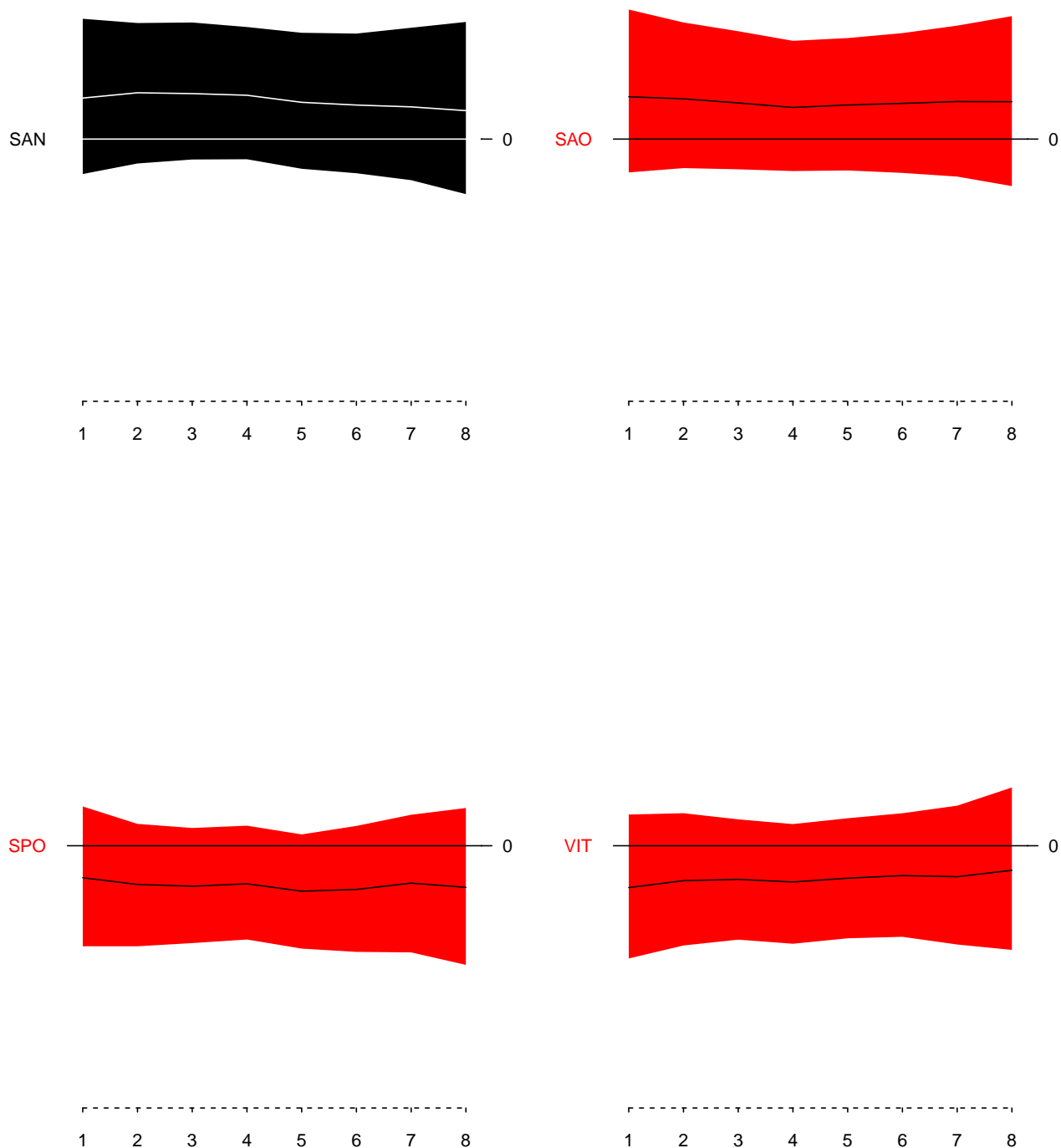


Figura A.21: Estimativas do fator defesa para o Campeonato brasileiro de 2016 - Parte 5.

De forma compatível com o que foi escrito no texto, não se vê grande movimento temporal na maior parte dos parâmetros do modelo. No entanto, há algumas mudanças interessantes, para três times, todas em seus fatores de ataque. Flamengo, Internacional e Vitória apresentaram fatores campo e de defesa que se mantiveram razoavelmente estáveis, ao longo do torneio. Por outro lado, os três tiveram mudanças em seus fatores de ataque que acabaram sendo importantes para suas campanhas.

Para o Flamengo, a melhora de seu ataque levou a uma série de vitórias da equipe, que chegou a se aproximar do primeiro colocado, Palmeiras. Entretanto, essa melhora não foi suficiente, e o Palmeiras acabou sendo o campeão. Por outro lado, as mudanças de Vitória e Internacional foram fundamentais. Os ataques de ambas as equipes foram em direções diferentes, nas últimas rodadas do campeonato, o do Internacional piorando e o do Vitória melhorando. Como discutido anteriormente, isso foi suficiente para permitir que o Vitória não fosse rebaixado, com o Internacional indo em seu lugar para a segunda divisão.

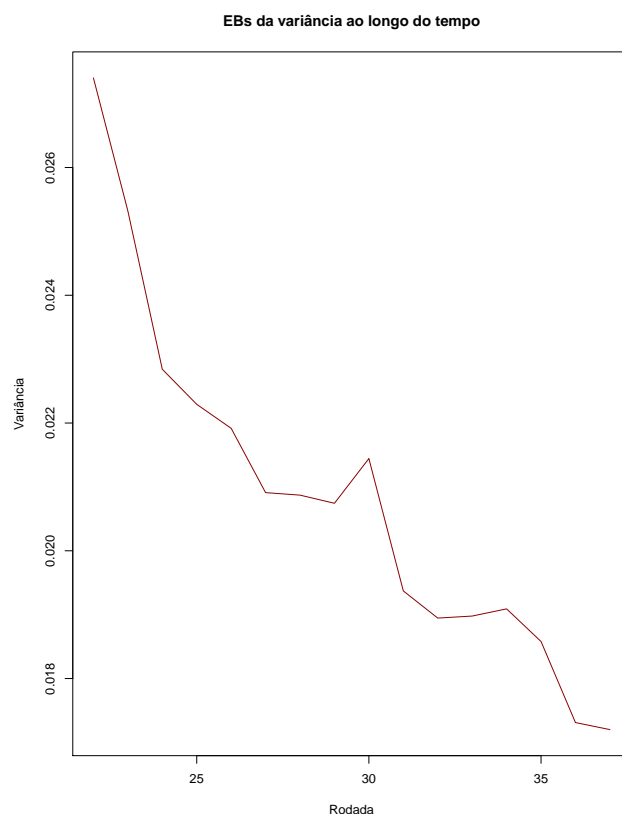


Figura A.22: Estimativas da variância das perturbações para o Campeonato brasileiro de 2016.

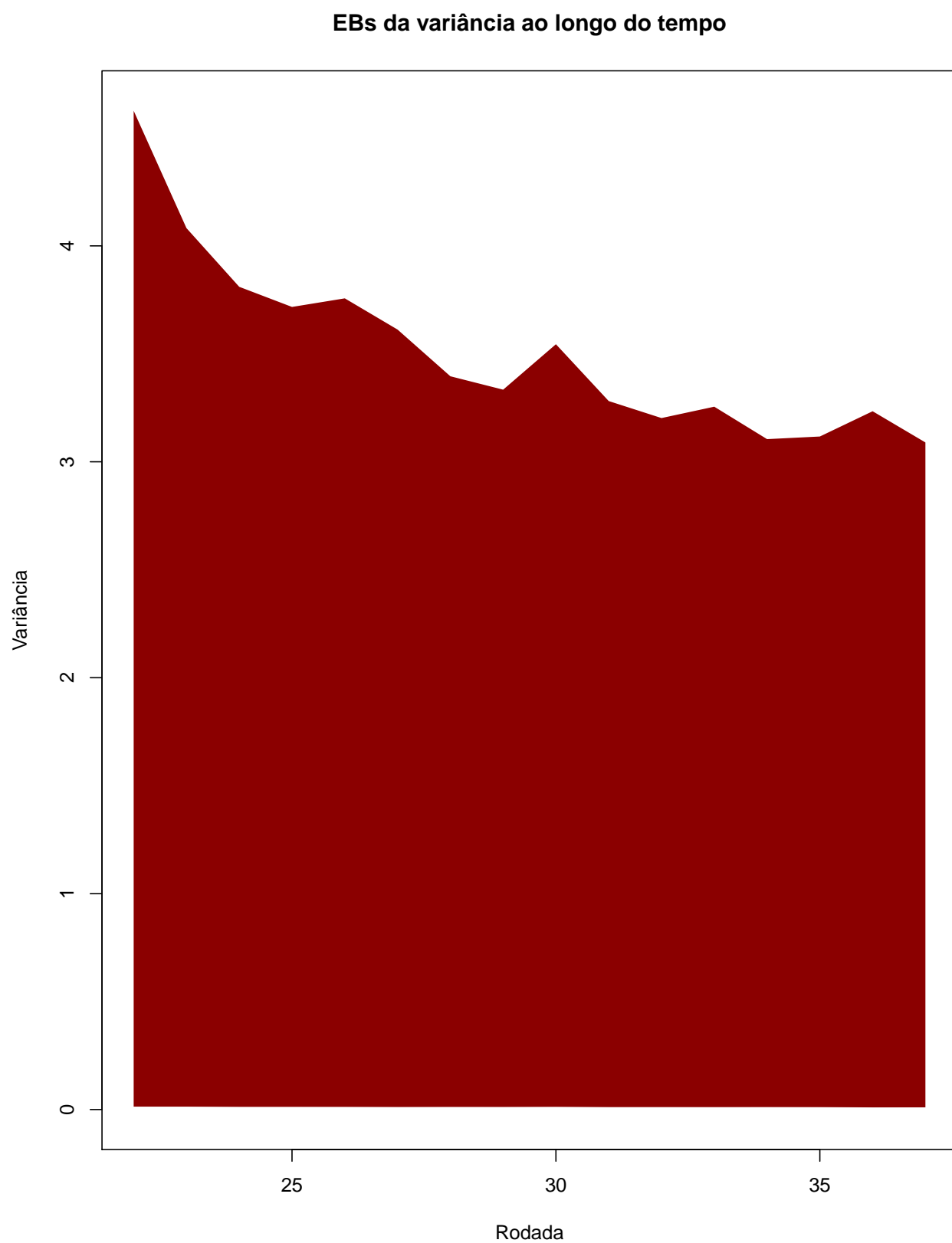


Figura A.23: Estimativas da variância das perturbações para o Campeonato brasileiro de 2016.

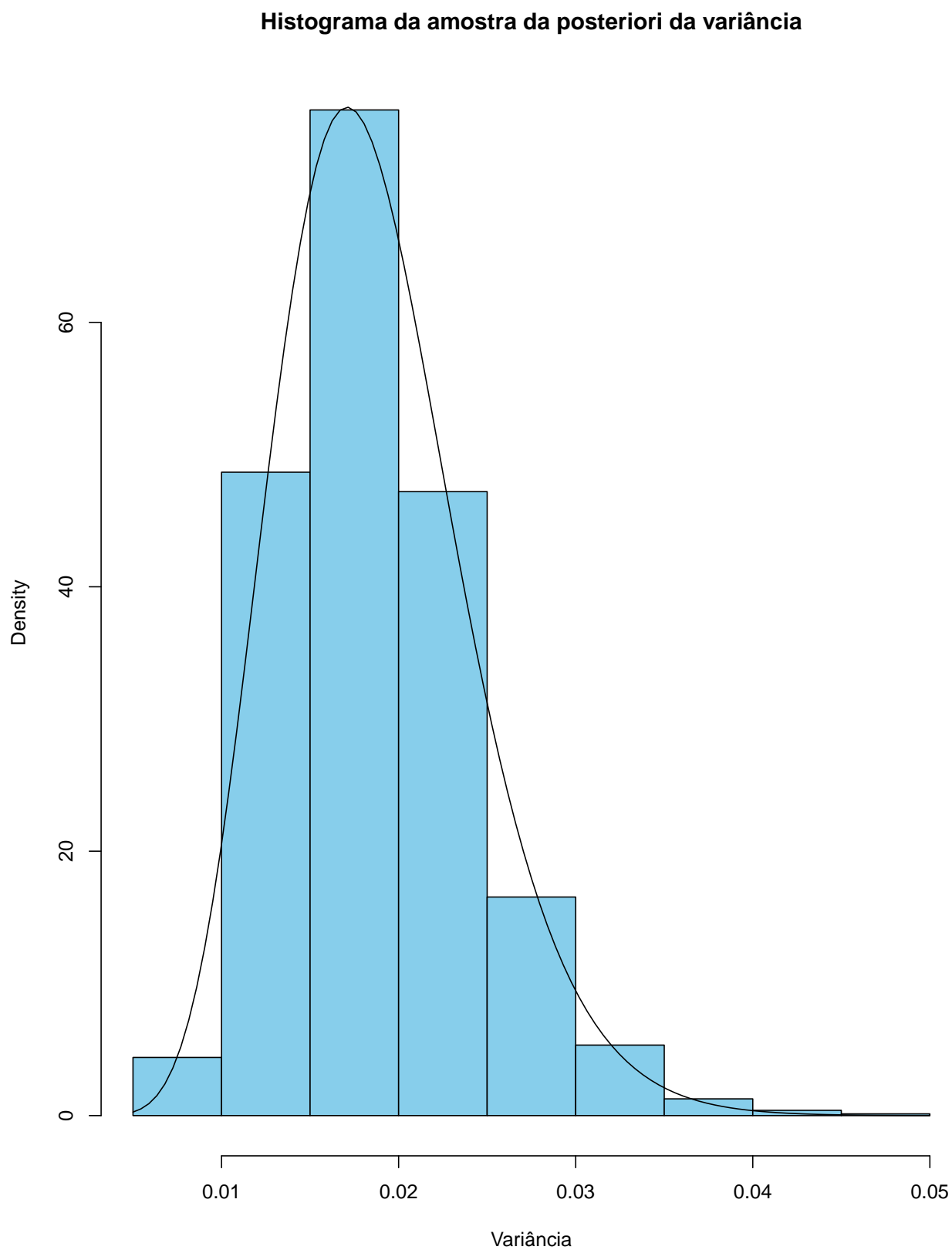


Figura A.24: Estimativas da variância das perturbações para o Campeonato inglês brasileiro de 2016.

Pode-se falar também das estimativas para a variância das perturbações normais. Como visto em (4.24), a priori usada no modelo dinâmico para o inverso dessa variância, σ_ω^2 , é uma distribuição gama. A Figura A.22 mostra as estimativas pontuais de Bayes para a variância ao longo das rodadas do campeonato, enquanto a Figura A.23 mostra as estimativas intervalares para a mesma. Finalmente, na Figura A.24 está um histograma da amostra aproximada da posteriori da variância. Sobre o histograma está representada a curva teórica da posteriori da variância, caso os parâmetros dessa distribuição fossem iguais às estimativas pontuais para os mesmos.

Ou seja, a curva apresentada na Figura A.24 é uma função de densidade que tem a forma apresentada em (A.1),

$$f(x|\alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, \quad (\text{A.1})$$

onde $f(x|\alpha, \beta)$ é a função de densidade de uma variável aleatória X que segue uma distribuição Gama com parâmetros α e β .

No caso da curva exibida na Figura A.24,

$$\beta = \hat{\beta} = \frac{E[\sigma_\omega^2|\underline{x}]}{\widehat{V[\sigma_\omega^2|\underline{x}]}} \quad e \quad \alpha = \hat{\alpha} = E[\sigma_\omega^2|\underline{x}]\hat{\beta}, \quad (\text{A.2})$$

onde a distribuição de $\sigma_\omega^2|\underline{x}$ é a distribuição a posteriori de σ_ω^2 com base nos dados do campeonato brasileiro de 2016, representados por \underline{x} nesse caso. Adicionalmente, $E[\sigma_\omega^2|\underline{x}]$ representa a esperança da posteriori da variância, enquanto $\widehat{V[\sigma_\omega^2|\underline{x}]}$ representa a estimativa da variância dessa posteriori. Essa estimativa é a variância amostral da amostra aproximada da posteriori de σ_ω^2 , como visto em (A.3),

$$\widehat{V[\sigma_\omega^2|\underline{x}]} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad (\text{A.3})$$

onde n é o tamanho da amostra, cada x_i representa um valor dela e \bar{x} representa sua média.

Uma análise conjunta das Figuras A.22 a A.24 parece apontar para a convergência das cadeias de Markov, no processo de MCMC.

A.3.3 Verossimilhanças preditivas para os modelos

Nessa subseção serão mostradas representações gráficas das verossimilhanças preditivas de todos os modelos avaliados, ao longo dos campeonatos, além de tabelas com os valores. Todos os gráficos apresentam os valores das verossimilhanças preditivas de cada modelo, junto com o jogo ou rodada referente ao valor, enquanto cada tabela apresenta os valores para cada modelo, a cada rodada avaliada.

As Figuras A.25 a A.28 contêm os valores das verossimilhanças preditivas padronizadas do modelo estático e dos modelos usados para comparação neste trabalho. Especificamente, as Figuras A.26 e A.28 contêm as verossimilhanças preditivas padronizadas do modelo estático. A Figura A.26 apresenta as verossimilhanças preditivas para cada jogo do Campeonato brasileiro de 2016, enquanto a Figura A.28 apresenta as verossimilhanças do modelo por rodada.

As Figuras referentes ao modelo dinâmico são as Figuras A.29 a A.32. A única diferença dessas Figuras para as do modelo estático é que o modelo dinâmico passa a ser representado nelas. Elas foram geradas com o propósito de permitir comparações entre os modelos estático e dinâmico. Por isso, só foram coletadas previsões para o modelos dinâmico a partir das rodadas em que se considerou que havia ocorrido a estabilização das previsões do modelo estático.

Quando se compara as Figuras A.26 e A.30, não se vê grandes diferenças qualitativas entre as previsões dos modelos estático e dinâmico, por jogo do Campeonato brasileiro de 2016. O mesmo é visto, agora para as previsões por rodada, quando se compara as Figuras A.28 e A.32.

Uma análise qualitativa das figuras apresentadas combinada com uma análise das Tabelas A.1 e ??, aponta fortemente para uma grande mudança qualitativa nas verossimilhanças preditivas do modelo estático, por volta da rodada 23. Após esse ponto de corte, o modelo estático deixa de ser muito pior que os modelos nulo, simples e Chance de Gol e passa a ter as melhores previsões do grupo.

As previsões do modelo estático começaram a ser coletadas apenas a partir da sétima rodada, pois antes disso algumas cadeias de Markov não convergiram, durante o uso de algoritmos MCMC, e as do dinâmico só começaram a ser coletadas a partir da rodada 23 por causa dos diversos possíveis modelos dinâmicos que estavam sendo comparados, como apresentado na seção A.2, o que tornava muito custoso computacionalmente a coleta de previsões de cada modelo dinâmico para um conjunto de 32 rodadas.

Interessantemente, esses pontos arbitrários de corte, rodadas 7 e 23, dividiram o

conjunto de previsões do modelo estático coletadas em metade. E essa divisão mostra a mudança qualitativa de grande magnitude que ocorreu nas previsões da segunda metade do conjunto, como mostra a Tabela A.2, na qual os valores das verossimilhanças preditivas do modelo estático para a primeira metade são muito piores que as do modelo nulo - o que apontaria, para um estatístico, a necessidade de descartar o modelo estático - enquanto as da segunda metade são muito boas, as melhores de todos os modelos comparados, com as do modelo dinâmico tendo qualidade muito semelhante.

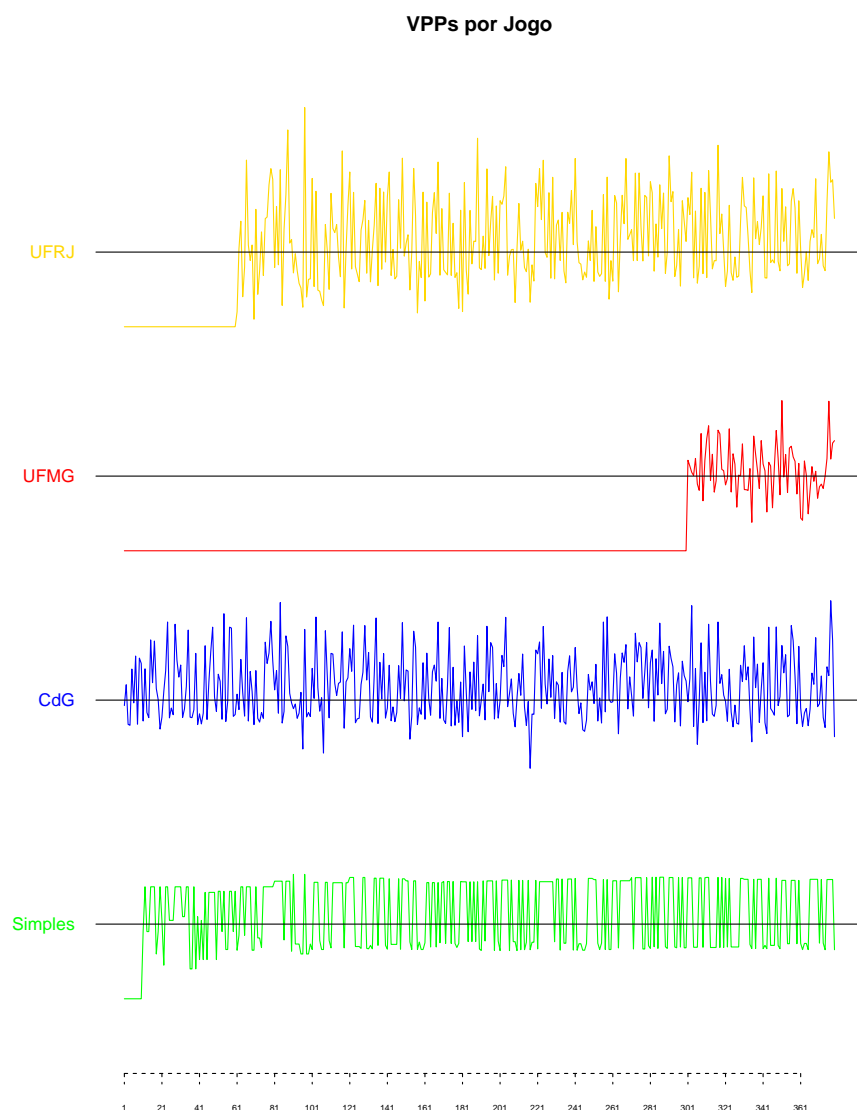


Figura A.25: Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2016.

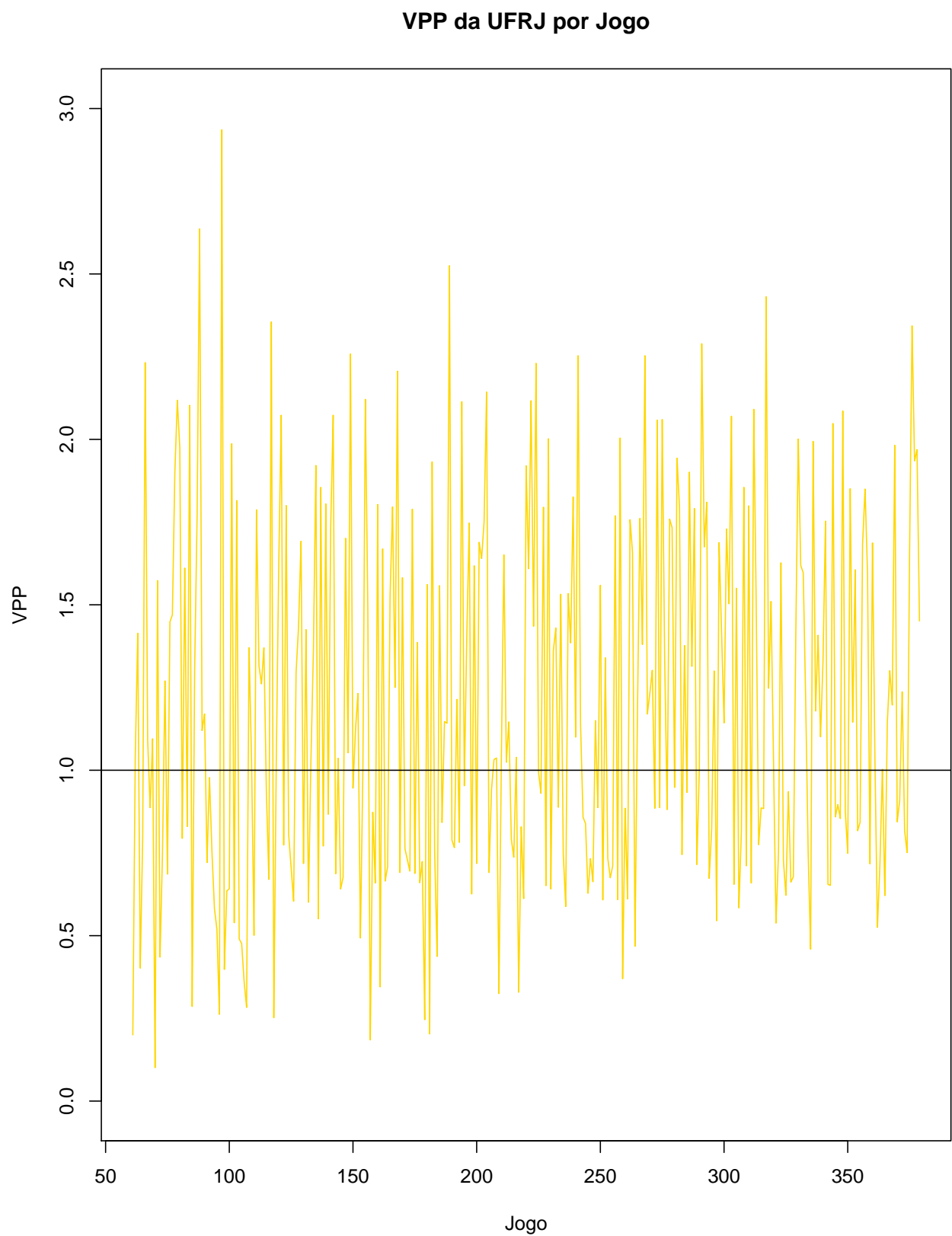


Figura A.26: Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2016.

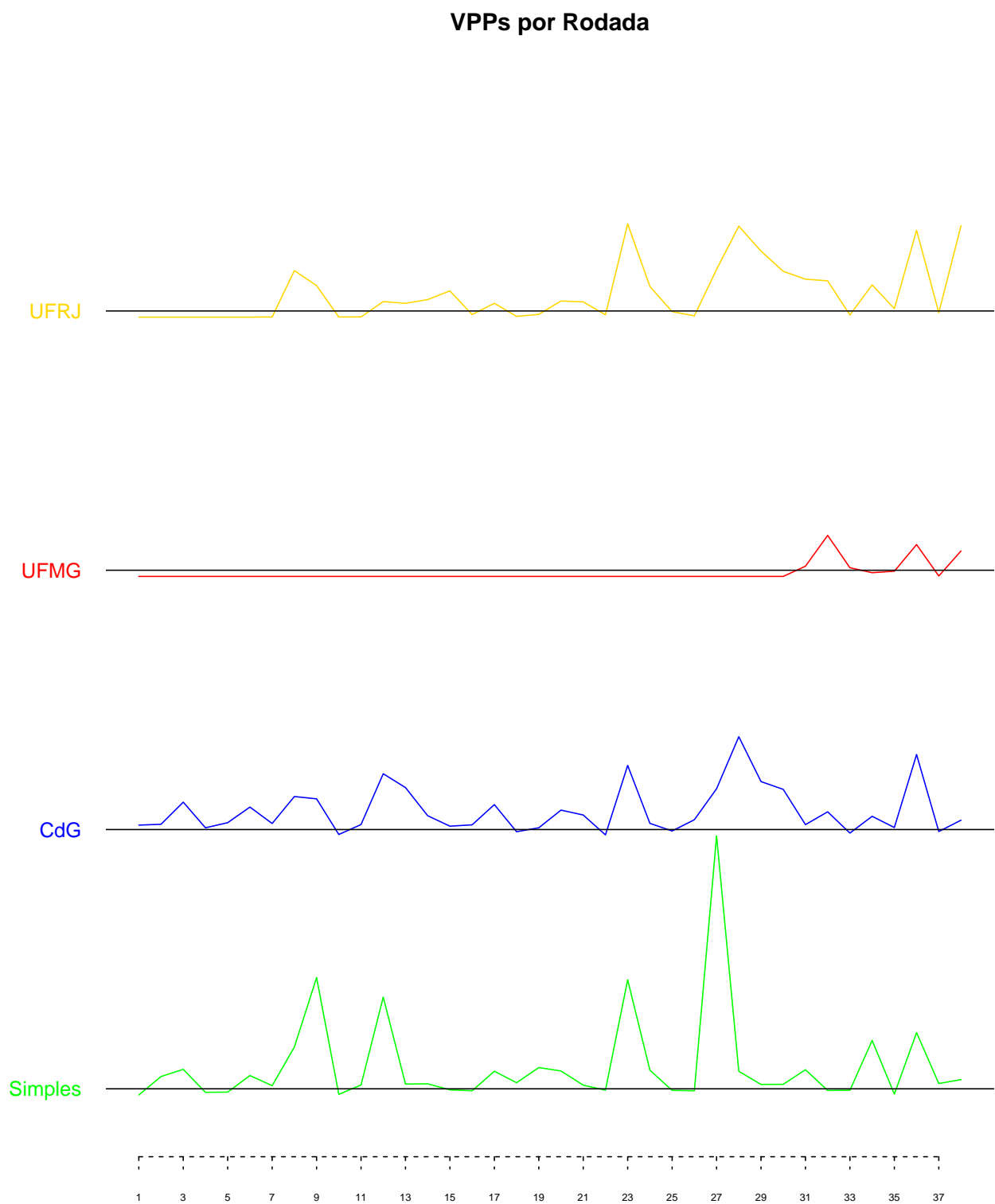


Figura A.27: Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2016.

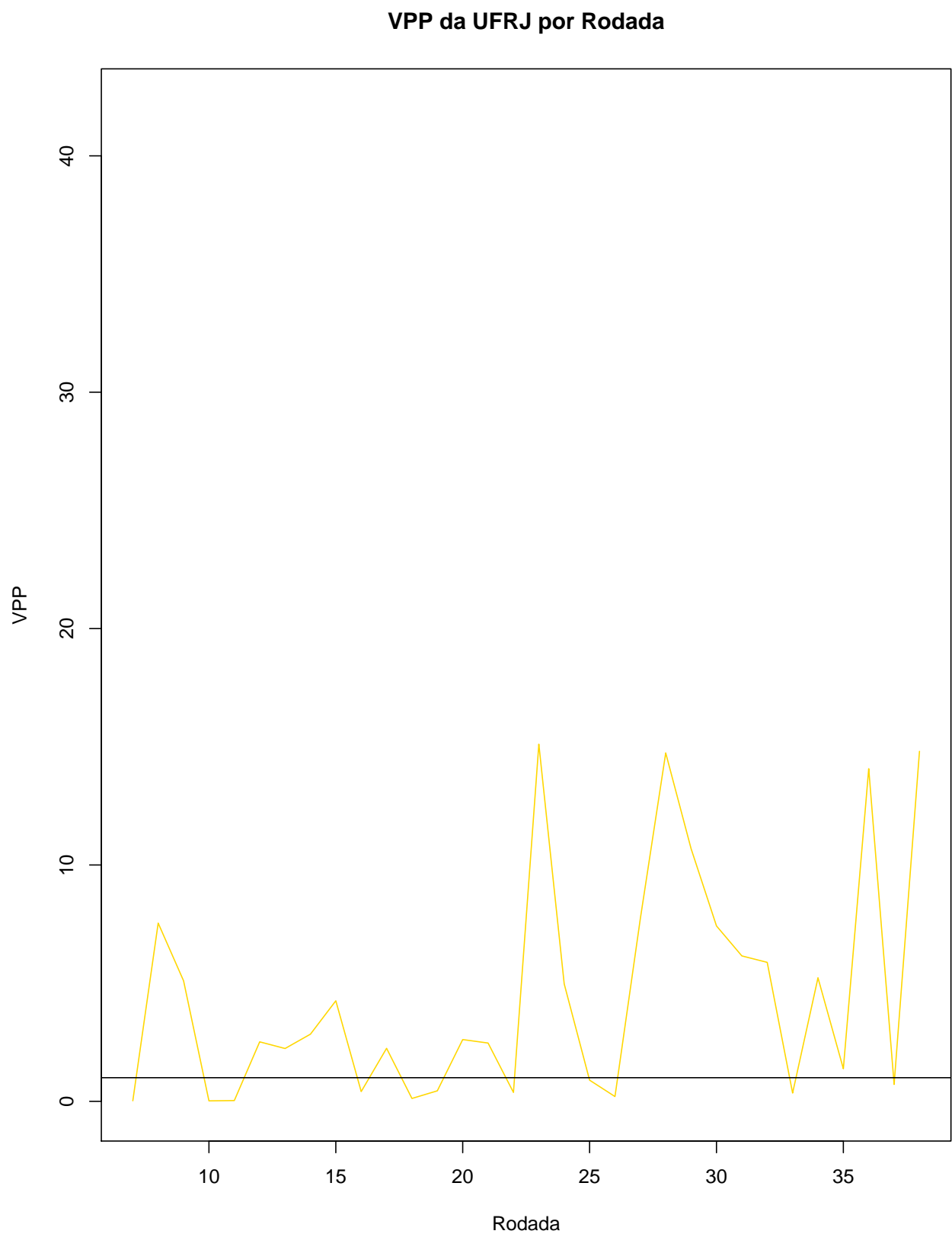


Figura A.28: Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2016.

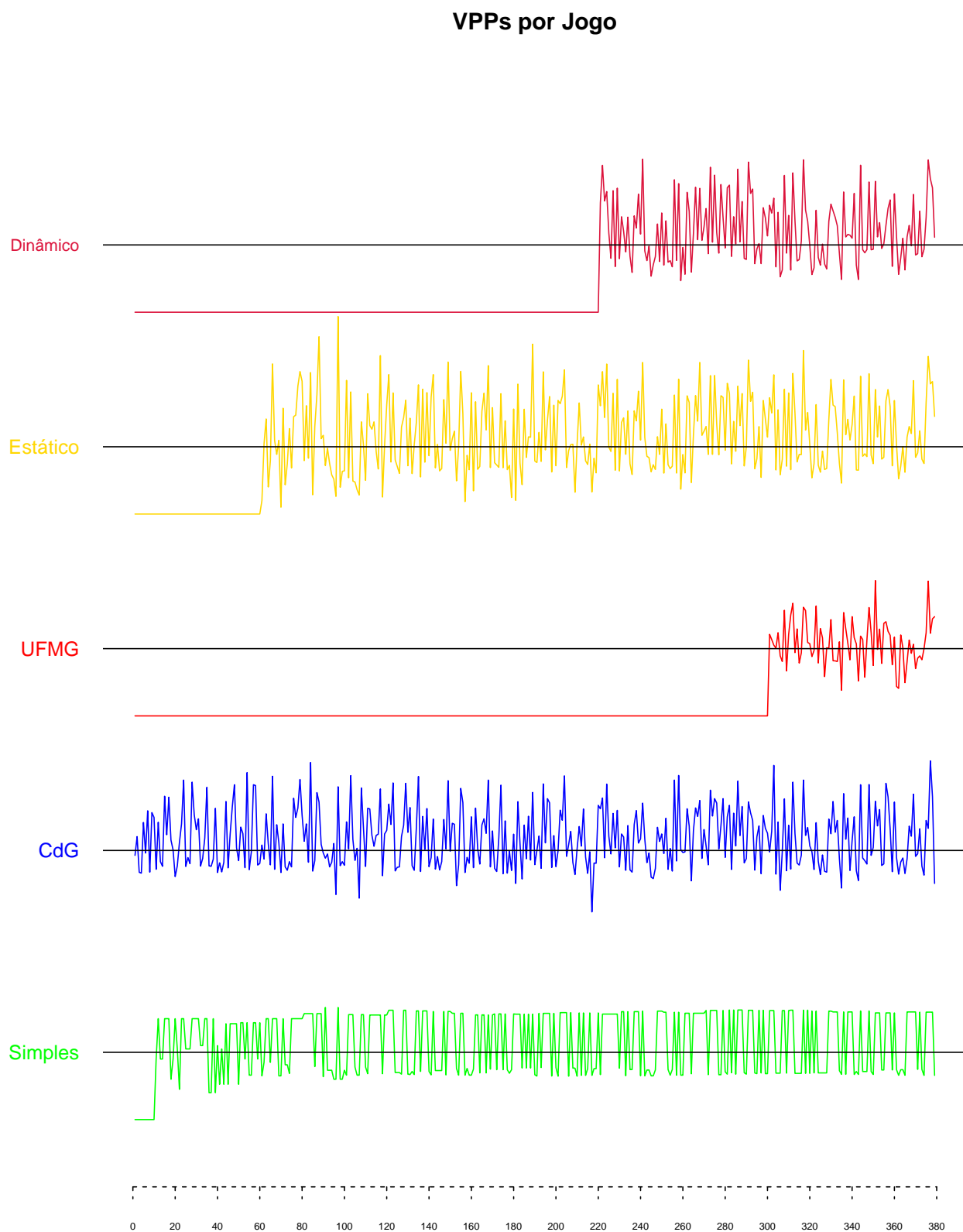


Figura A.29: Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2016.

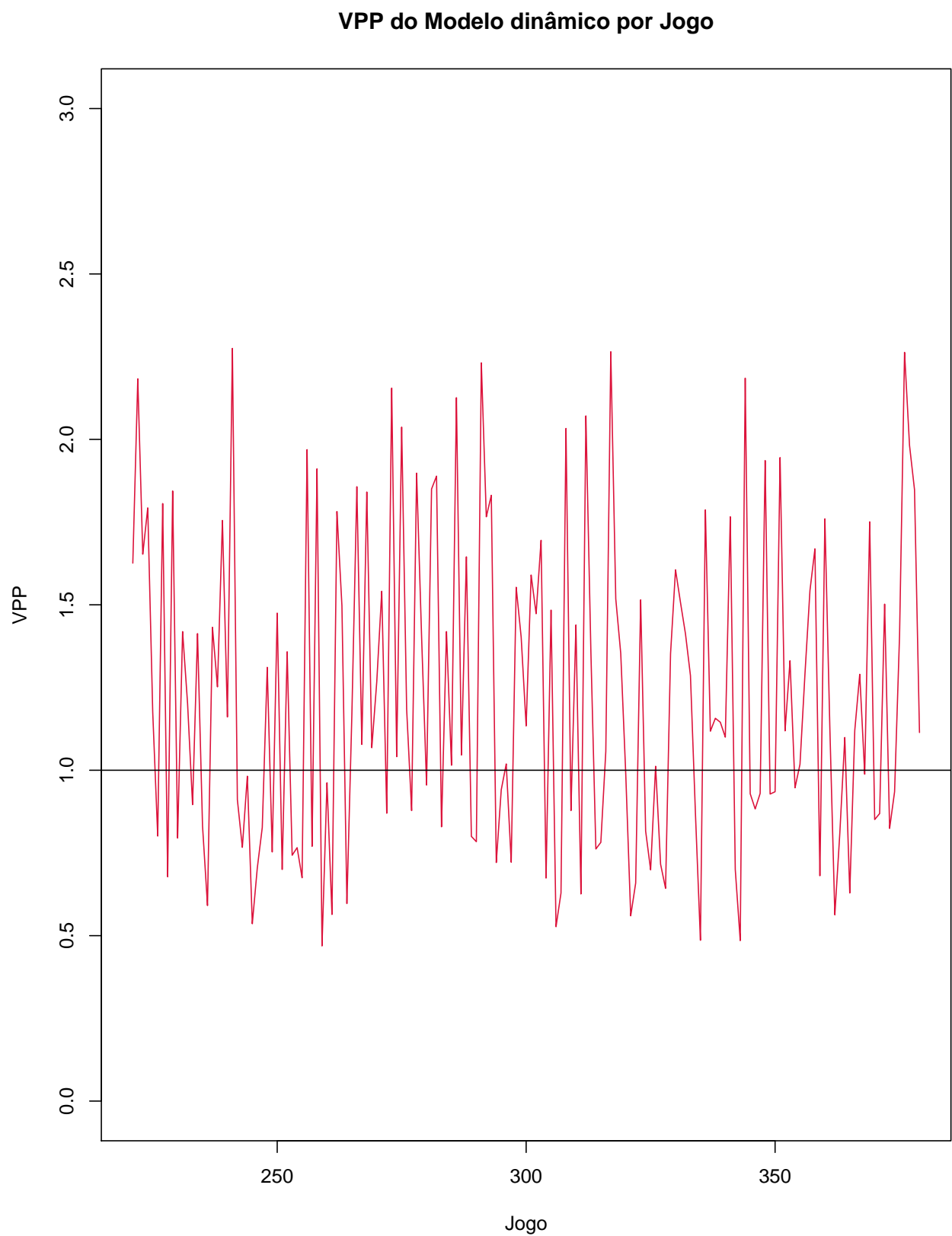


Figura A.30: Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2016.

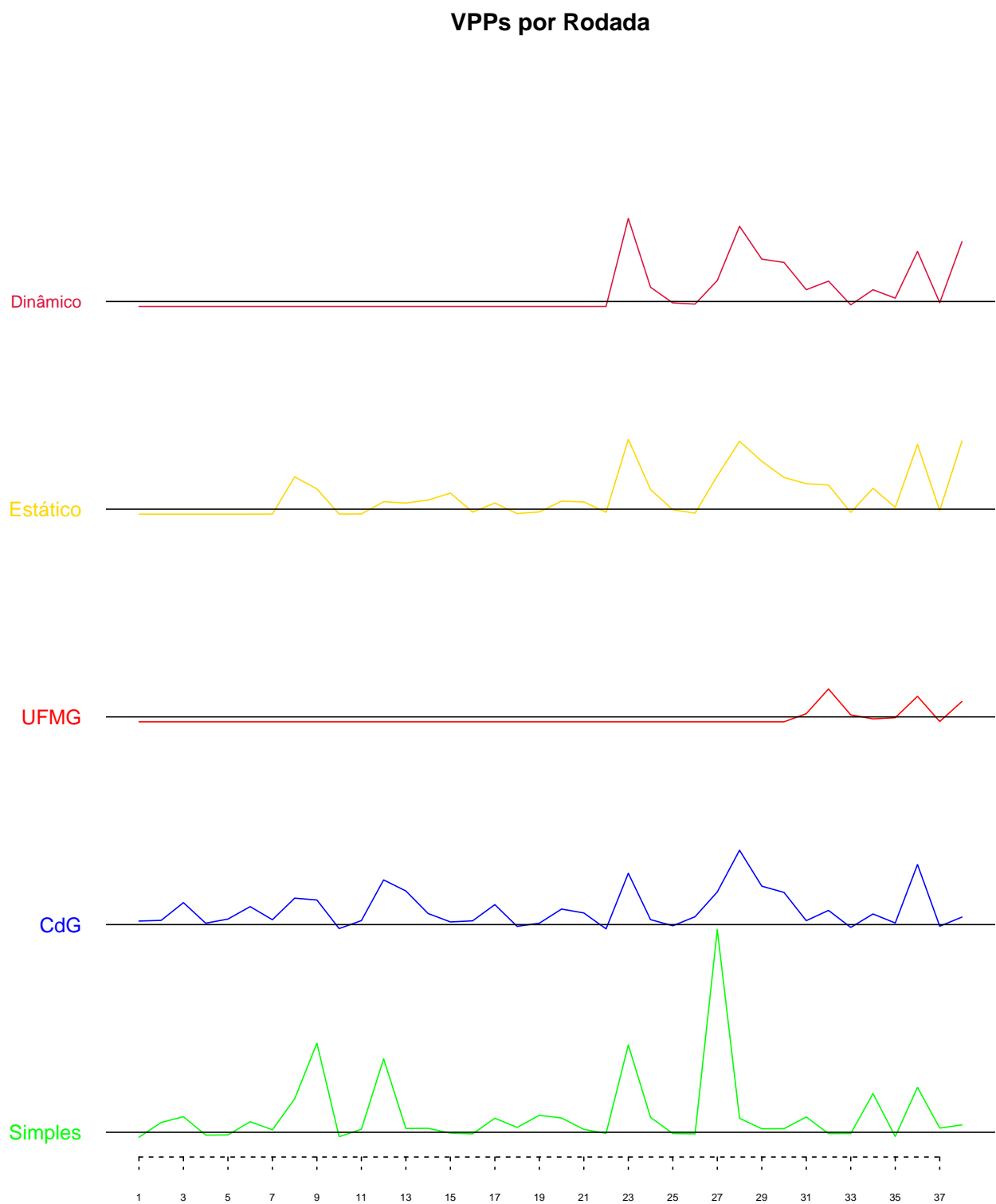


Figura A.31: Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2016.

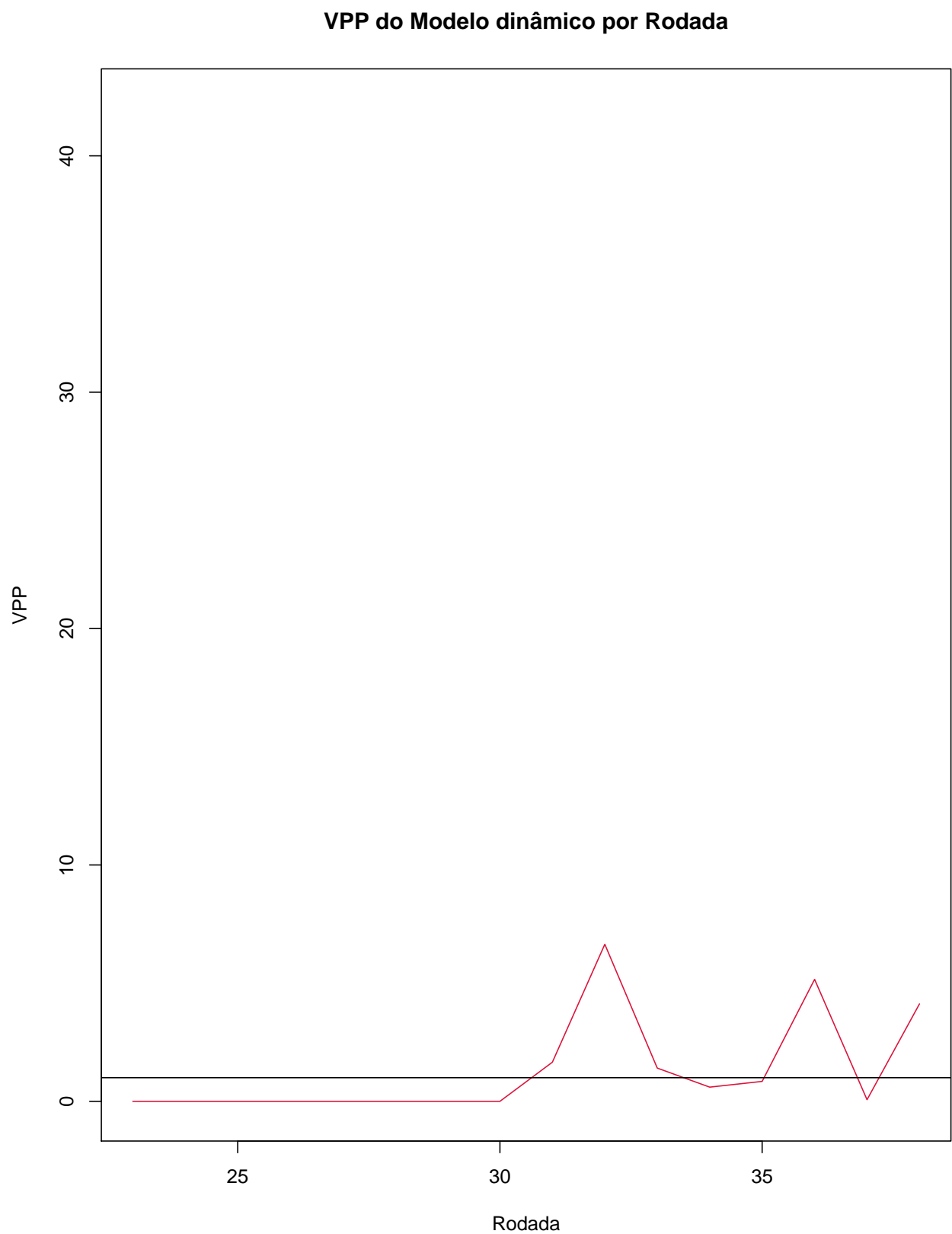


Figura A.32: Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2016.

Tabela A.1: Verossimilhanças preditivas para o Campeonato brasileiro de 2016.

Rodadas	Dinâmico	Estático	CdG	UFMG	Nulo	Simples
7 a 10	0	0,02	14,66	0	1	23,17
11 a 20	0	0,29	$2,13 \cdot 10^4$	0	1	$6,11 \cdot 10^3$
21 a 30	0	$1,14 \cdot 10^5$	$1,59 \cdot 10^5$	0	1	$2,25 \cdot 10^4$
31 a 38	3470,3	13609,47	277,43	11,45	1	137,4
Total	3470,3	$7,84 \cdot 10^6$	$1,38 \cdot 10^{13}$	11,45	1	$4,37 \cdot 10^{11}$

Tabela A.2: Verossimilhanças preditivas para o Campeonato brasileiro de 2016.

Rodadas	Dinâmico	Estático	CdG	Nulo	Simples
7 a 22	0	$4,45 \cdot 10^{-3}$	$1,37 \cdot 10^5$	1	$1,76 \cdot 10^5$
23 a 38	$5,86 \cdot 10^8$	$1,68 \cdot 10^9$	$1,01 \cdot 10^8$	1	$2,48 \cdot 10^6$

Para que se entendesse o porquê dessa mudança drástica na qualidade das previsões, foram feitas algumas investigações. Uma delas foi sobre as posteriores dos parâmetros do modelo estático. Imaginou-se que as prioris escolhidas estavam conferindo mobilidade demais aos parâmetros, por serem não informativas. Dessa forma, as amostras das posteriores desses parâmetros estariam apresentando uma variância desproporcionalmente grande antes do ponto de estabilização das previsões do campeonato.

Essa hipótese pode ser testada via uma análise das amostras das posteriores de um mesmo parâmetro, em pontos diferentes do campeonato. Caso a hipótese estivesse correta, notar-se-ia que as amostras teriam uma variância muito grande no começo do campeonato, e que ela diminuiria com o tempo, diminuindo muito pouco depois do ponto de estabilização das previsões.

No caso do Campeonato brasileiro de 2016, as previsões começaram a ser coletadas a partir da sétima rodada, o que significa que há amostras das posteriores a partir da sexta rodada. Já a estabilização das previsões ocorreu após a rodada 23. Portanto, as Figuras A.33 a A.36 apresentam histogramas das amostras aproximadas da posteriori de um mesmo parâmetro ao longo do campeonato. Foram escolhidos como pontos de interesse as rodadas seis (primeira a ter estimativas), 20 (anterior ao ponto de estabilização, mas próximo dele), 27 (posterior ao ponto de estabilização, mas próximo dele) e 38 (última rodada da competição). Na Figura A.33 estão as amostras para o fator de ataque do Coritiba, enquanto na Figura A.34 estão as amostras para o fator campo do Coritiba. Na Figura A.35, o parâmetro escolhido foi o fator de defesa do América-MG. Finalmente, na

Figura A.36 estão as amostras do intercepto do campeonato.

Uma análise dos gráficos mostra que a hipótese levanta parece ser verdadeira. Realmente há uma diminuição constante da dispersão dentro de cada amostra, enquanto as amostras tomadas após o ponto de estabilização são muito pouco diferentes, no tocante à dispersão de seus componentes. Isso parece apontar para a necessidade de novas prioris, mais informativas.

É preciso apontar que em cada gráfico não há somente os histogramas. A cada figura foi adicionada uma curva representando a densidade de uma distribuição normal. Foram definidos como os parâmetros dessa distribuição a média e a variância amostrais de cada amostra. Ou seja,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i, \quad e \quad (A.4)$$

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad (A.5)$$

onde n é o tamanho da amostra, cada x_i representa um valor dela e $\hat{\mu}$ substitui μ na equação (3.12), enquanto $\hat{\sigma}^2$ substitui σ^2 .

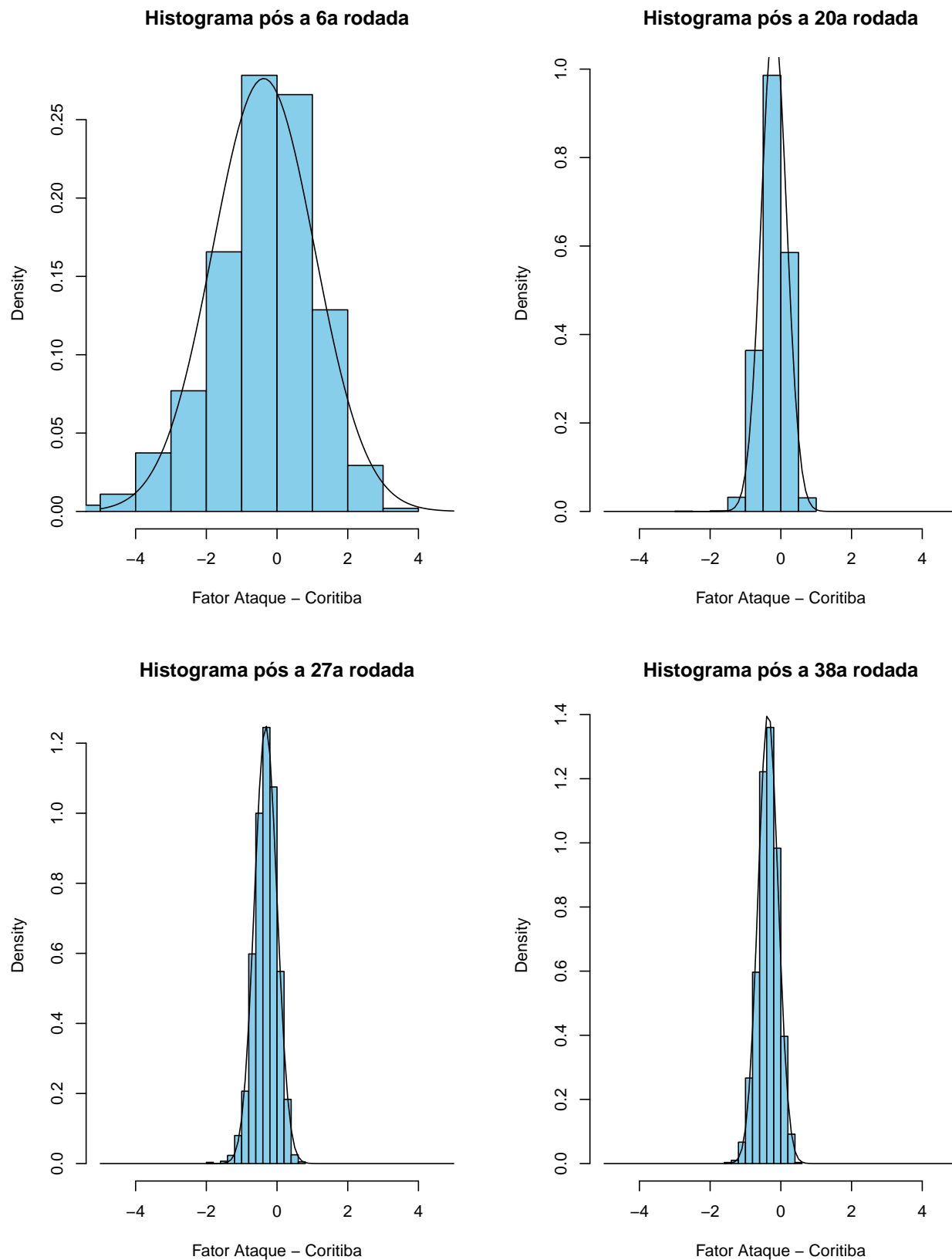


Figura A.33: Amostras aproximadas da posteriori do fator de ataque do Coritiba.

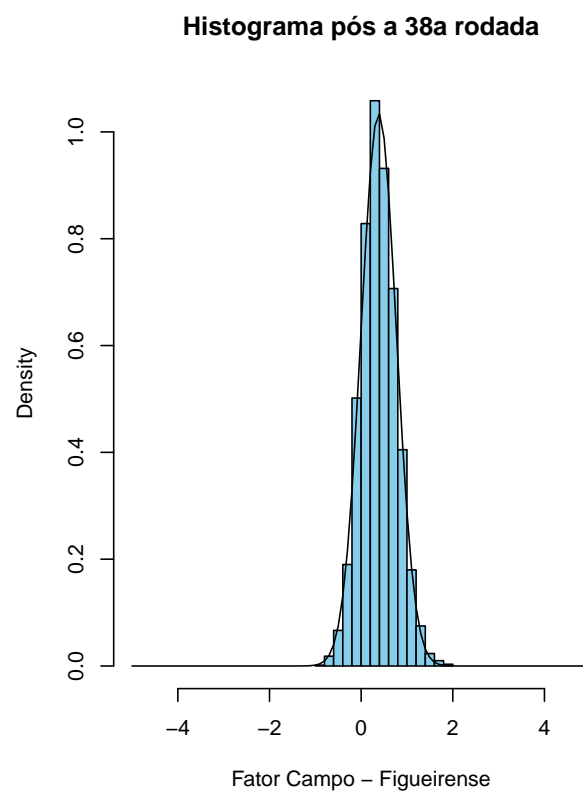
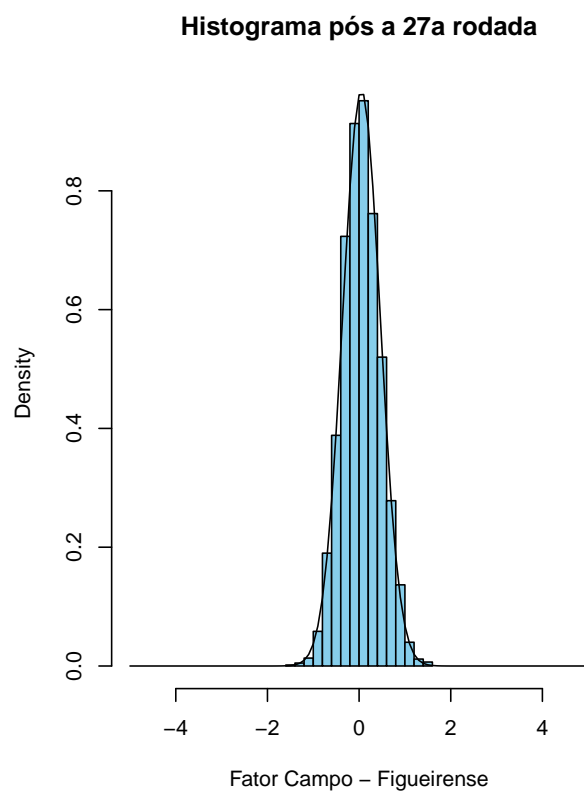
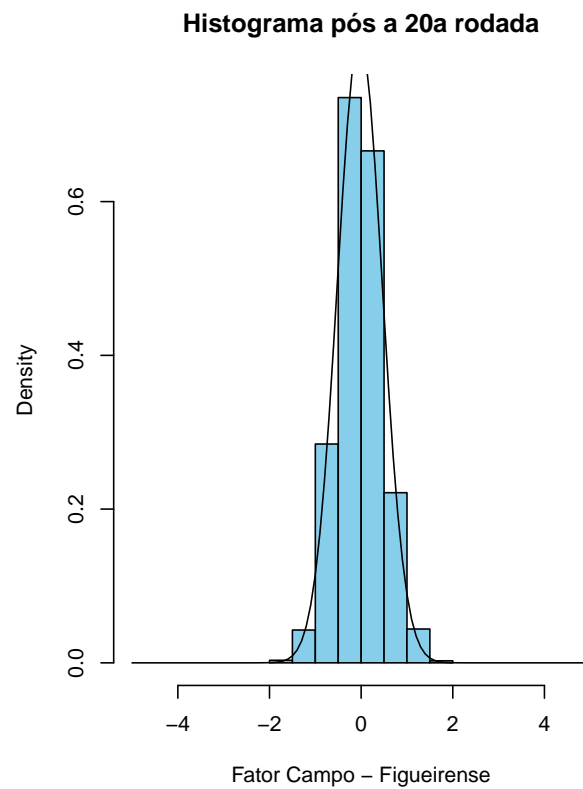
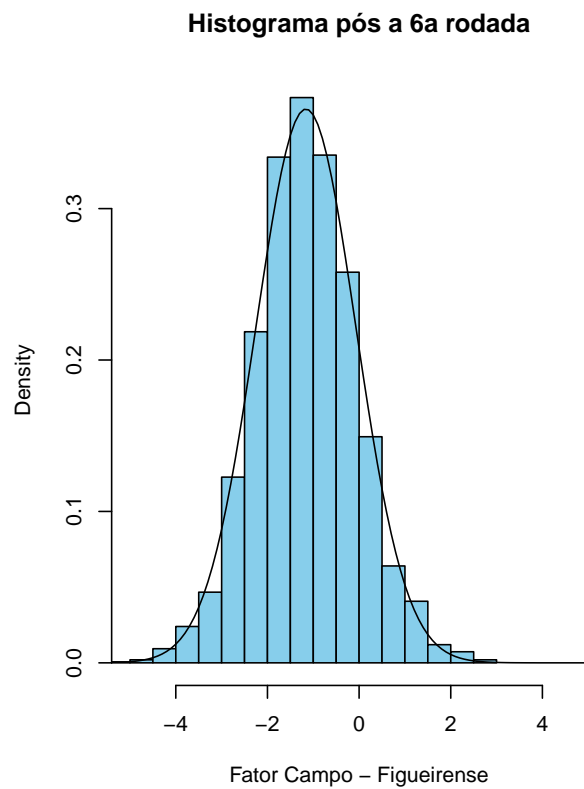


Figura A.34: Amostras aproximadas da posteriori do fator campo do Figueirense.

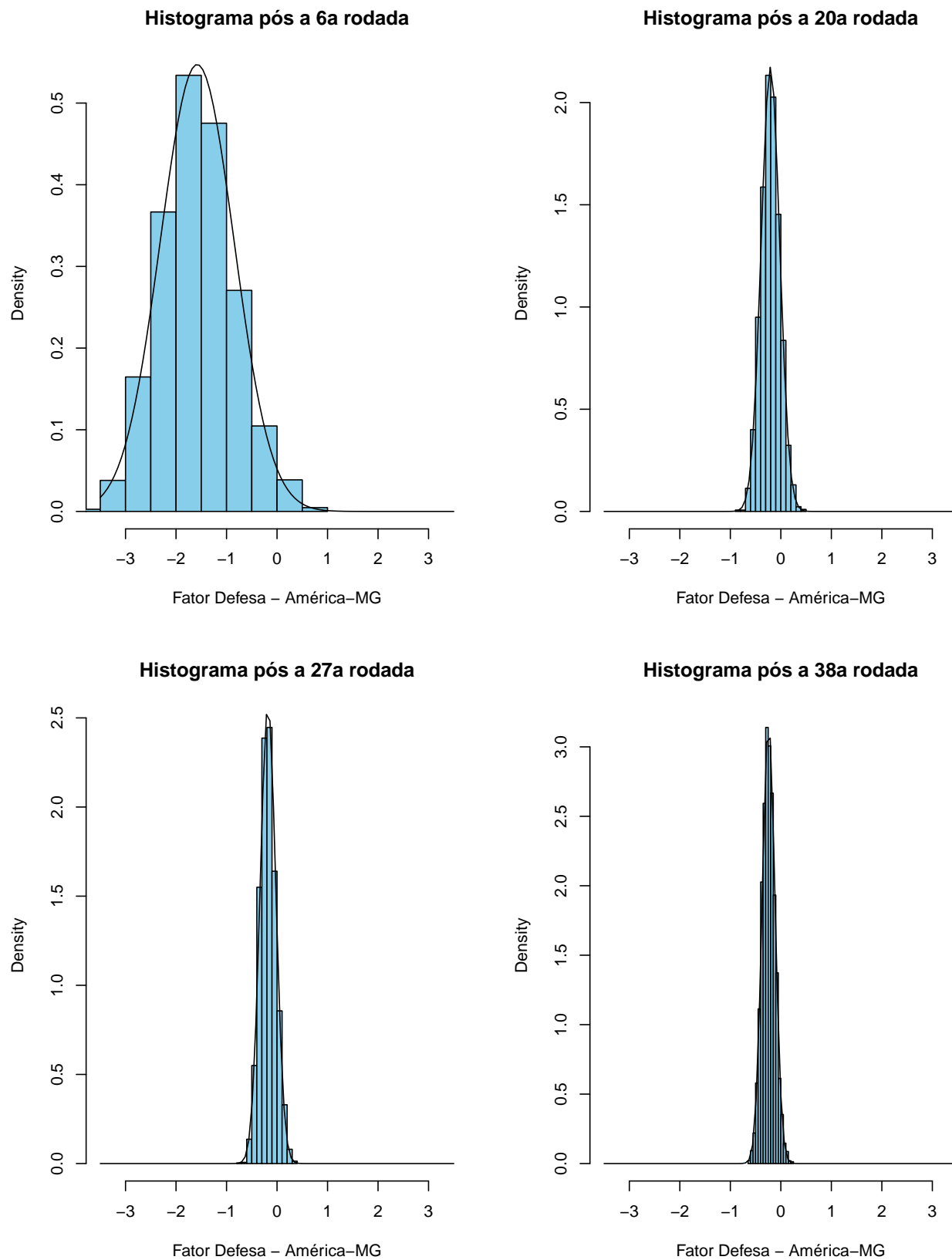


Figura A.35: Amostras aproximadas da posteriori do fator de defesa do América-MG.

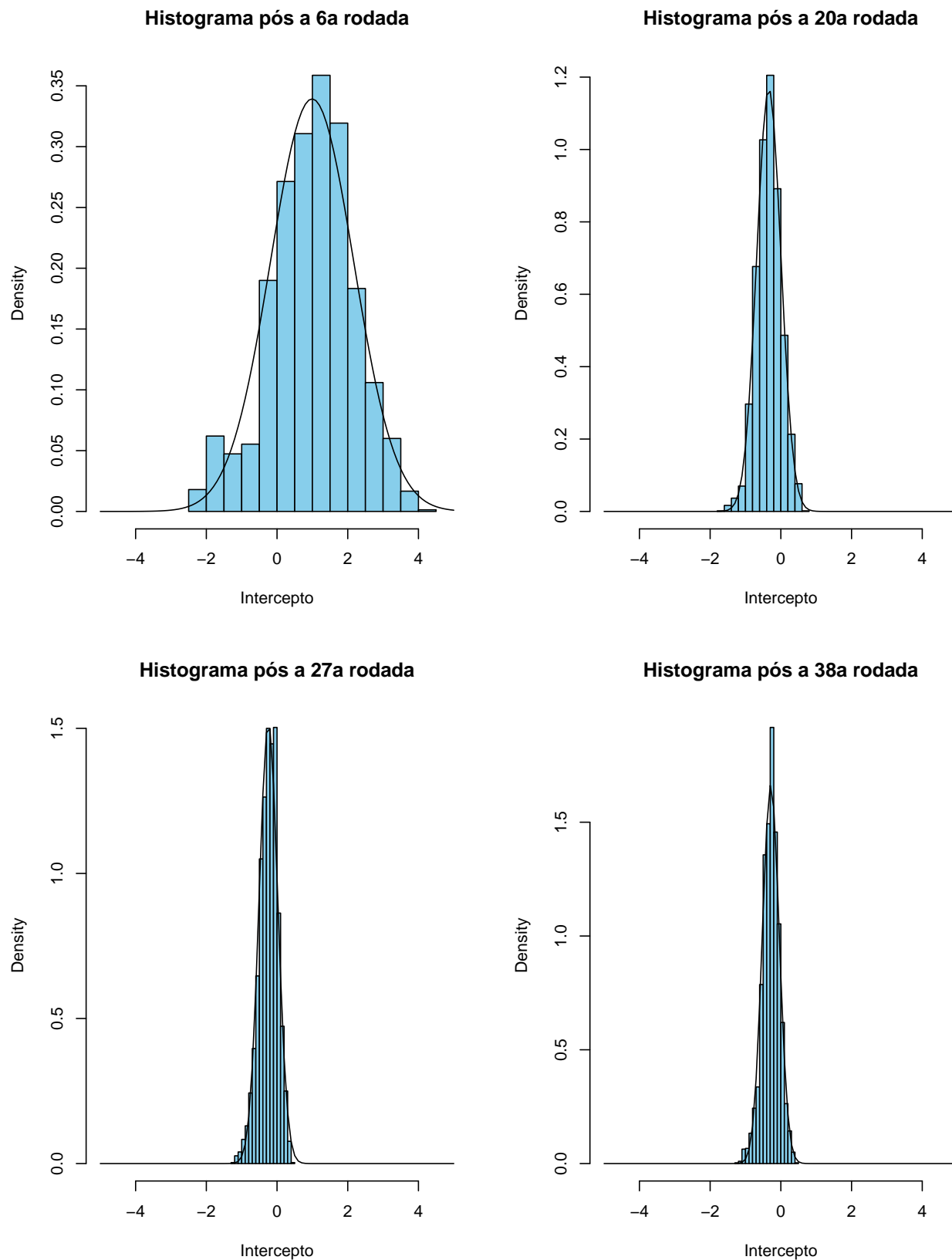


Figura A.36: Amostras aproximadas da posteriori do intercepto do Campeonato brasileiro de 2016.

A.4 Campeonato brasileiro de 2017

As figuras e as tabelas analisadas nesta seção se referem à edição de 2017 do campeonato brasileiro, e foram geradas a partir dos resultados dos modelo estático e dinâmico para esse torneio. Na subseção [A.4.1](#), o modelo discutido é o estático, enquanto o dinâmico é comentado na subseção seguinte, [A.4.2](#).

A.4.1 Análises e resultados do Modelo estático

As figuras analisadas nessa subseção se referem à edição de 2017 do campeonato brasileiro, e foram geradas a partir dos resultados do modelo estático para esse torneio.

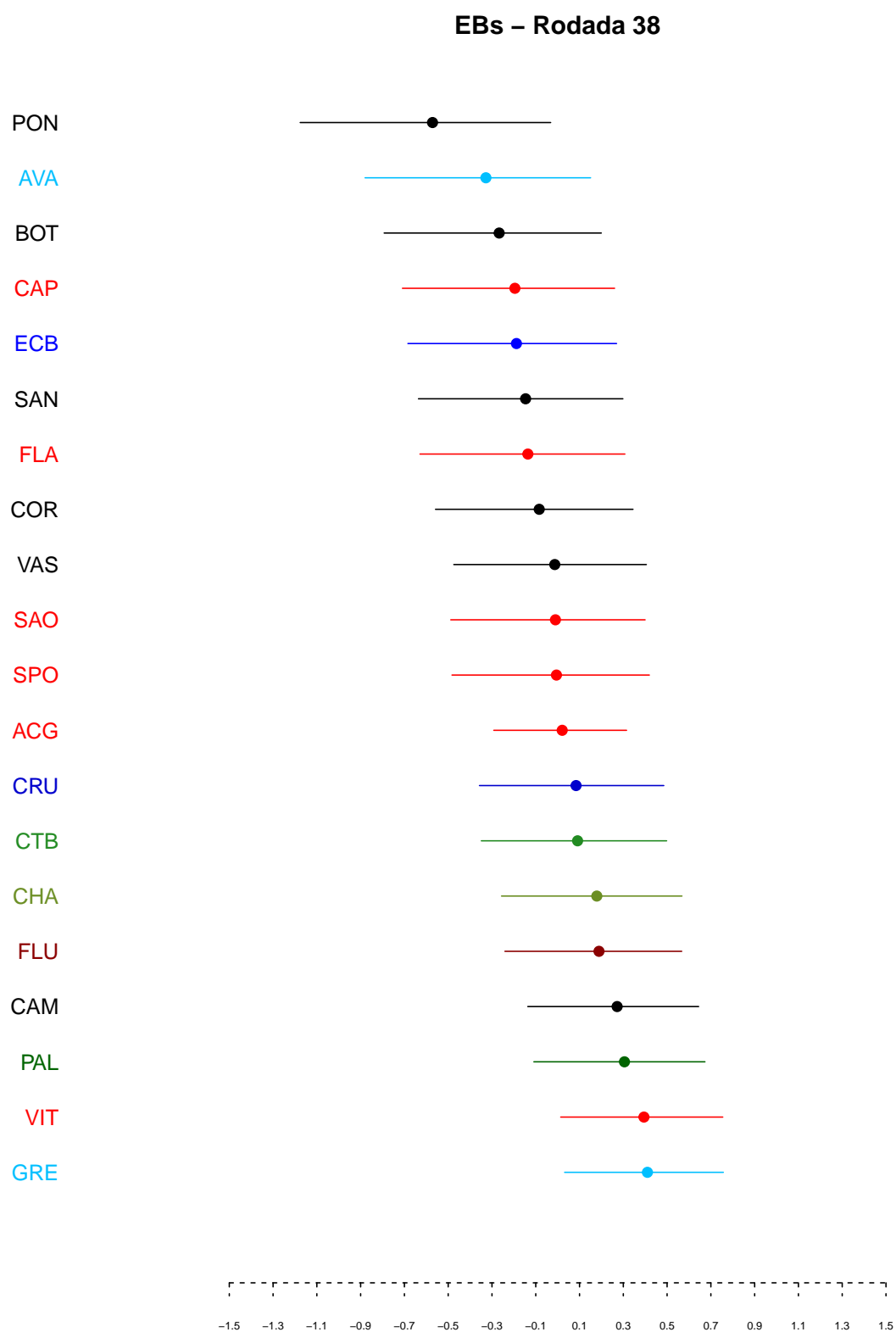


Figura A.37: Estimativas do fator ataque para o Campeonato brasileiro de 2017.

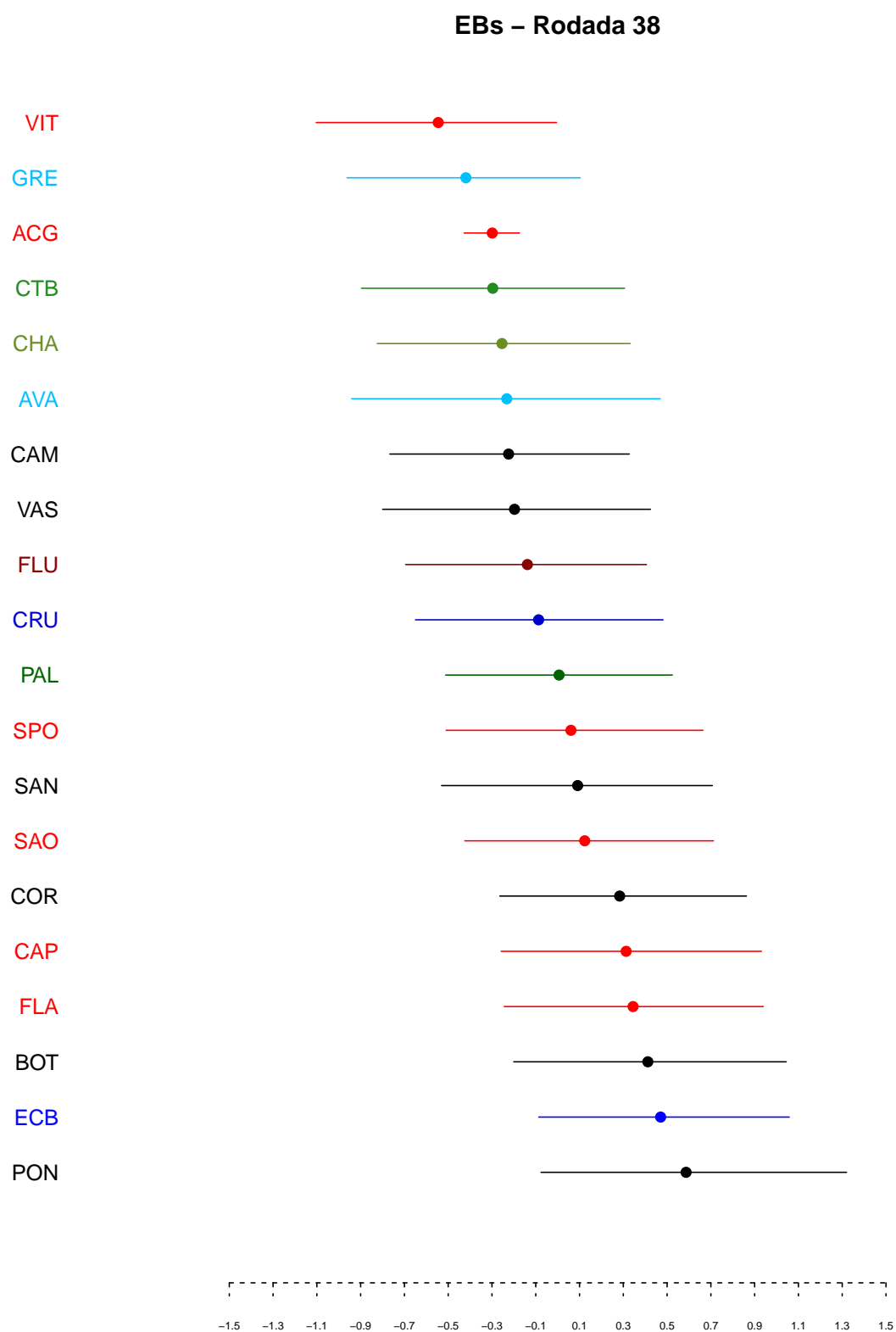


Figura A.38: Estimativas do fator campo para o Campeonato brasileiro de 2017.

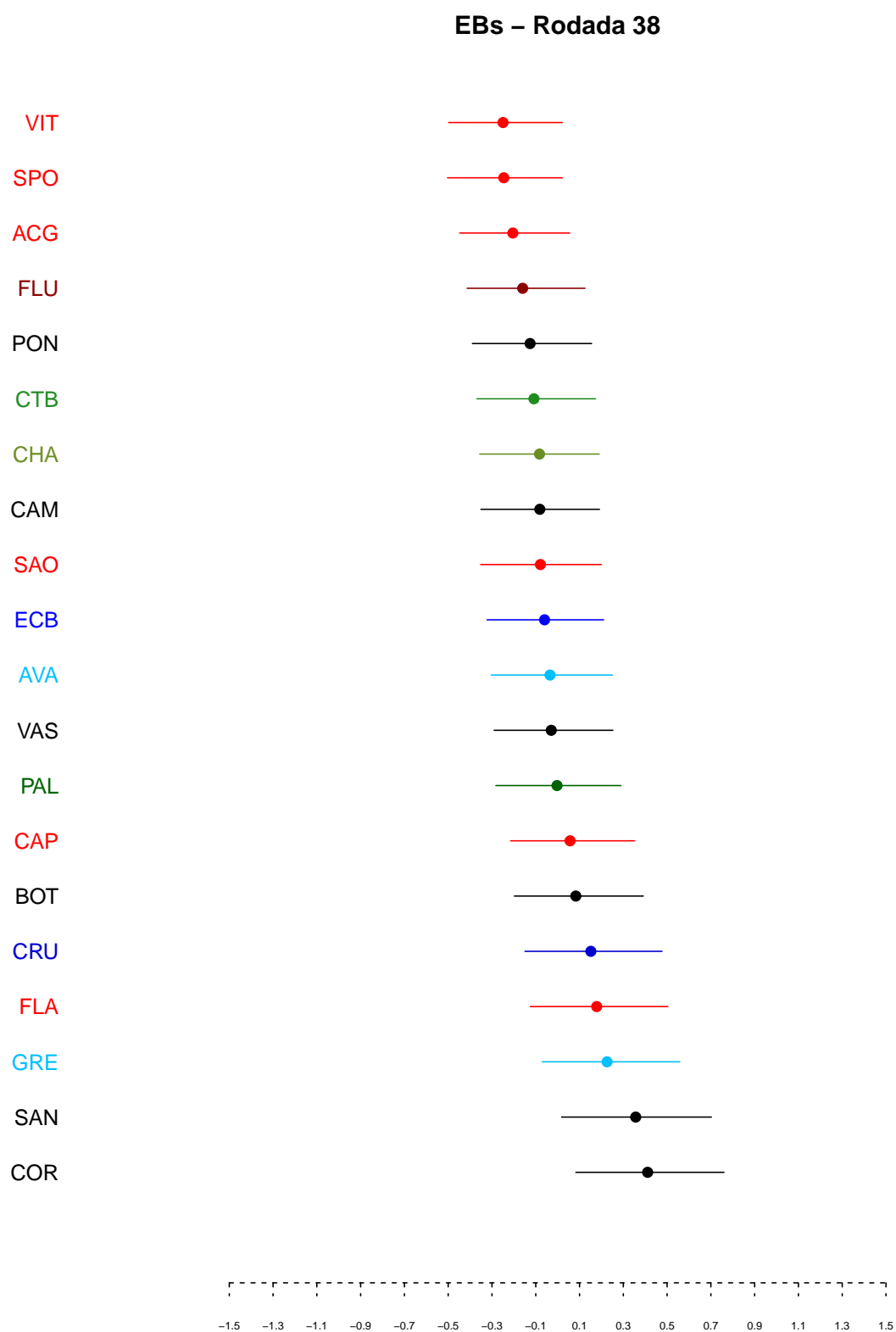


Figura A.39: Estimativas do fator defesa para o Campeonato brasileiro de 2017.

A partir de uma análise das Figuras A.37 a A.39, é possível notar que o Campeonato brasileiro de 2017 se assemelha à sua edição anterior, em alguns aspectos. Primeiramente, times como Botafogo, Bahia, Flamengo e Ponte Preta apresentam a combinação de ataque fraco com mando de campo forte que aponta para uma dificuldade do modelo de separar propriamente os parâmetros. Nesse quesito, é natural indicar que o Atlético-PR volta a ter essa combinação, o que pode significar que essa equipe, em especial, tem uma grande vantagem por jogar em seu estádio.

Em seguida, pode-se apontar que Grêmio e Vitória têm o problema oposto, o de um ataque forte combinado com um mando de campo fraco. Levando em conta que o Flamengo fez parte desse grupo no ano anterior, e foi citado no grupo de mando de campo forte, nesse ano, pode-se especular que a qualidade do mando de campo de um time flutua muito mais do que a qualidade de seu ataque e de sua defesa.

O campeão Corinthians claramente pode dar à sua defesa o crédito pelo título, visto que seu ataque e seu mando de campo não se destacaram fortemente, enquanto sua defesa foi a melhor do campeonato, com o único time a ter uma defesa de qualidade semelhante sendo o terceiro colocado Santos.

Olhando agora para os times rebaixados, pode-se ver que o Atlético-GO teve um desempenho semelhante ao do Santa Cruz, no ano anterior, sendo rebaixado por causa de sua defesa. Já a Ponte Preta pode culpar sua defesa e sua incapacidade de fazer gols fora de casa. Finalmente, é difícil de ver um porquê definitivo para o rebaixamento de Avaí e Coritiba, enquanto o Vitória continuou na primeira divisão, assim como no ano anterior, quando o Vitória ficou na primeira divisão e o Internacional foi rebaixado. Em ambos os anos, o Vitória pôde culpar sua defesa pelo baixo rendimento, e agradecer a seu ataque, pelo não-rebaixamento.

A.4.2 Análises e resultados do Modelo dinâmico

Nessa subseção, sai-se do modelo estático para o dinâmico. Dessa forma, as análises feitas serão sobre figuras relacionadas às estimativas do modelo dinâmico para seus parâmetros.

Para o Campeonato brasileiro de 2017, essas estimativas estão contidas nas Figuras A.40 a A.54. As Figuras A.40 a A.44 contêm as estimativas para os fatores de ataque das equipes do campeonato. Já as Figuras A.45 a A.49 contêm as estimativas para os fatores campo, enquanto as estimativas para os fatores de defesa estão nas Figuras A.50 a A.54.

No campeonato, o Vitória voltou a não ser rebaixado por causa de uma melhora de

seu ataque, ao final da competição. Diferentemente do Internacional, nem Avaí nem Coritiba tiveram quedas em seu desempenho. O Avaí chegou a melhorar seu ataque, também. Ao final, o Coritiba foi rebaixado por causa da melhora súbita do Vitória, tal como o Internacional, enquanto o Avaí não ficou na primeira divisão porque sua melhora não começou cedo o suficiente. A equipe teve desempenho decepcionante no começo do campeonato, o que a deixou atrás do Vitória até o final da competição.

Já Corinthians e Grêmio apresentaram comportamento oposto. Os ataques de ambos os times pioraram ao longo da competição, enquanto a defesa do Corinthians fez o mesmo. Durante grande parte do campeonato, as duas equipes foram as primeiras colocadas, mas as quedas em seus desempenhos levaram a consequências negativas, especialmente para o Grêmio. Para o Corinthians, sua queda significou apenas que seus competidores mais próximos se aproximaram, mas o time ainda foi o vencedor da competição. Para o Grêmio, sua queda de qualidade levou a uma queda em seu posicionamento no torneio, com a equipe terminando em quarto lugar, apesar de ter sido a segunda colocada por muito tempo.

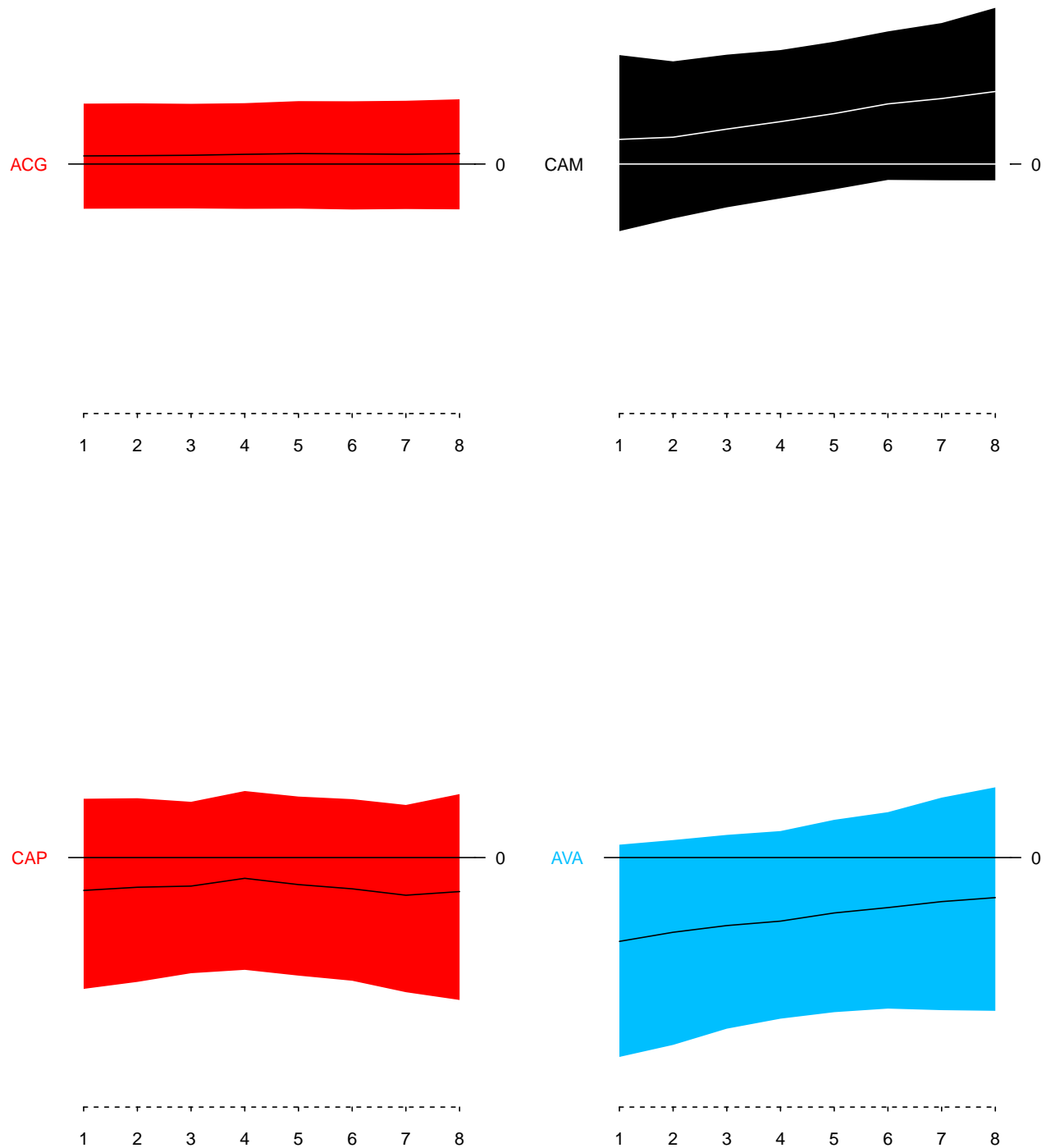


Figura A.40: Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 1.

EBs – Modelo Dinâmico

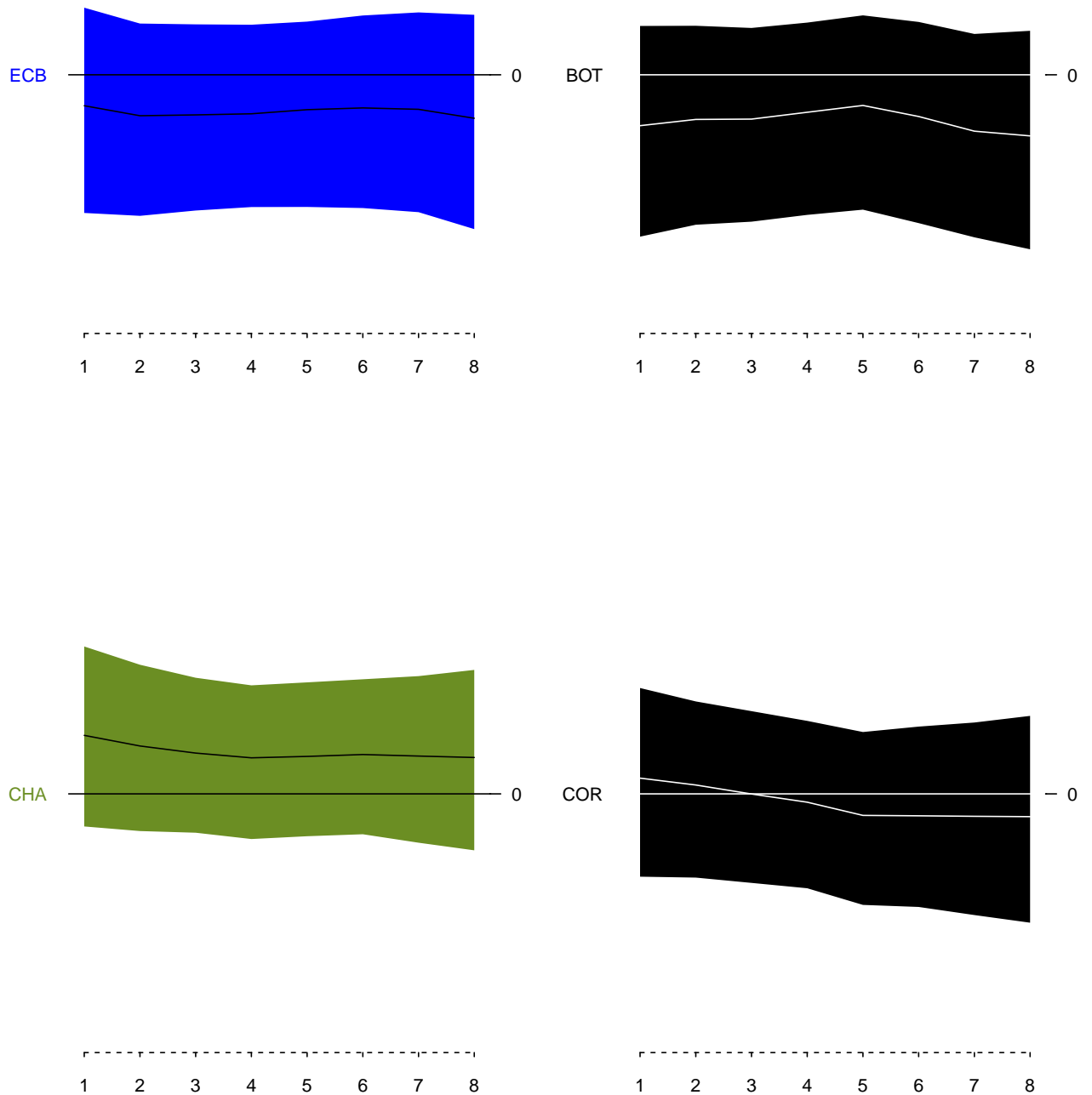


Figura A.41: Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 2.

EBs – Modelo Dinâmico

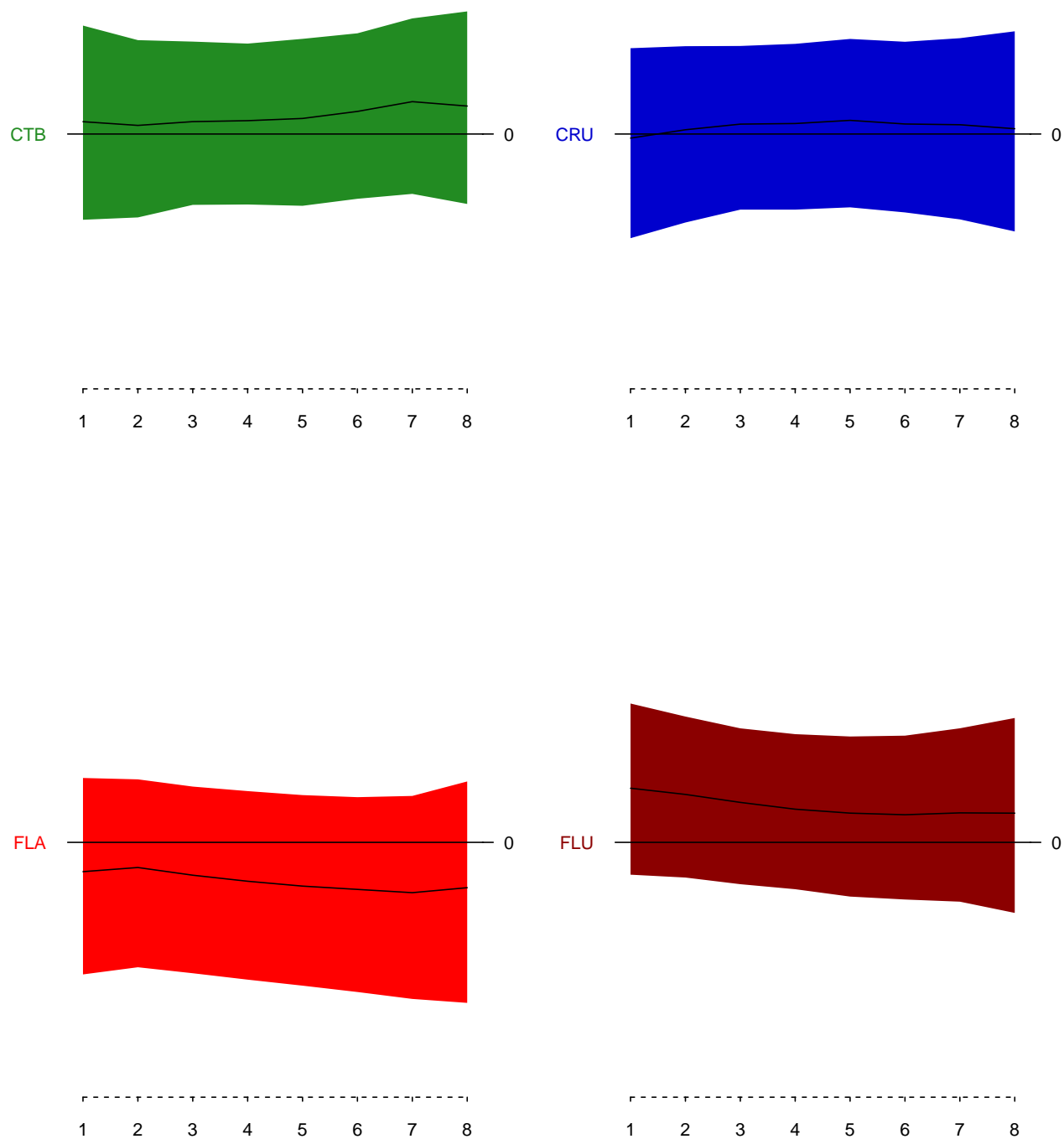


Figura A.42: Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 3.

EBs – Modelo Dinâmico

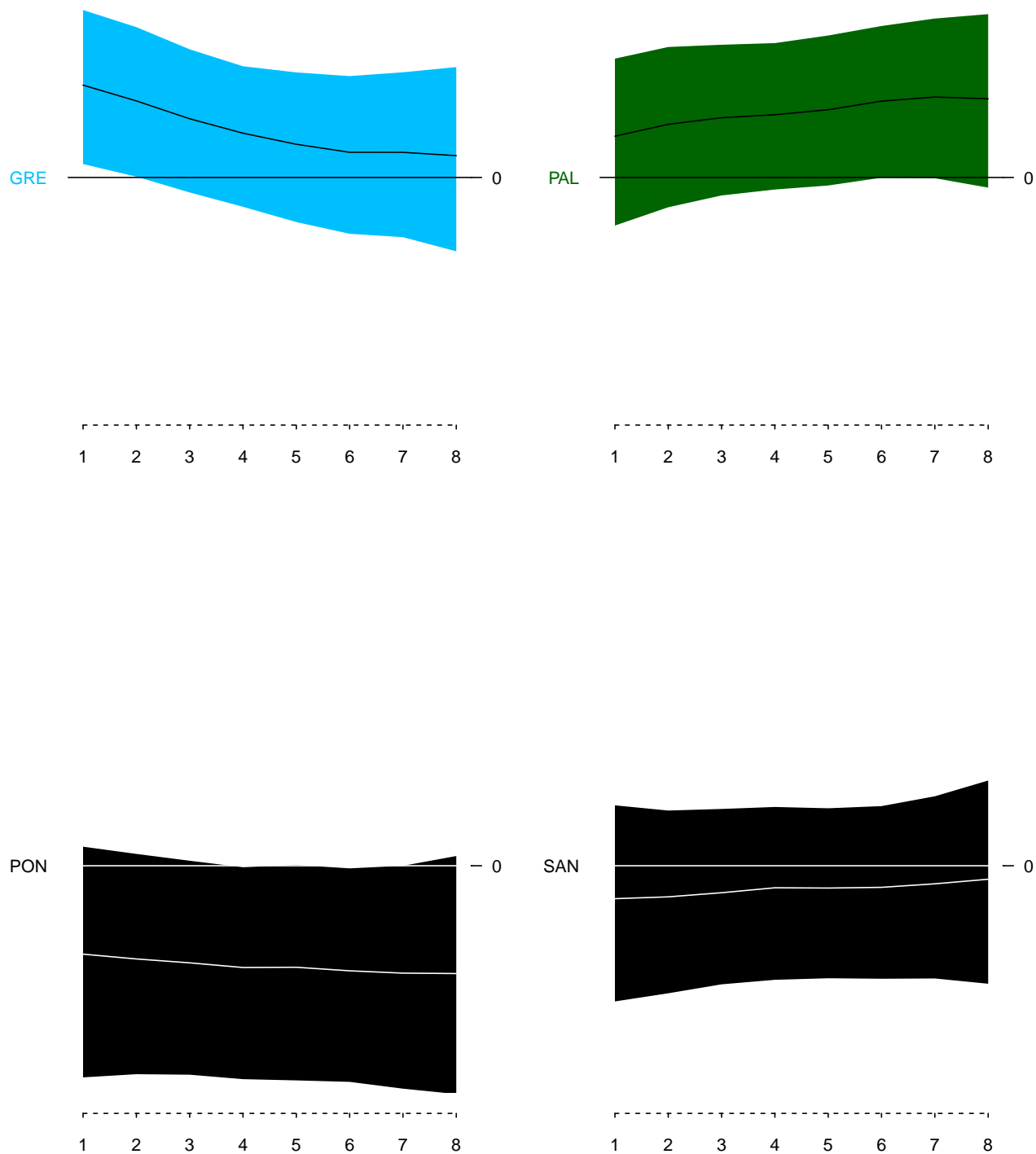


Figura A.43: Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 4.

EBs – Modelo Dinâmico

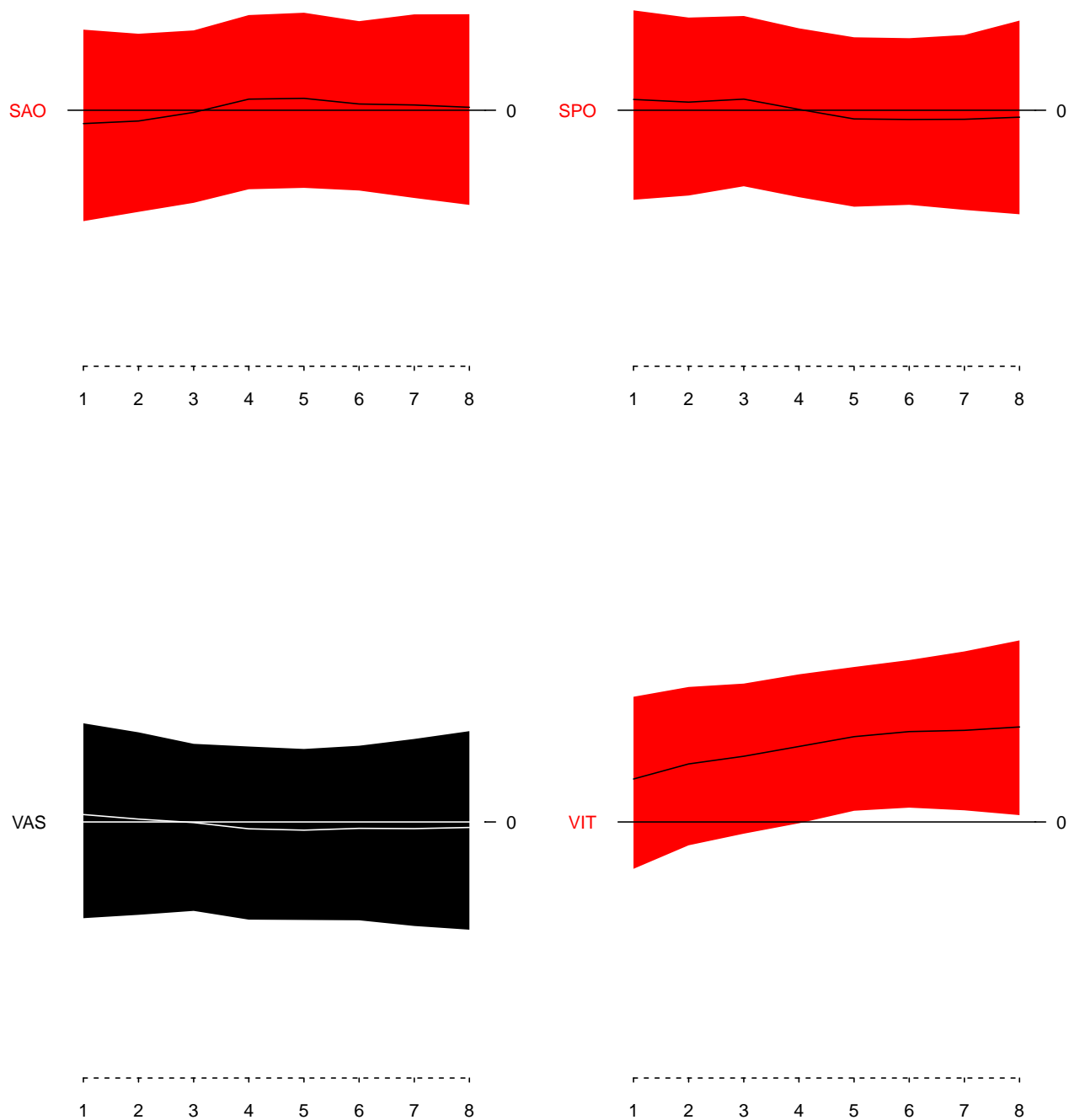


Figura A.44: Estimativas do fator ataque para o Campeonato brasileiro de 2017 - Parte 5.

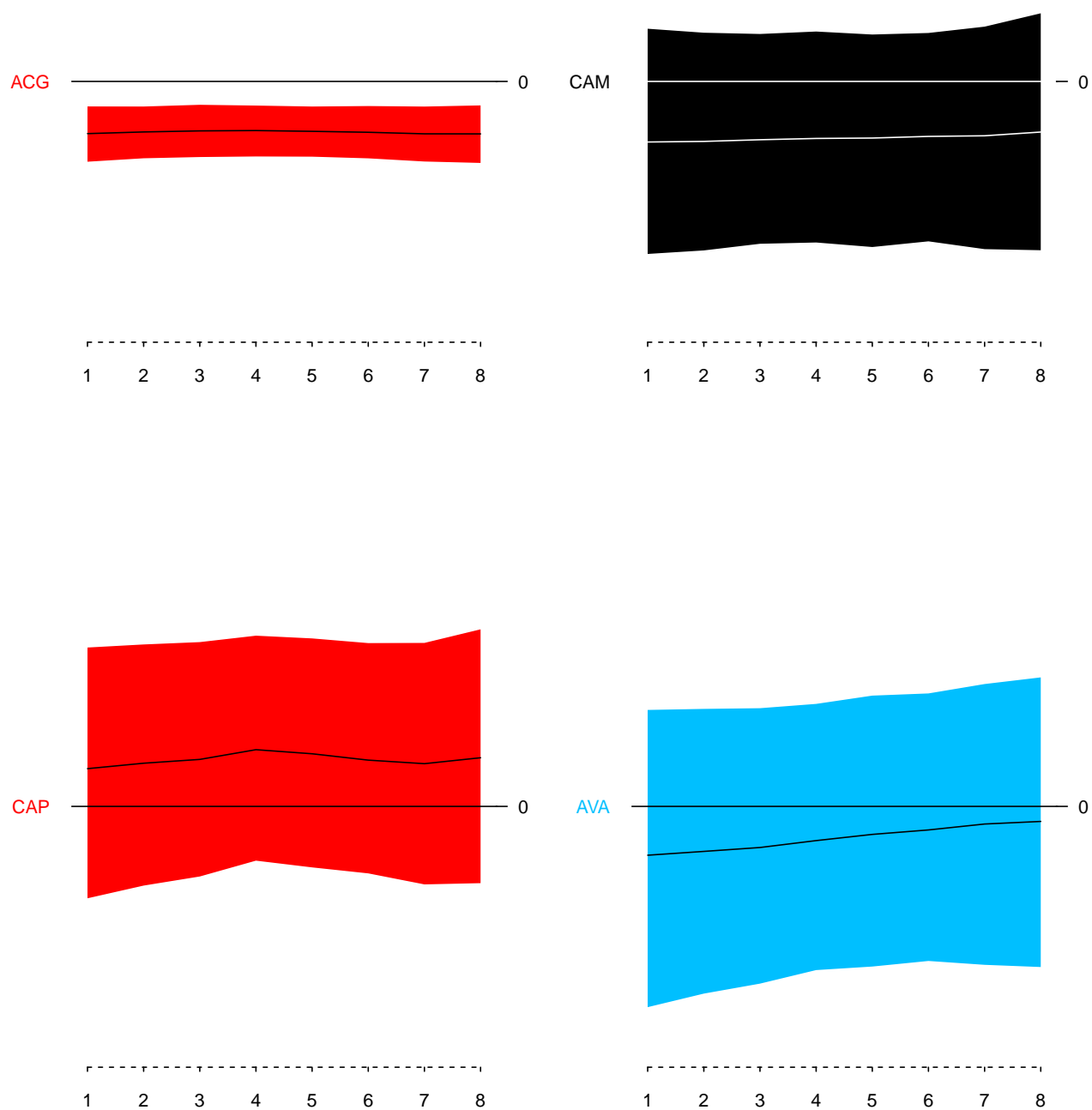


Figura A.45: Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 1.

EBs – Modelo Dinâmico

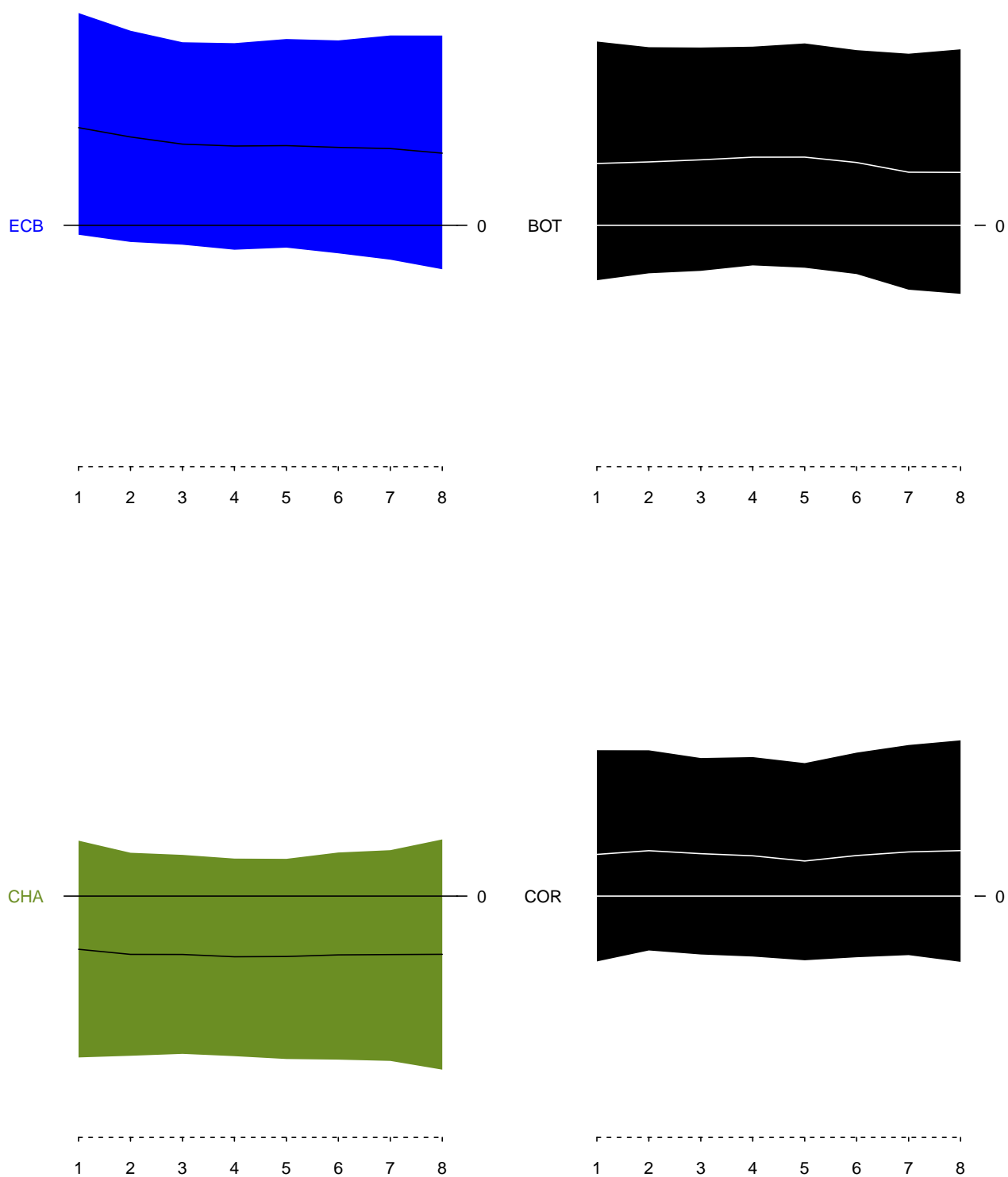


Figura A.46: Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 2.

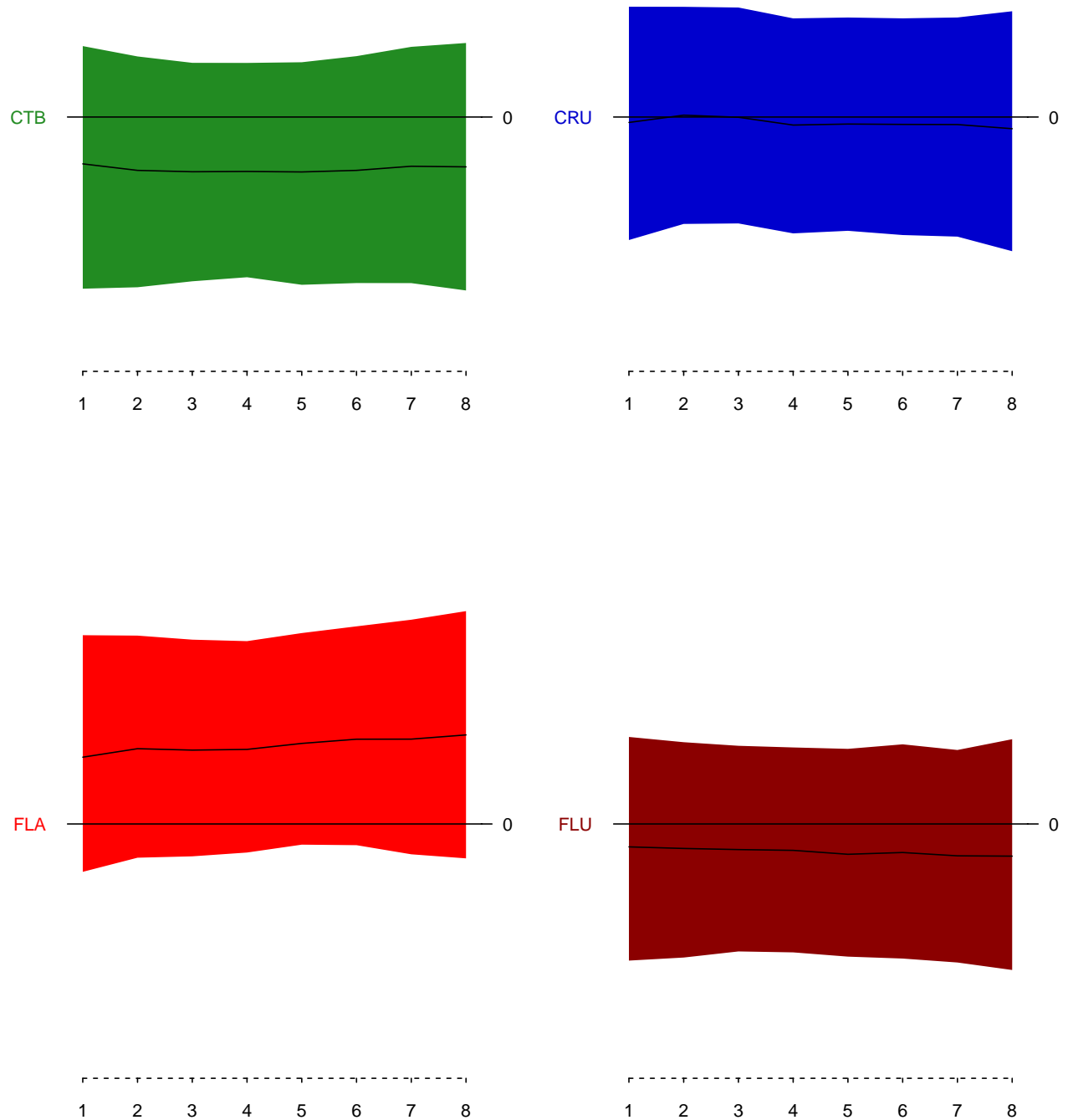


Figura A.47: Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 3.

EBs – Modelo Dinâmico

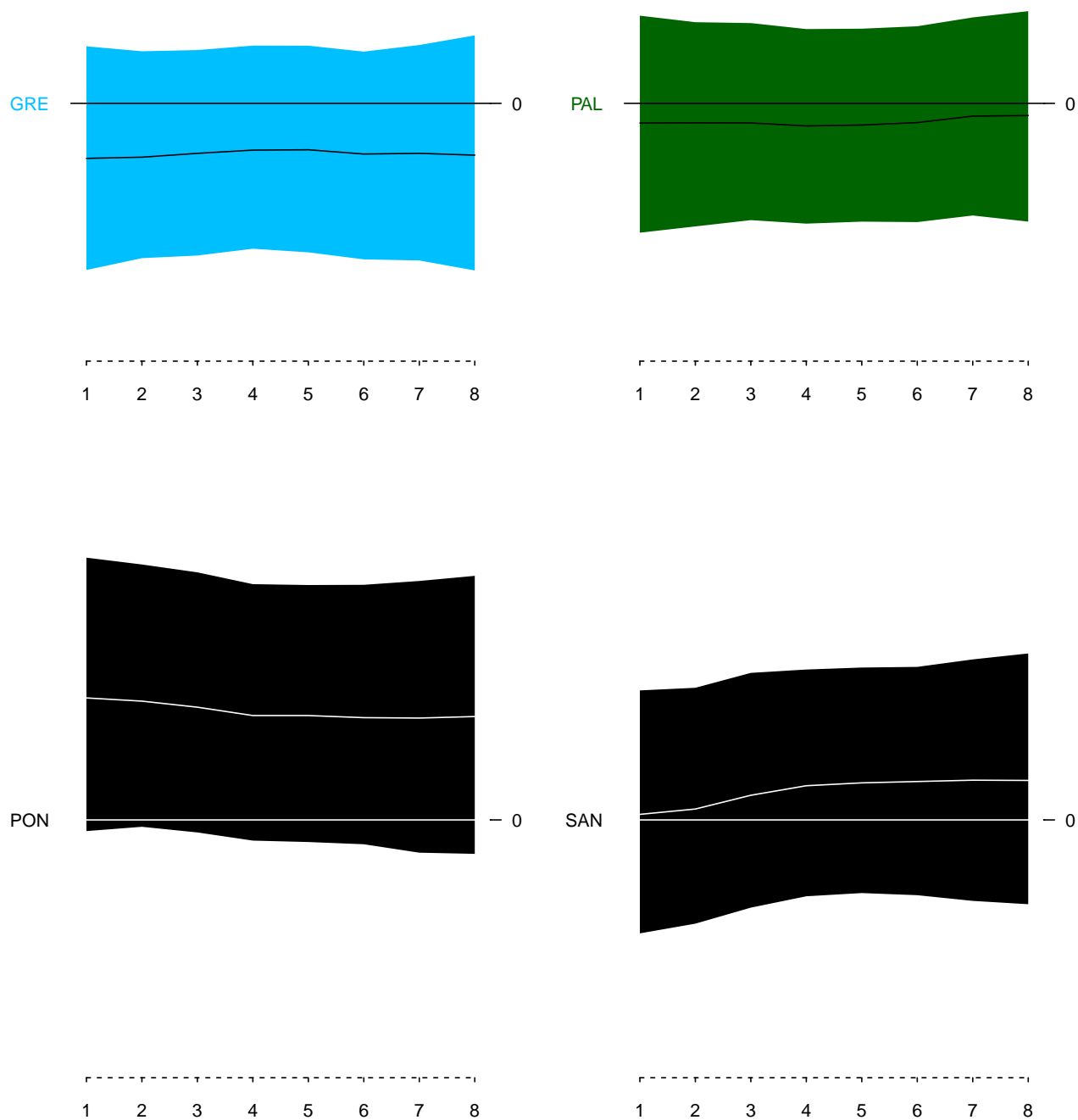


Figura A.48: Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 4.

EBs – Modelo Dinâmico

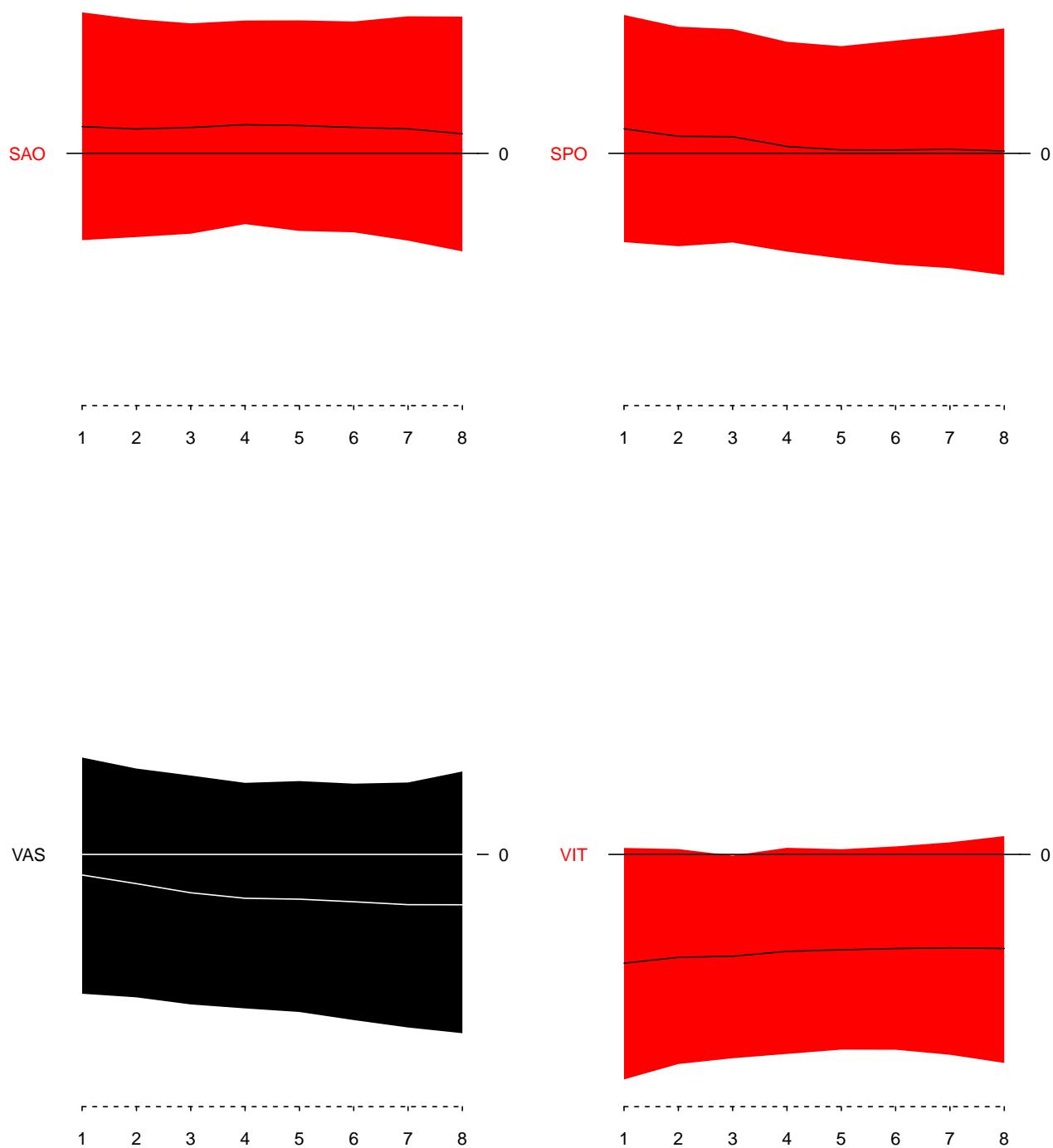


Figura A.49: Estimativas do fator campo para o Campeonato brasileiro de 2017 - Parte 5.

EBs – Modelo Dinâmico

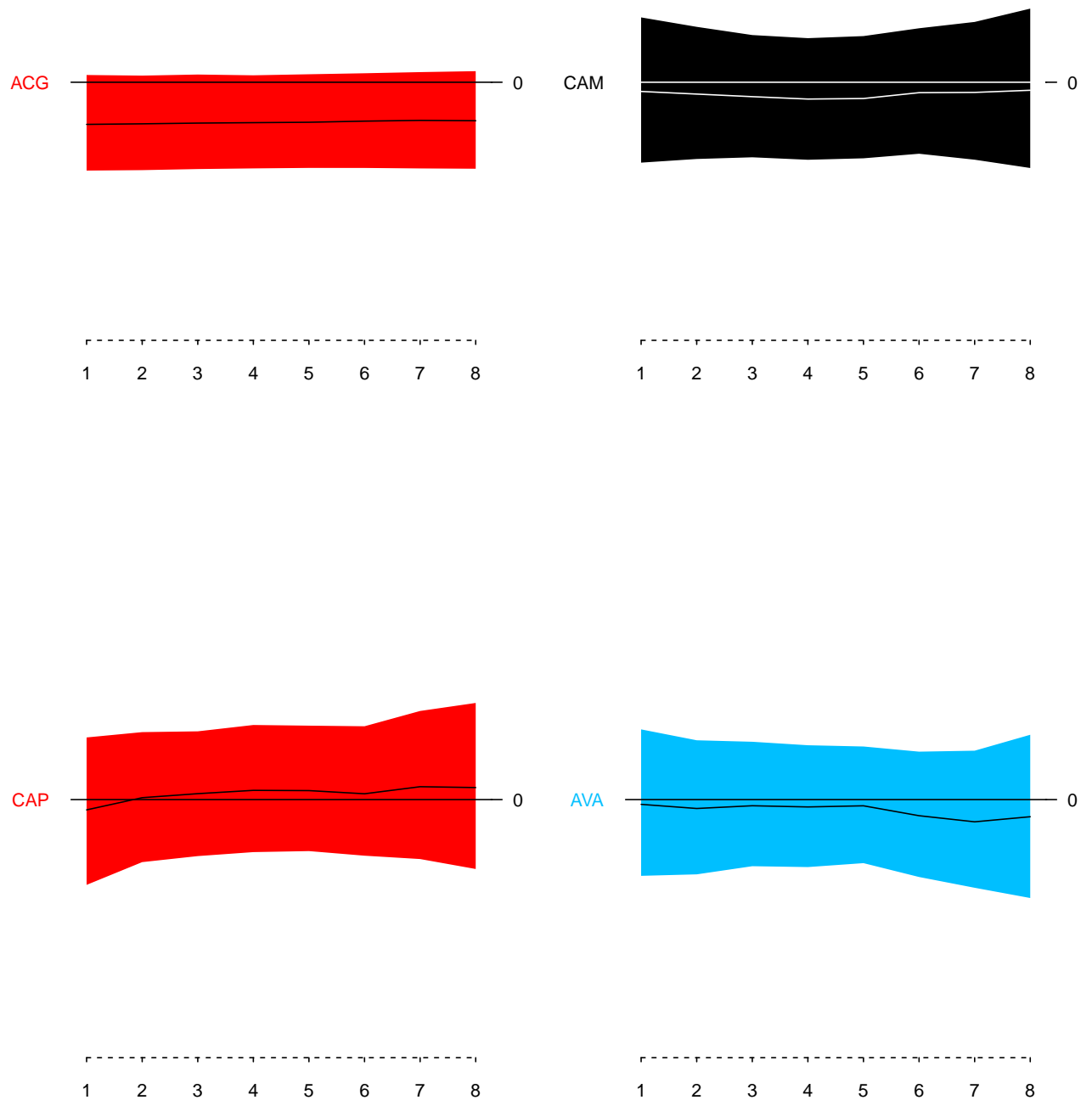


Figura A.50: Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 1.

EBs – Modelo Dinâmico

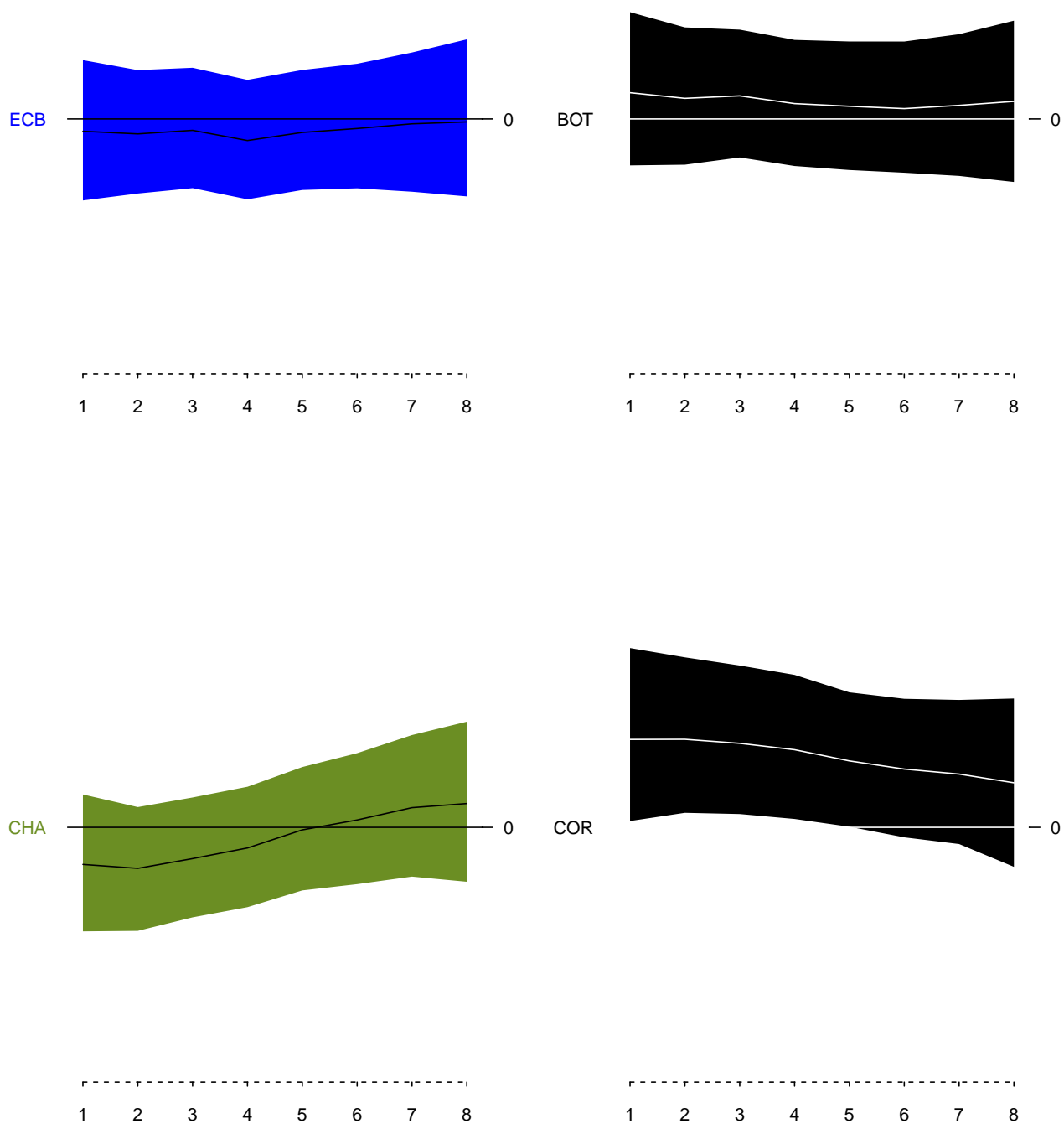


Figura A.51: Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 2.

EBs – Modelo Dinâmico

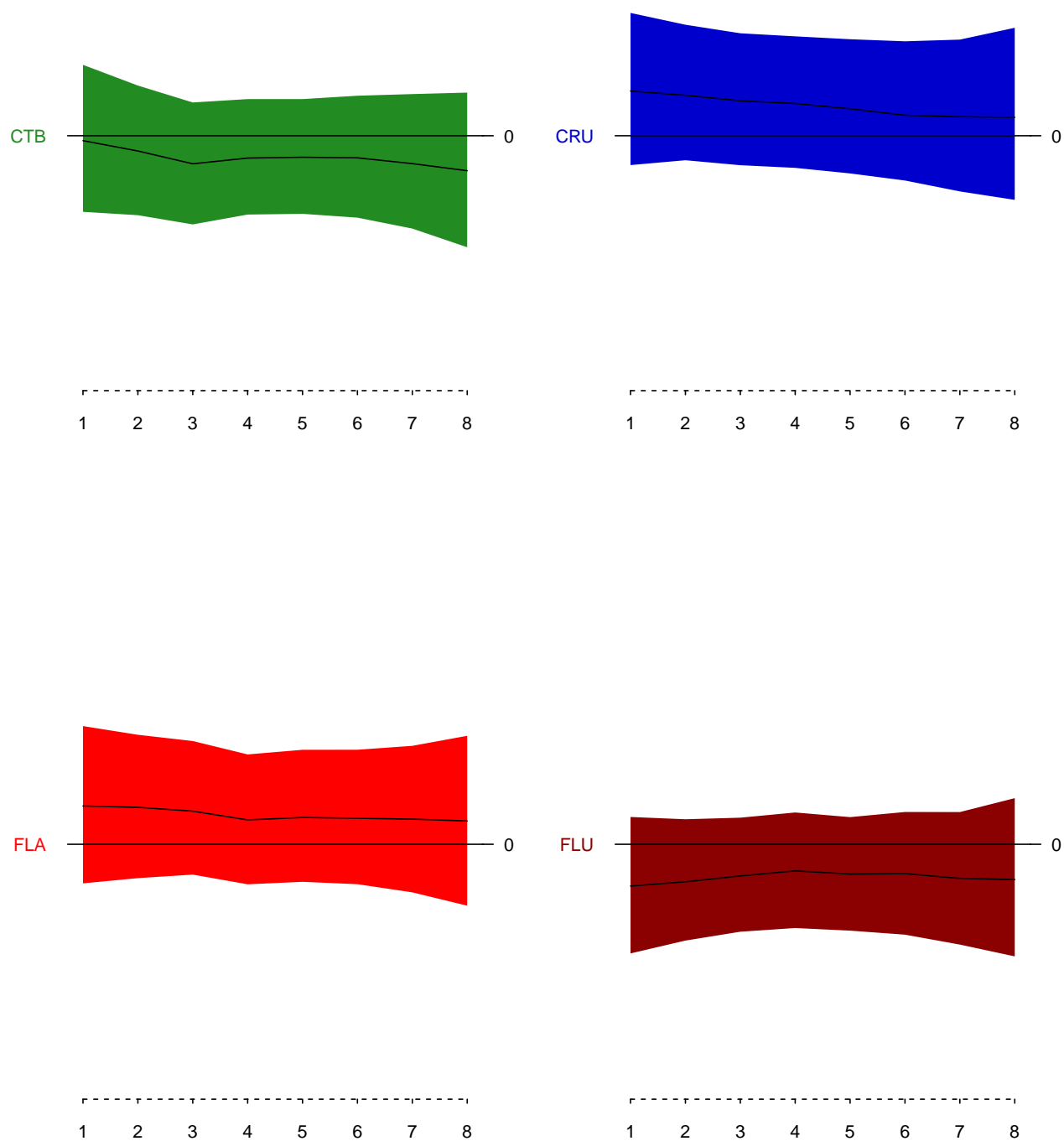


Figura A.52: Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 3.

EBs – Modelo Dinâmico

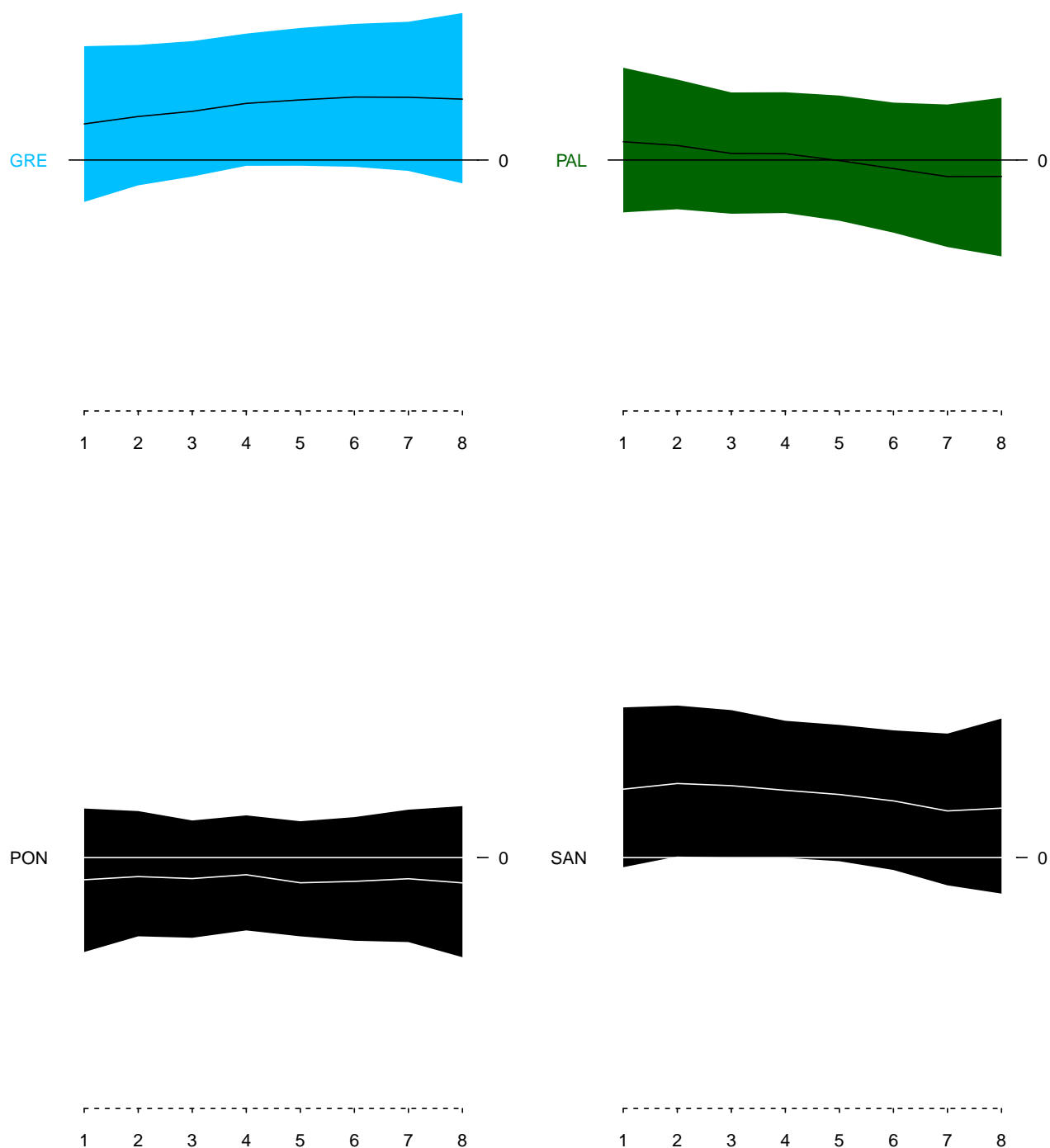


Figura A.53: Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 4.

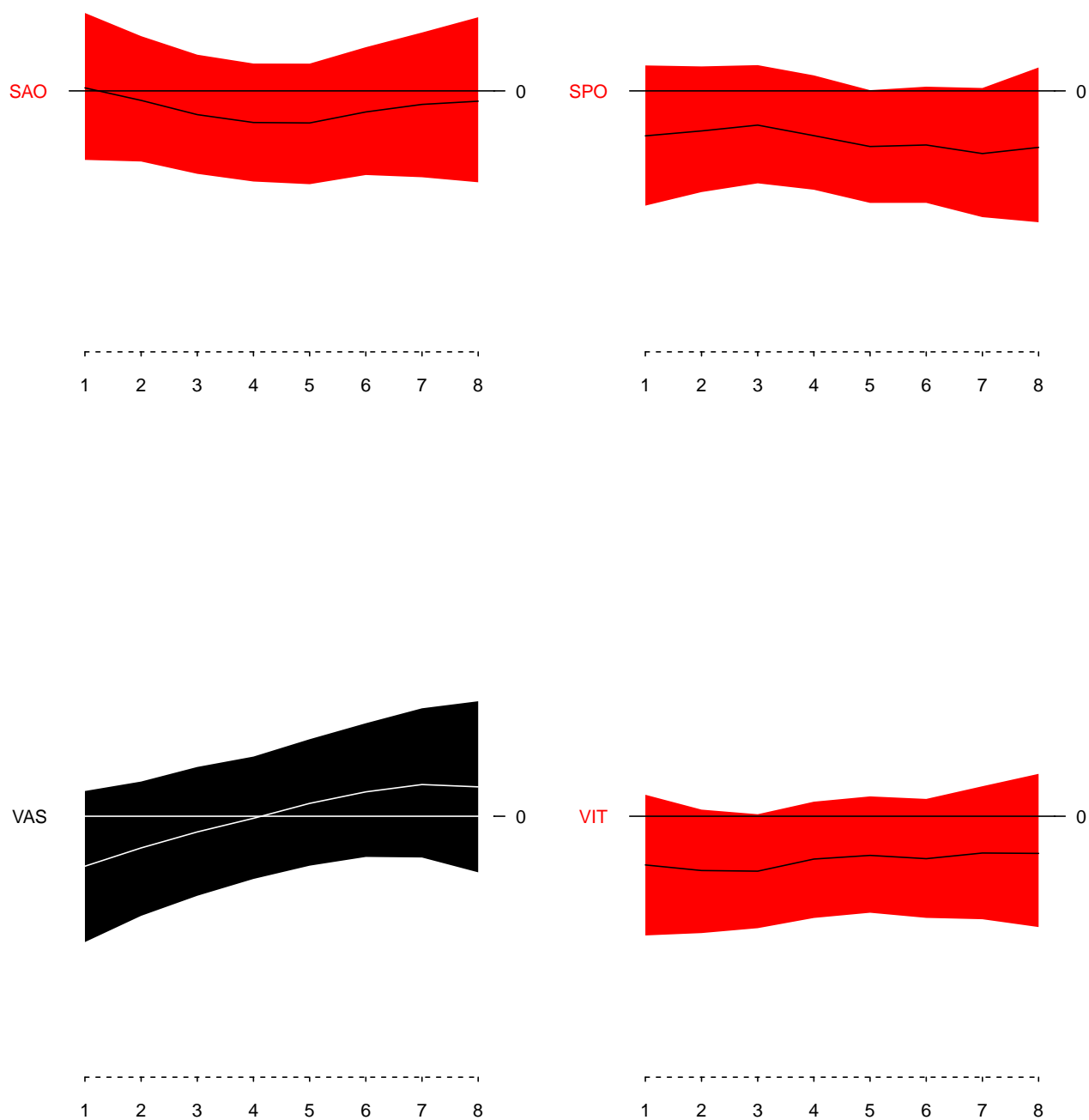


Figura A.54: Estimativas do fator defesa para o Campeonato brasileiro de 2017 - Parte 5.

As conclusões tiradas sobre as Figuras A.55 a A.57 são as mesmas tiradas sobre as Figuras A.22 a A.24, na seção A.3.2.

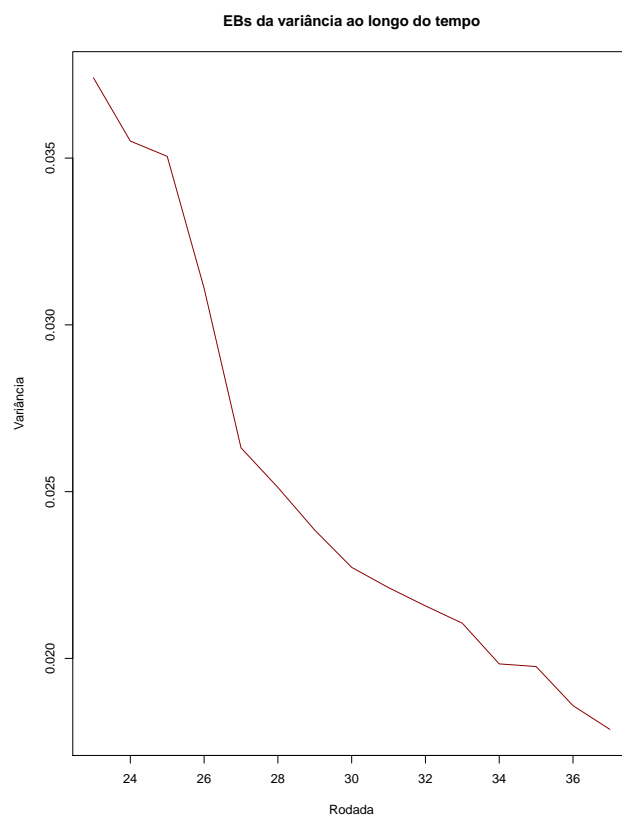


Figura A.55: Estimativas da variância das perturbações para o Campeonato brasileiro de 2017.

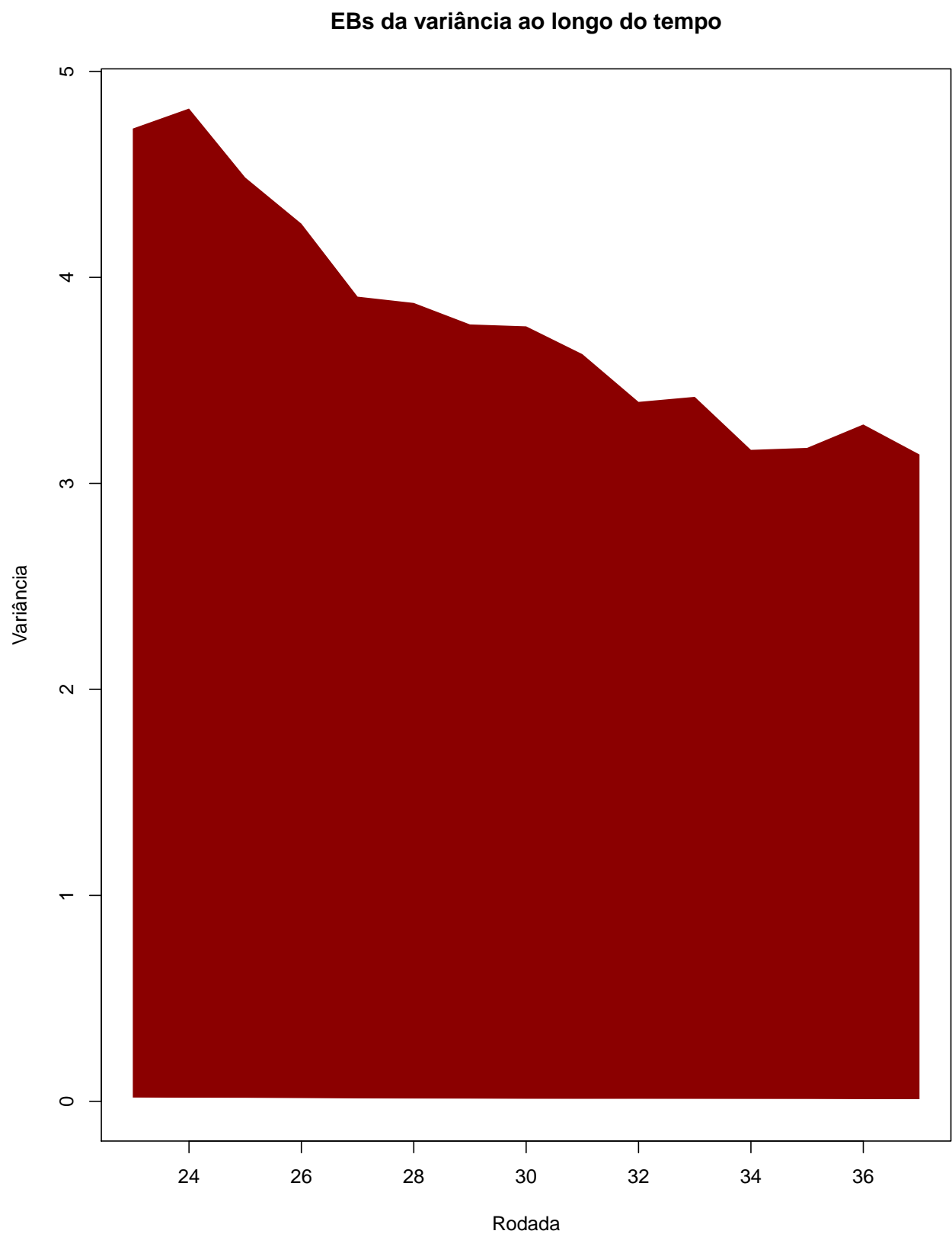


Figura A.56: Estimativas da variância das perturbações para o Campeonato brasileiro de 2017.

Histograma da amostra da posteriori da variância

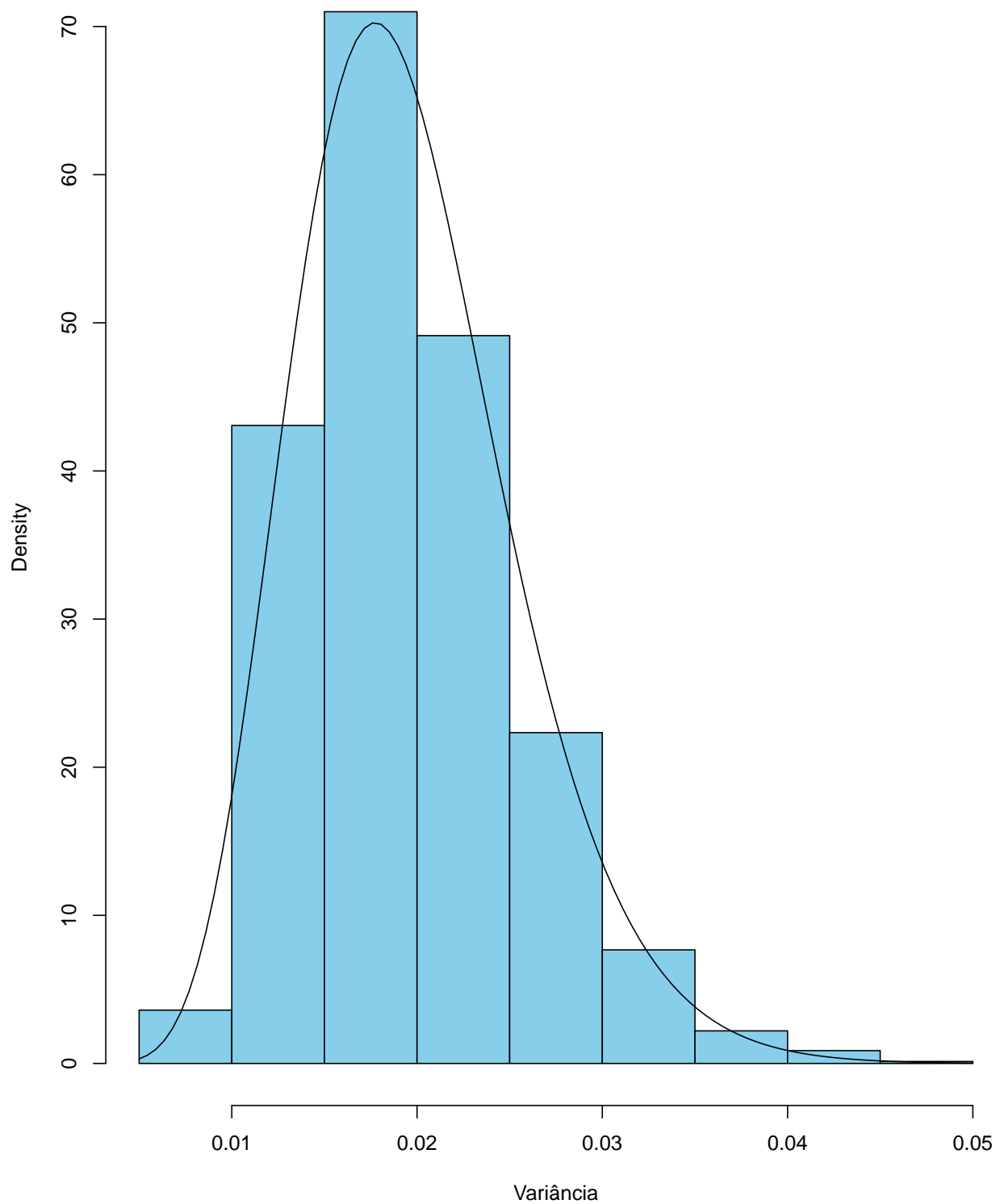


Figura A.57: Estimativas da variância das perturbações para o Campeonato brasileiro de 2017.

A.4.3 Verossimilhanças preditivas para os modelos

Nessa subseção serão mostradas representações gráficas das verossimilhanças preditivas de todos os modelos avaliados, ao longo dos campeonatos, além de tabelas com os valores. Todos os gráficos apresentam os valores das verossimilhanças preditivas de cada modelo, junto com o jogo ou rodada referente ao valor, enquanto cada tabela apresenta os valores para cada modelo, a cada rodada avaliada.

As Figuras A.58 a A.61 contêm os valores das verossimilhanças preditivas padronizadas do modelo estático e dos modelos usados para comparação neste trabalho. Especificamente, as Figuras A.59 e A.61 contêm as verossimilhanças preditivas padronizadas do modelo estático. A Figura A.59 apresenta as verossimilhanças preditivas para cada jogo do Campeonato brasileiro de 2017, enquanto a Figura A.61 apresenta as verossimilhanças do modelo por rodada.

As Figuras referentes ao modelo dinâmico são as Figuras A.62 a A.65. A única diferença dessas Figuras para as do modelo estático é que o modelo dinâmico passa a ser representado nelas. Elas foram geradas com o propósito de permitir comparações entre os modelos estático e dinâmico. Por isso, só foram coletadas previsões para o modelos dinâmico a partir das rodadas em que se considerou que havia ocorrido a estabilização das previsões do modelo estático.

Quando se compara as Figuras A.59 e A.63, não se vê grandes diferenças qualitativas entre as previsões dos modelos estático e dinâmico, por jogo do Campeonato brasileiro de 2017. O mesmo é visto, agora para as previsões por rodada, quando se compara as Figuras A.61 e A.65.

Para o Campeonato brasileiro de 2017, pode-se ver que perto do jogo de número 250 do campeonato, há uma mudança qualitativa nos valores de suas verossimilhanças preditivas entre os jogo, como mostra a Figura A.59. Perto dessa partida, a amplitude do gráfico subitamente diminui e os pontos do gráfico começam a se conservar mais no setor acima da reta negra.

A Figura A.61 confirma essas conclusões, e fica claro que há uma significativa mudança na qualidade das previsões do modelo, a partir da rodada 24. A partir dessa rodada, as verossimilhanças preditivas começam a apresentar valores mais altos.

Uma análise qualitativa das figuras apresentadas combinada com uma análise das Tabelas A.3 e ??, aponta fortemente para uma grande mudança qualitativa nas verossimilhanças preditivas do modelo estático, por volta da rodada 24. Após esse ponto de corte, o modelo estático deixa de ser muito pior que todos os outros modelos e passa a ter

previsões melhores que as dos modelos nulo e da UFMG.

As previsões do modelo estático começaram a ser coletadas apenas a partir da sexta rodada, pois antes disso algumas cadeias de Markov não convergiram, durante o uso de algoritmos MCMC, e as do dinâmico só começaram a ser coletadas a partir da rodada 24 por causa dos diversos possíveis modelos dinâmicos que estavam sendo comparados, como apresentado na seção [A.2](#), o que tornava muito custoso computacionalmente a coleta de previsões de cada modelo dinâmico para um conjunto de 33 rodadas.

A divisão das previsões baseadas nas rodadas seis e 24 mostra a mudança qualitativa de grande magnitude que ocorreu nas previsões do modelo estático, como mostra a Tabela [A.4](#), na qual os valores das verossimilhanças preditivas do modelo estático para a primeira parte são muito piores que as do modelo nulo - o que apontaria, para um estatístico, a necessidade de descartar o modelo estático - enquanto as da segunda parte são melhores que as do modelo nulo. Já as previsões do modelo dinâmico são ainda melhores, sendo as terceiras melhores do conjunto de modelos.

VPPs por Jogo

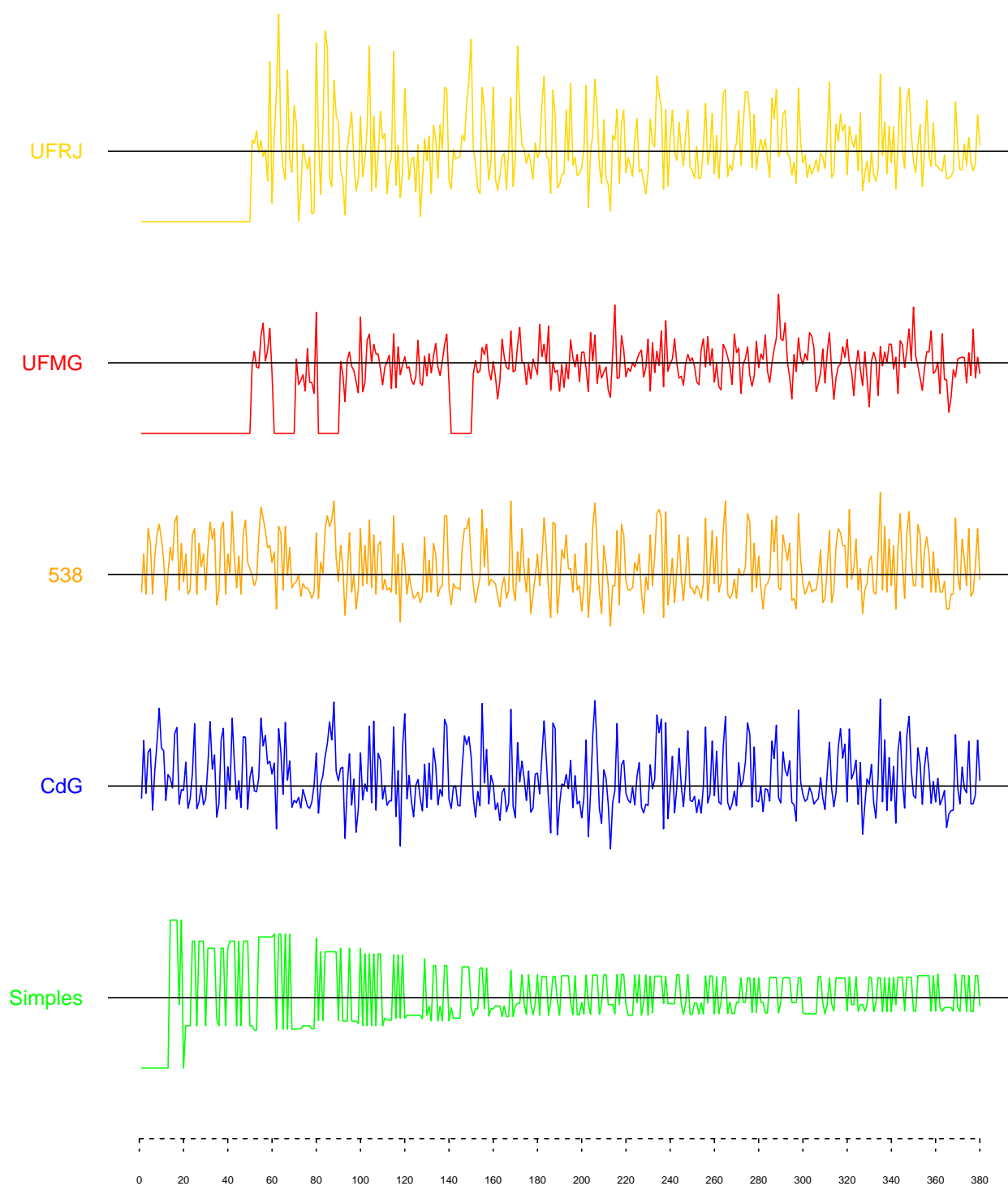


Figura A.58: Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2017.

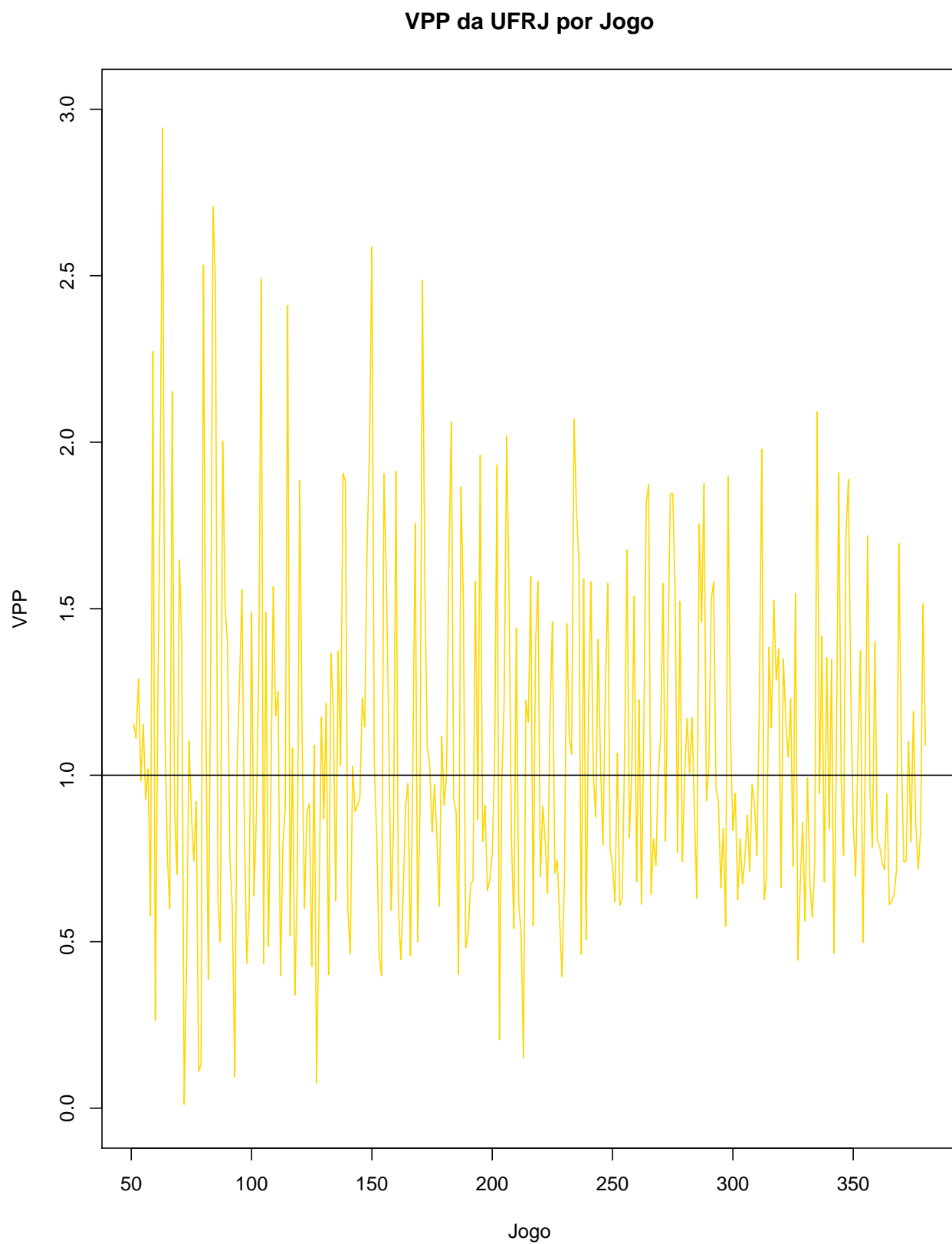


Figura A.59: Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2017.

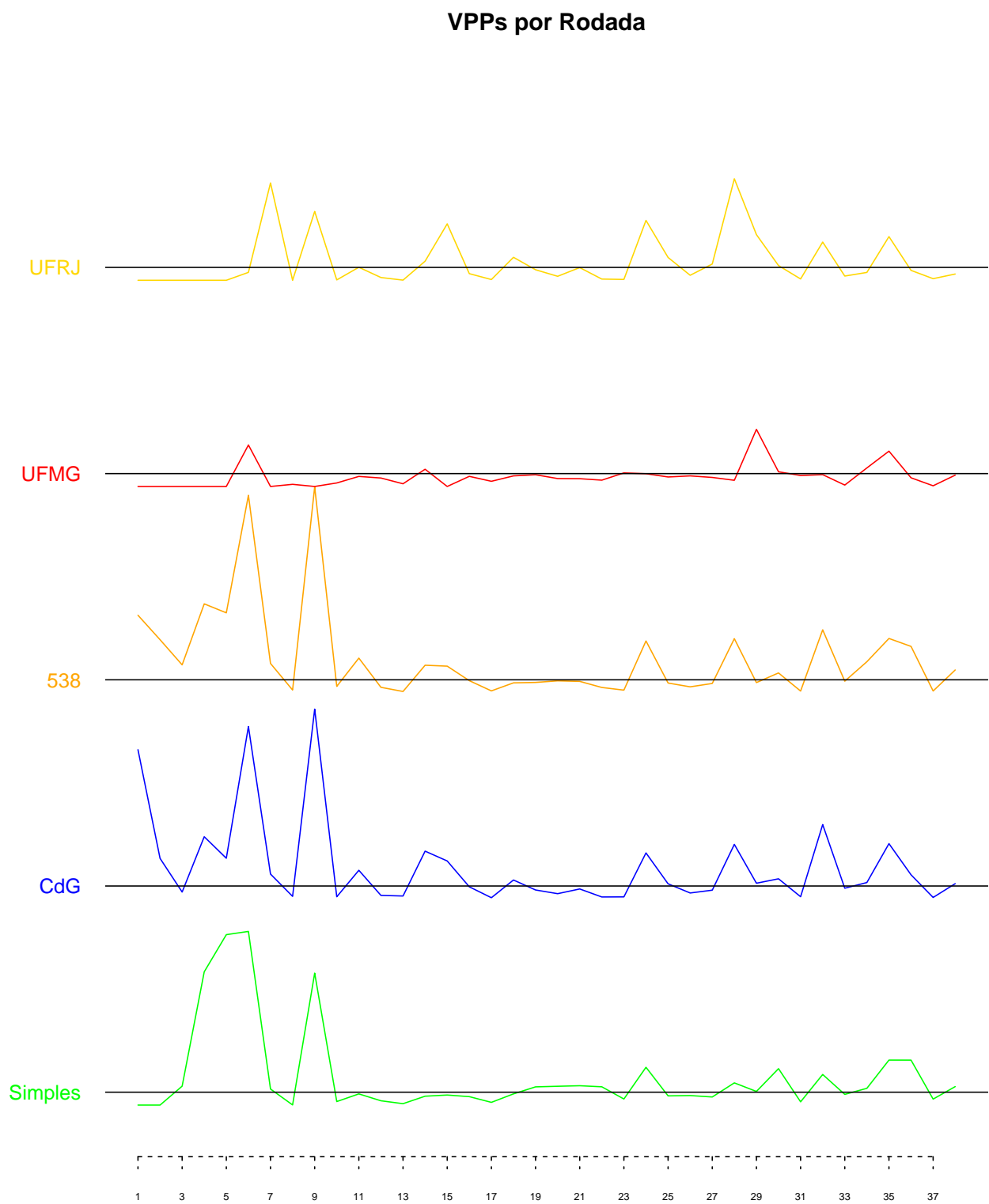


Figura A.60: Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2017.

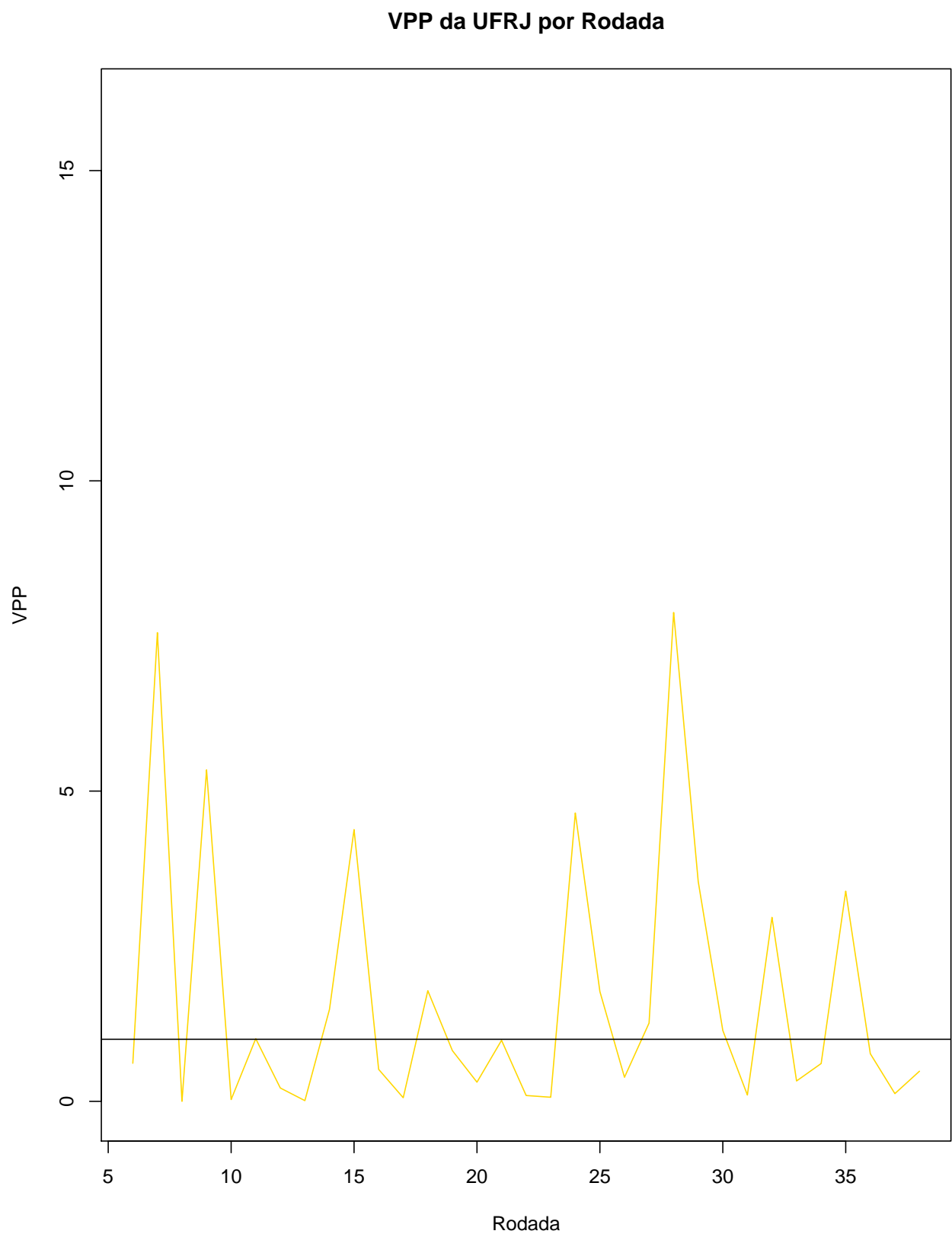


Figura A.61: Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2017. 142

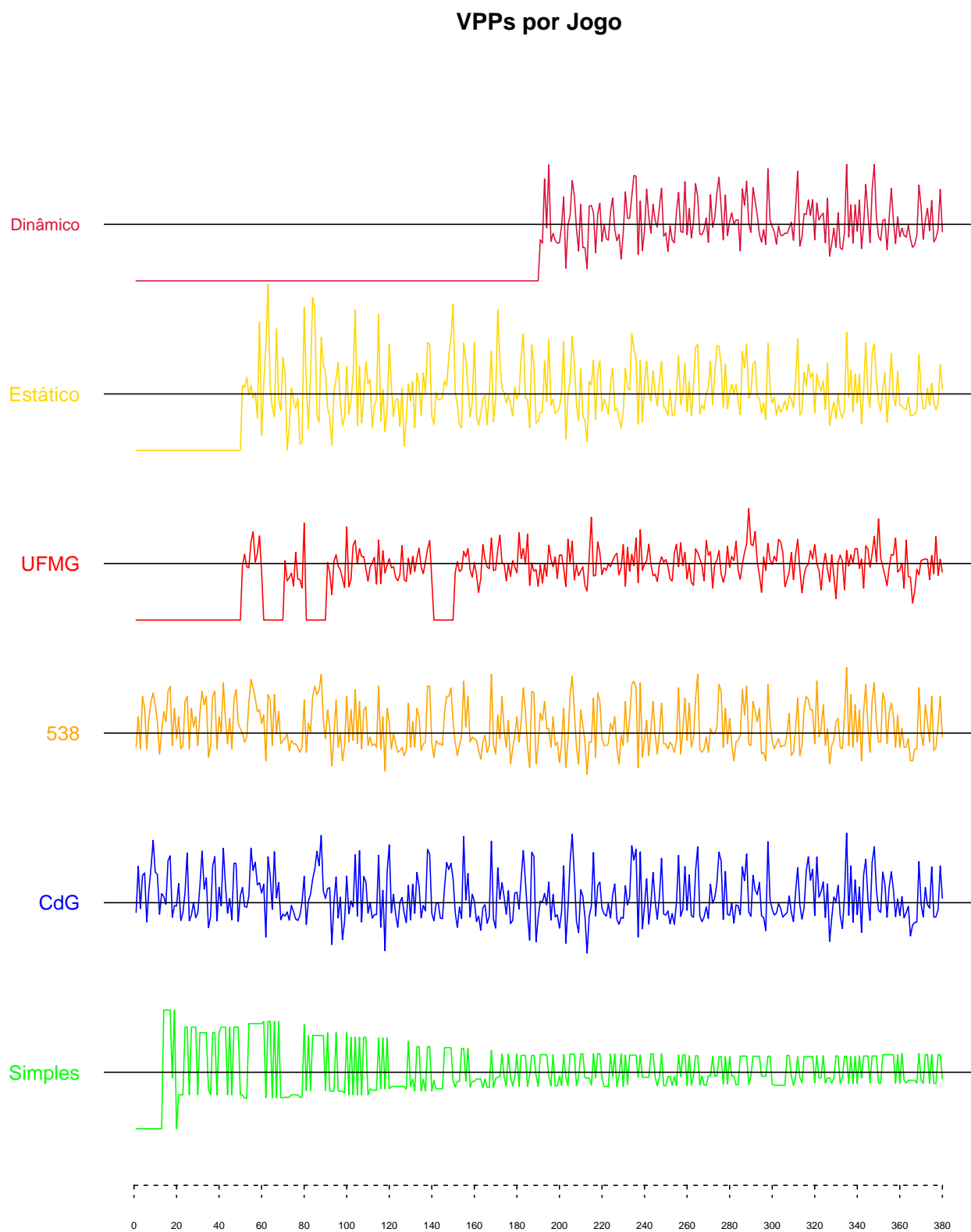


Figura A.62: Verossimilhanças preditivas padronizadas por jogo, para o Campeonato brasileiro de 2017.

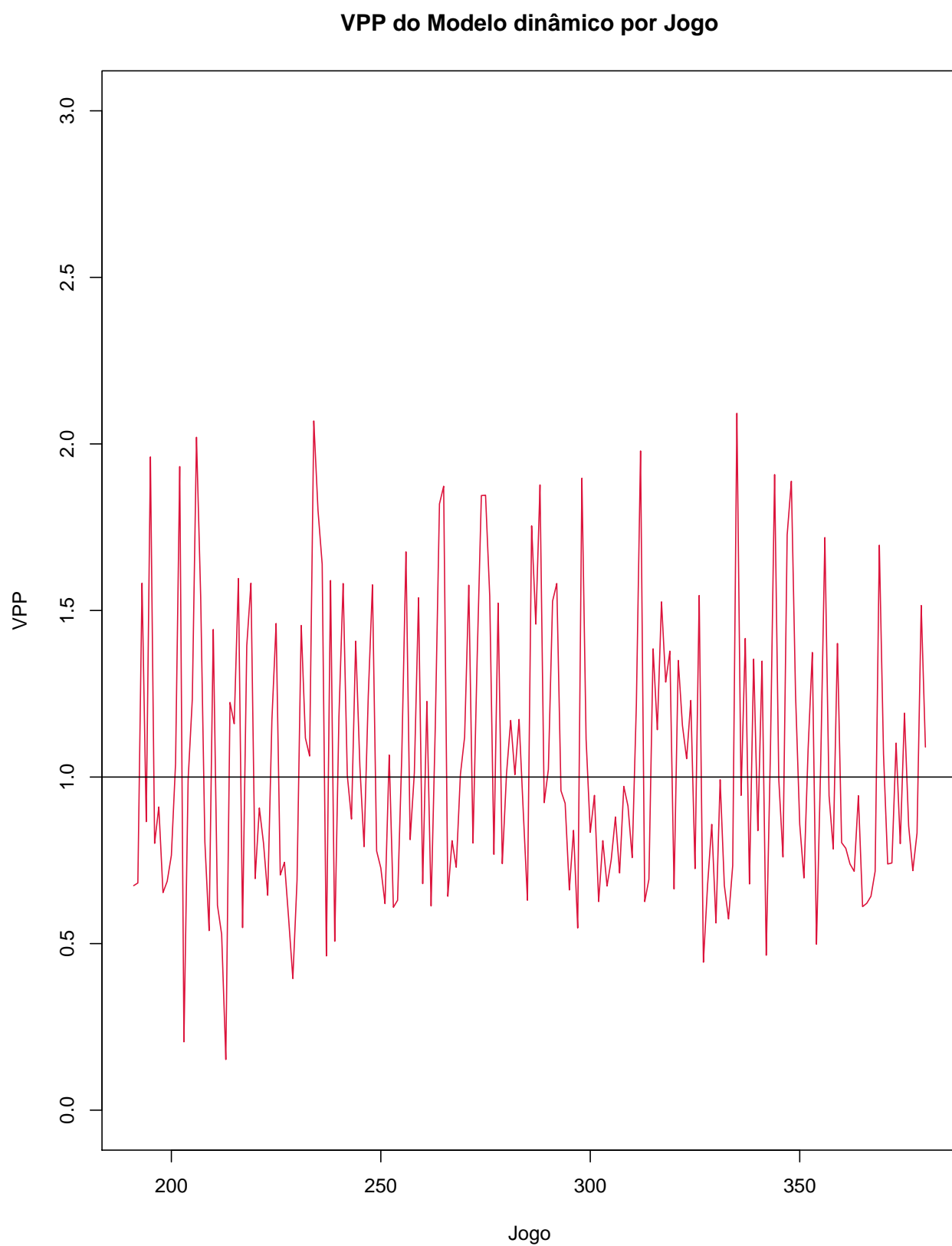


Figura A.63: Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato brasileiro de 2017.

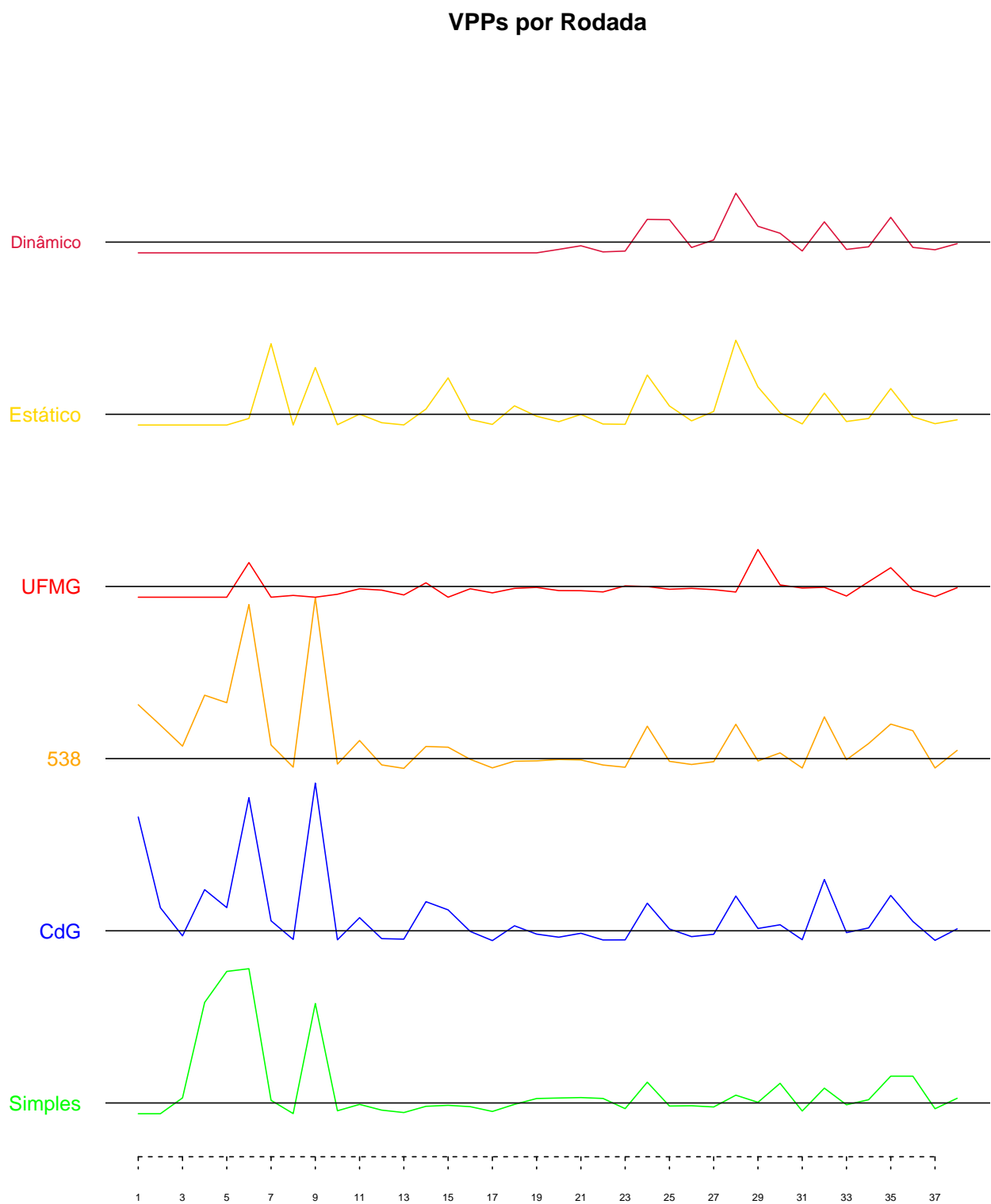


Figura A.64: Verossimilhanças preditivas padronizadas por rodada, para o Campeonato brasileiro de 2017.

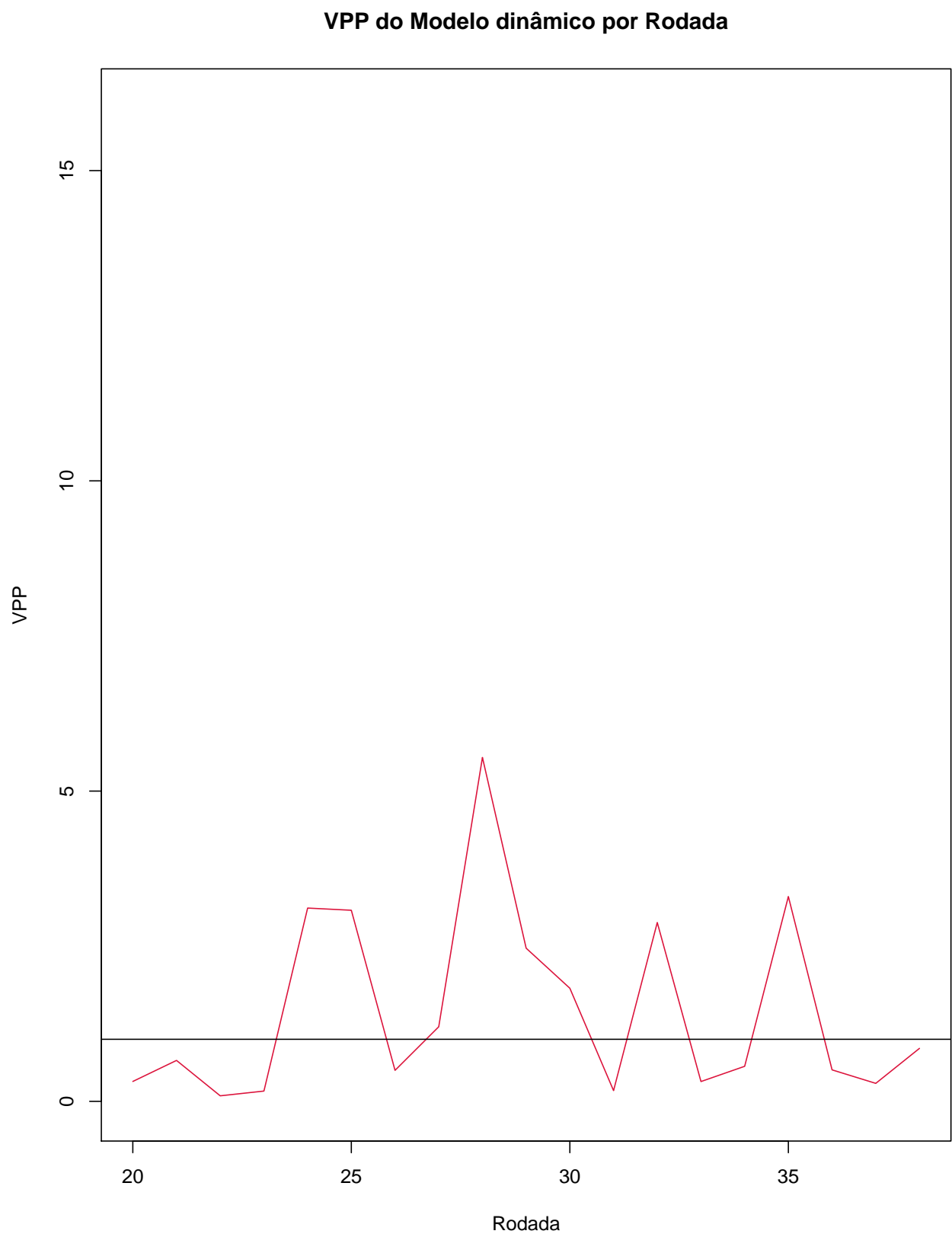


Figura A.65: Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato brasileiro de 2017. 146

Tabela A.3: Verossimilhanças preditivas para o Campeonato brasileiro de 2017.

Rodadas	Dinâmico	Estático	538	CdG	UFMG	Nulo	Simples
6 a 10	0	$1,18 \cdot 10^{-4}$	56,22	12,04	0	1	0,6
11 a 20	0	$2,17 \cdot 10^{-4}$	0,03	0,05	0	1	$4,2 \cdot 10^{-3}$
21 a 30	0	0,78	0,34	0,06	0,32	1	5,07
31 a 38	0,04	0,01	4,64	1,07	0,01	1	5,26
Total	0,04	$1,18 \cdot 10^{-10}$	3	0,04	$3,2 \cdot 10^{-3}$	1	0,07

Tabela A.4: Verossimilhanças preditivas para o Campeonato brasileiro de 2017.

Rodadas	Dinâmico	Estático	538	CdG	UFMG	Nulo	Simples
6 a 23	0	$1,55 \cdot 10^{-10}$	0,13	$3,7 \cdot 10^{-3}$	0	1	$2,54 \cdot 10^{-3}$
24 a 38	5,33	1,22	22,81	11,08	0,01	1	26,49

As investigações feitas na seção [A.3.3](#) sobre as posteriores dos parâmetros do modelo estático também foram feitas usando dados do Campeonato brasileiro de 2017. As Figuras [A.66](#) a [A.69](#) são análogas às [A.33](#) a [A.36](#), tendo mudado apenas o campeonato estudado e os times avaliados.

As conclusões, contudo, continuam sendo as mesmas: depois do ponto de estabilização das previsões, a dispersão das amostras para de diminuir significativamente e as prioris escolhidas parecem ter sido vagas demais.

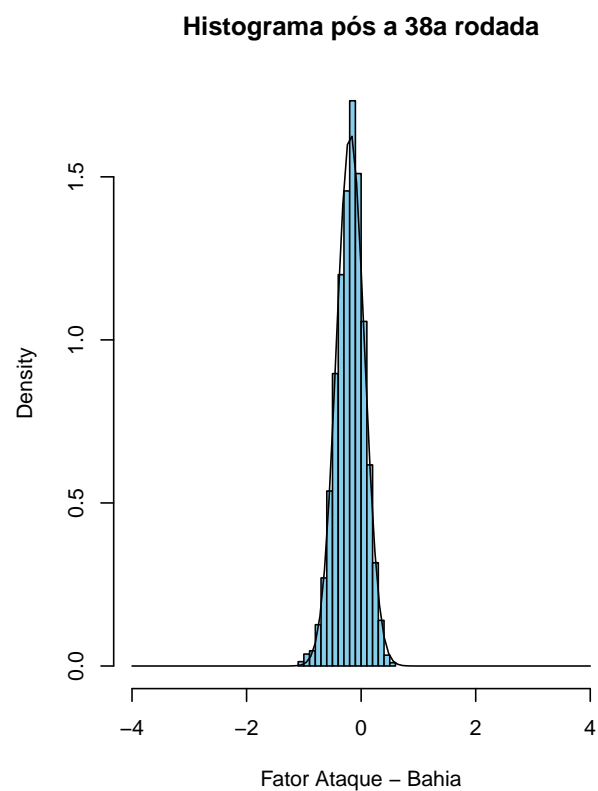
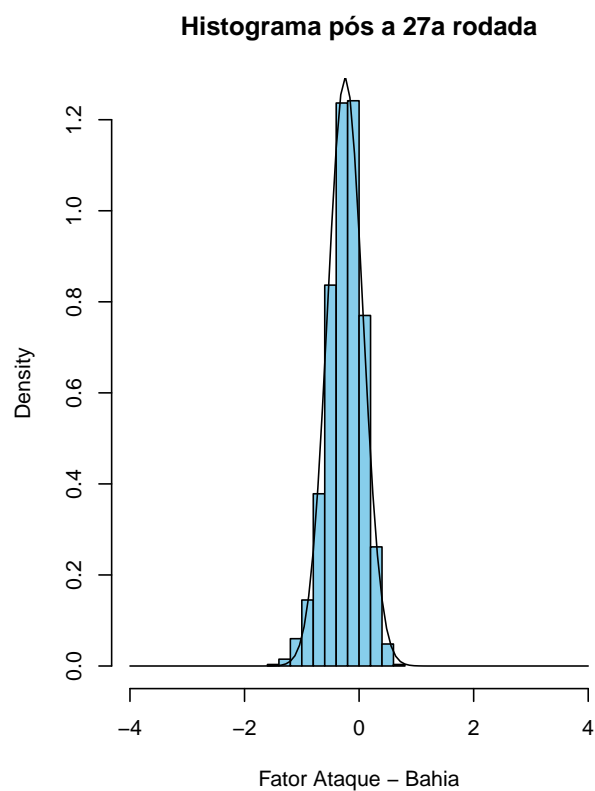
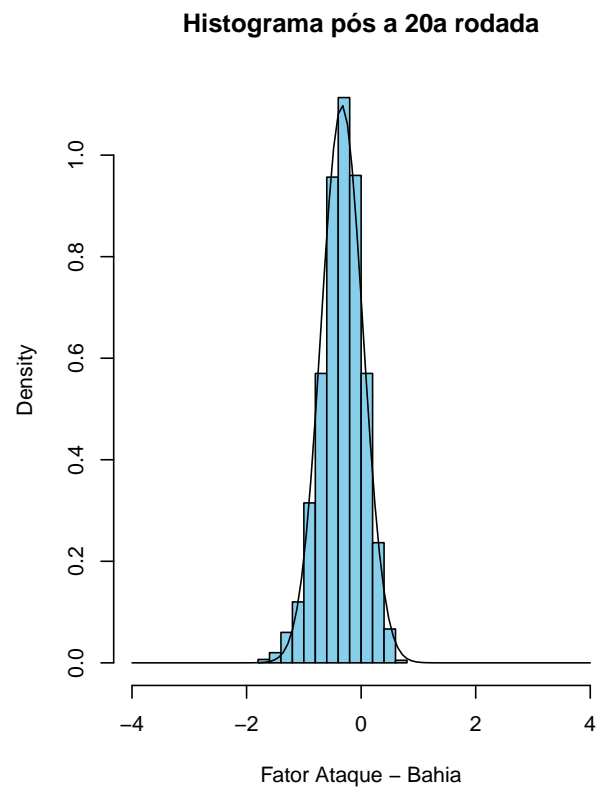
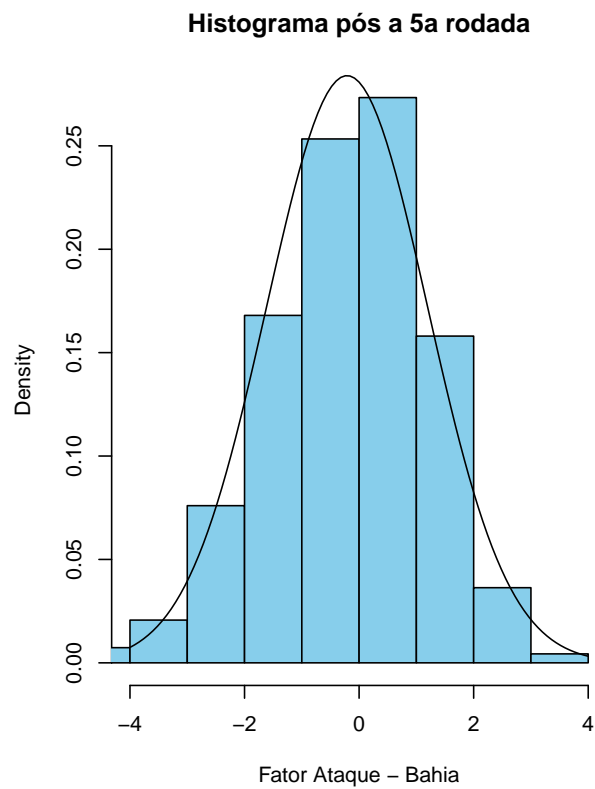


Figura A.66: Amostras aproximadas da posteriori do fator de ataque do Bahia.

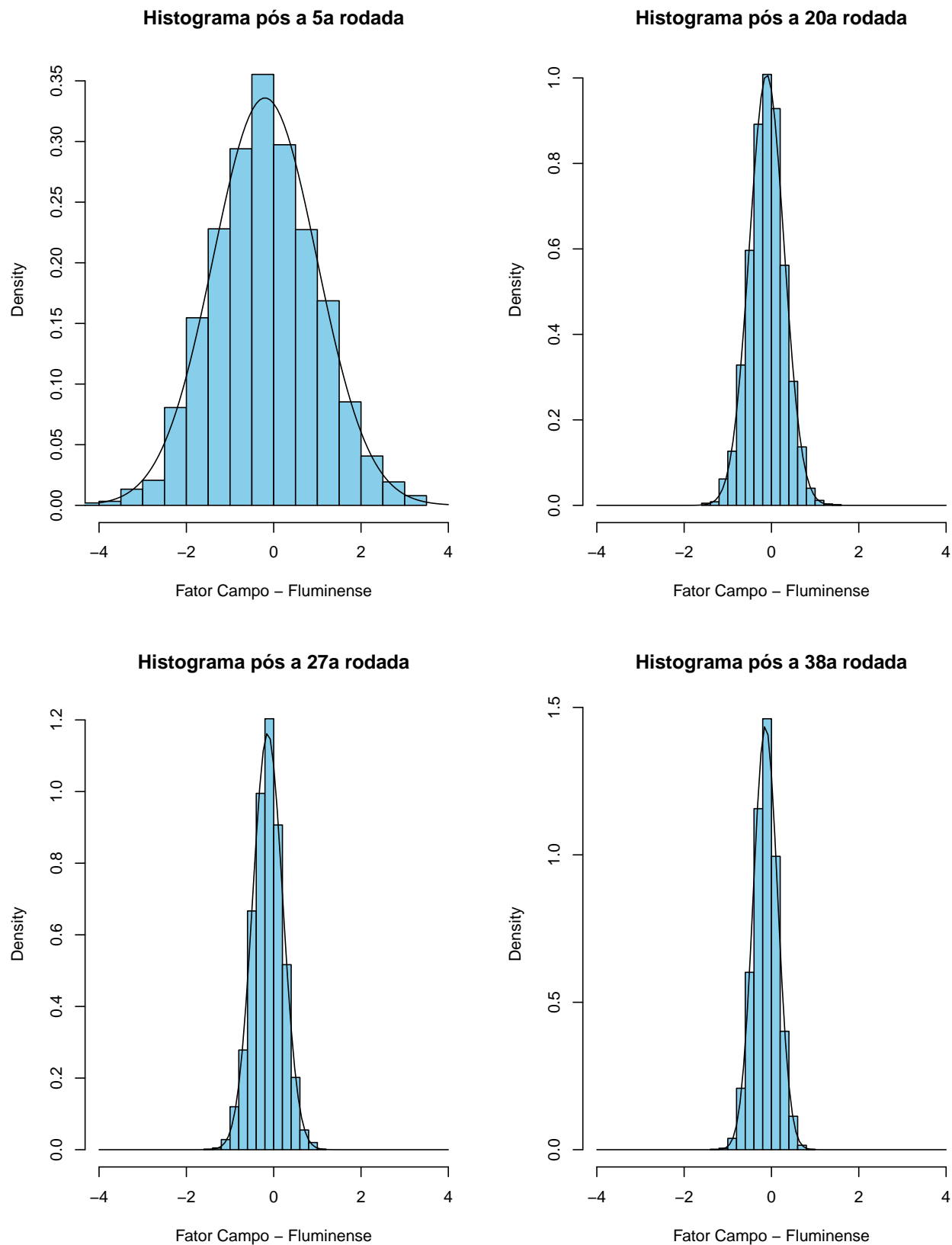


Figura A.67: Amostras aproximadas da posteriori do fator campo do Fluminense.

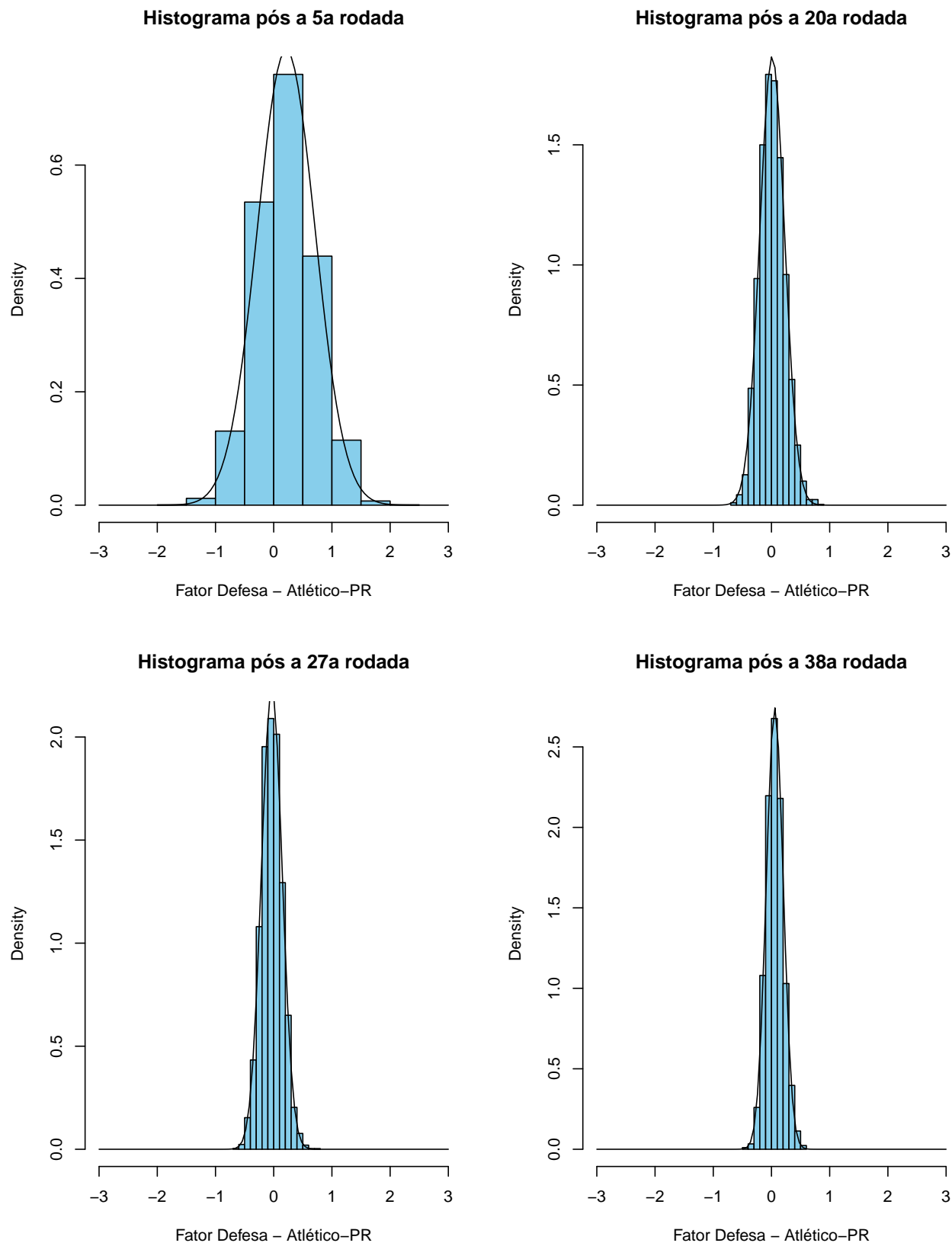


Figura A.68: Amostras aproximadas da posteriori do fator de defesa do Atlético-PR.

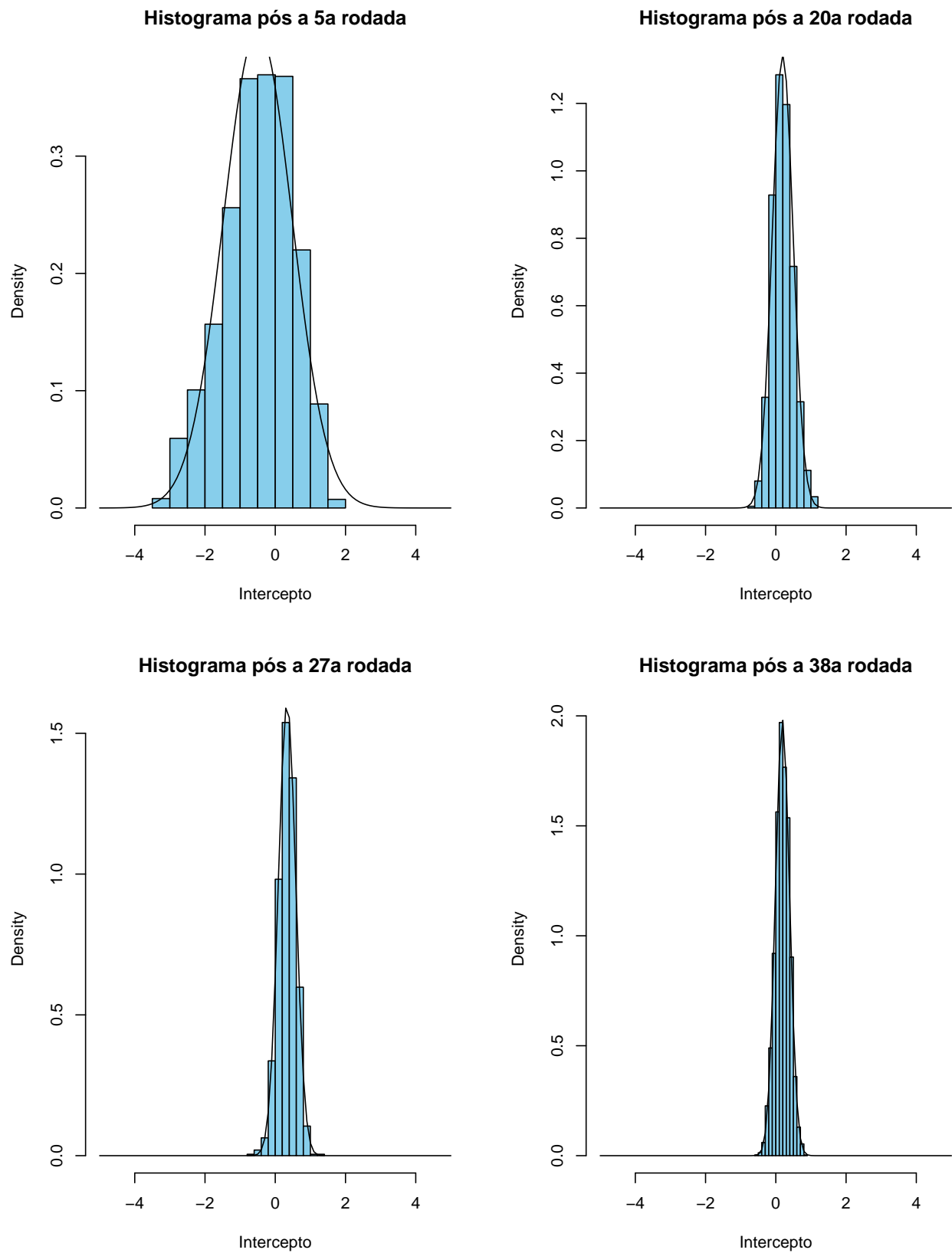


Figura A.69: Amostras aproximadas da posteriori do intercepto do Campeonato brasileiro de 2017.

A.5 Campeonato inglês de 2016/2017

Nesta seção, o campeonato discutido é o campeonato inglês da temporada de 2016/2017. e todos os resultados apresentados e comentados são a ele referentes. Esses resultados estão divididos de acordo com o modelo de que vieram, com a subseção [A.5.1](#) sendo reservada para o modelo estático e a [A.5.2](#) para o modelo dinâmico. Finalmente, esses dois modelos são comparados com os demais apresentados ao longo do trabalho na subseção [A.5.3](#), através de suas verossimilhanças preditivas.

A.5.1 Análises e resultados do Modelo estático

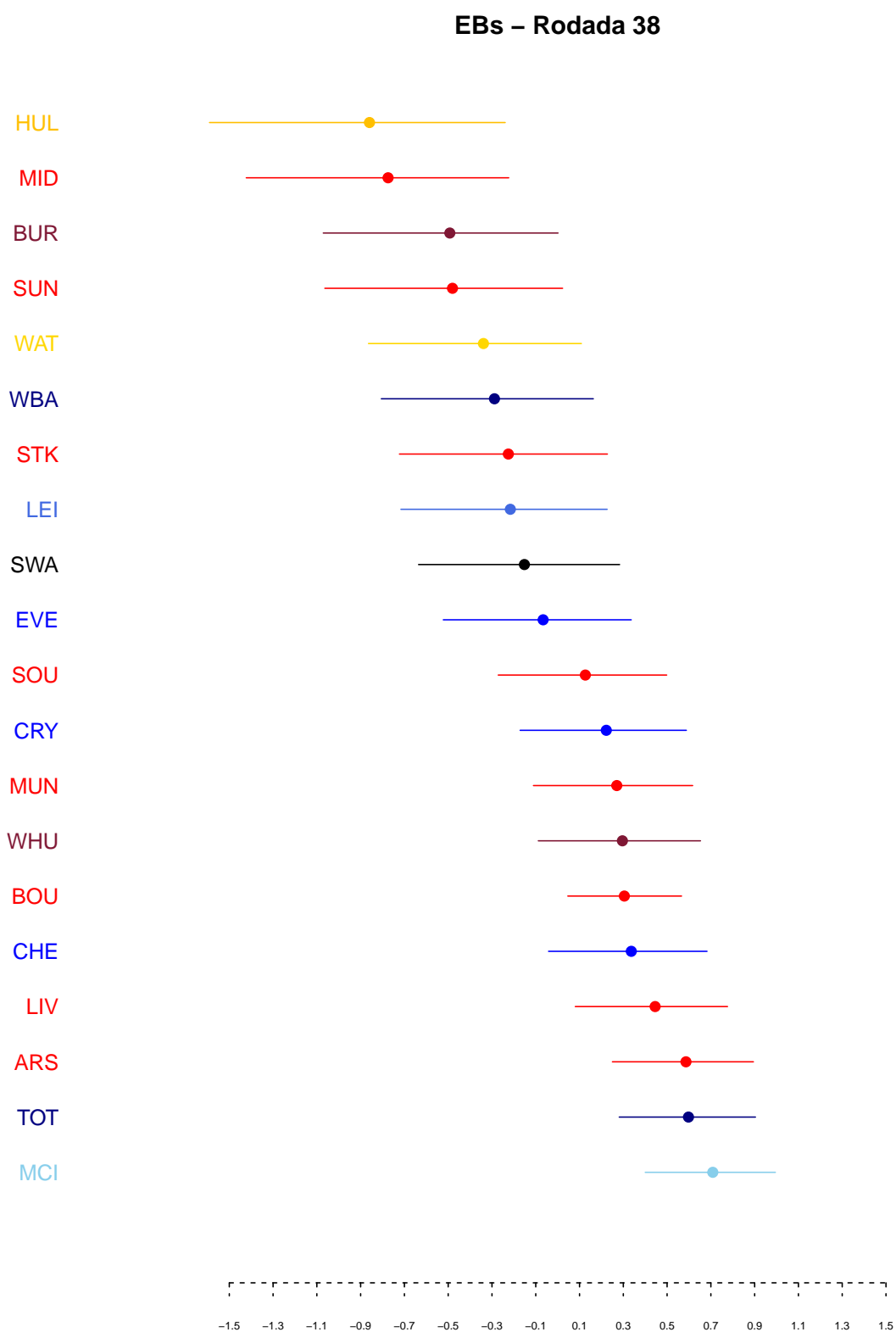


Figura A.70: Estimativas do fator ataque para o Campeonato inglês 2016/17.

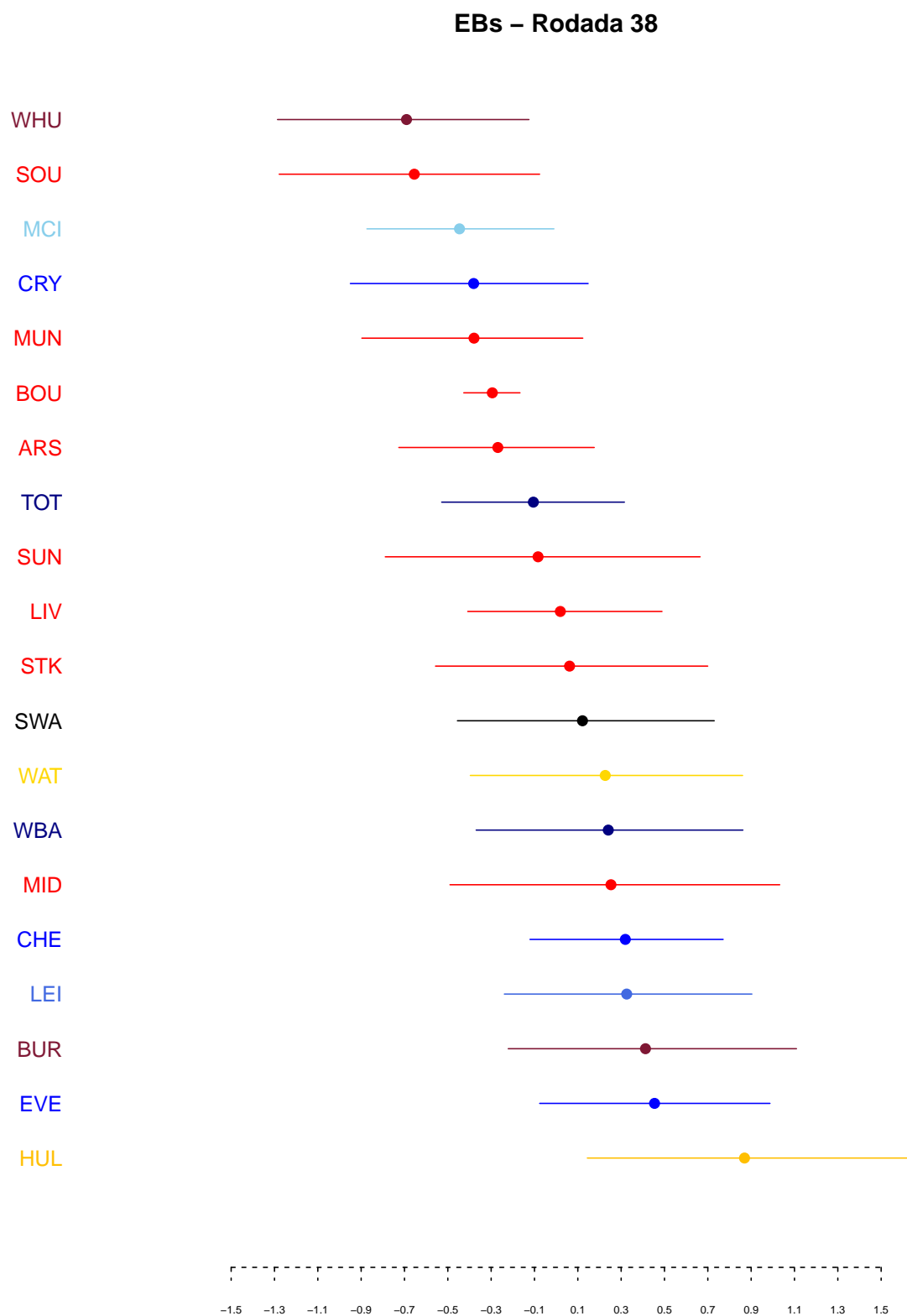


Figura A.71: Estimativas do fator campo para o Campeonato inglês 2016/17.

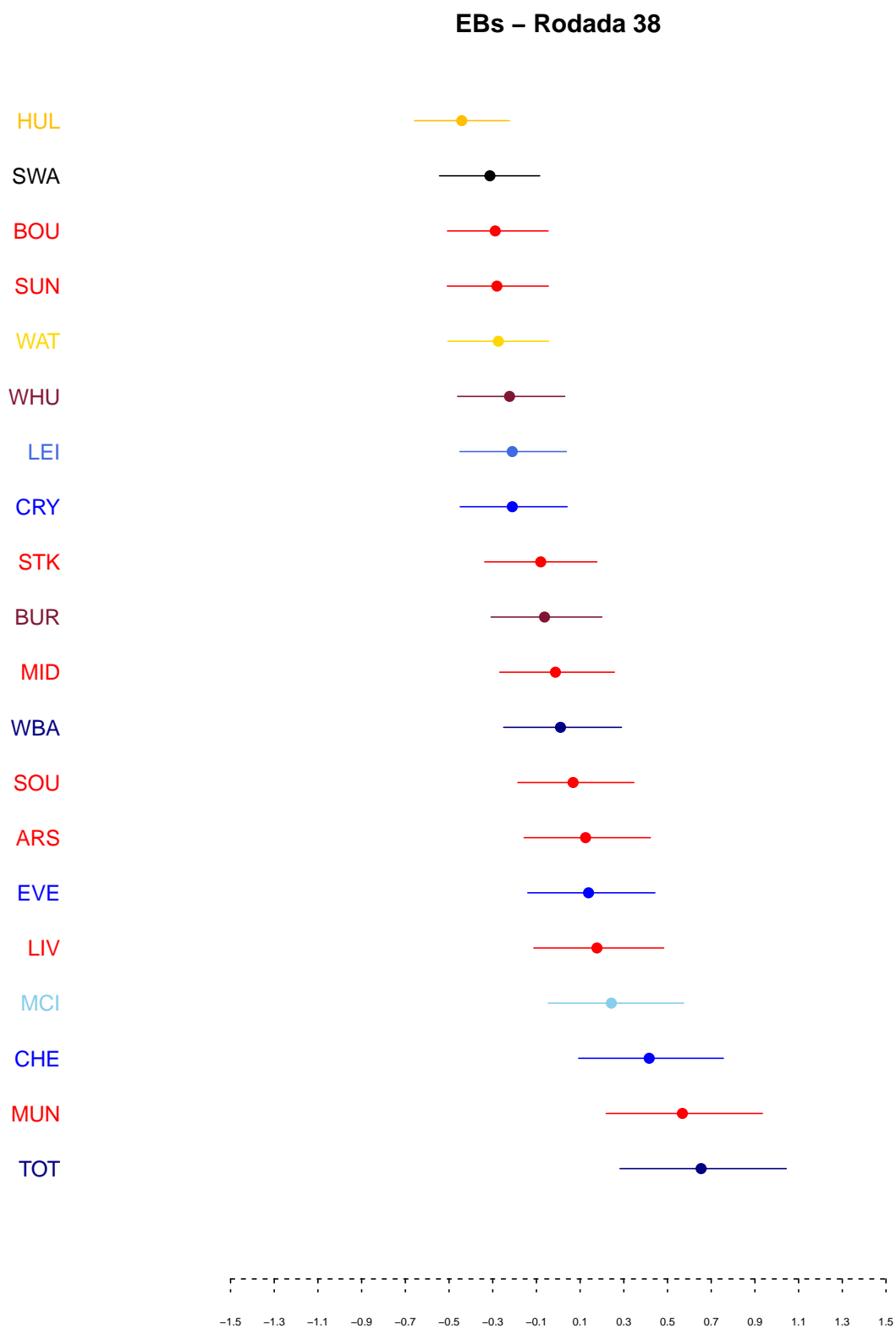


Figura A.72: Estimativas do fator defesa para o Campeonato inglês 2016/17.

Para o Campeonato inglês, as Figuras [A.70](#) a [A.72](#) se relacionam à competição, e as siglas utilizadas estão na seção [A.1](#).

Pode-se ver a partir das Figuras que o Hull City só não teve um rendimento pior por causa de seu excelente mando de campo, já que a equipe teve o pior ataque e a pior defesa da competição. Ainda assim, isso não foi suficiente para impedir o rebaixamento do time. Já o Sunderland foi rebaixado tendo tido desempenho ruim nas três fases medidas pelo modelo. Finalmente, o Middlesbrough pode culpar seu ataque, quando fora de casa, por ter sido rebaixado.

Também é difícil entender, a partir dessas estimativas, o porquê de o Chelsea ter sido o campeão com tantos pontos a mais que o Tottenham, visto que o vice-campeão teve o segundo melhor ataque e a melhor defesa do torneio. A diferença entre as equipes foi construída com base na capacidade do Chelsea de vencer em casa, enquanto o mando de campo do Tottenham ficou entre os mais fracos do campeonato.

Southampton e Burnley não tiveram desempenho compatível com a qualidade de seu time. O Southampton, oitavo colocado, teve a oitava melhor defesa da competição, além do décimo melhor ataque. O sétimo colocado, Everton, teve o décimo pior ataque e a sexta melhor defesa. No entanto, o Southampton teve o segundo pior mando de campo da competição, enquanto o Everton teve o segundo melhor. E isso parece ter sido o suficiente para o Everton terminar o campeonato 15 pontos a frente do adversário. Já o Burnley pode culpar seu ataque, quando jogando fora de casa, por não ter terminado o torneio numa posição melhor.

A.5.2 Análises e resultados do Modelo dinâmico

Para o Campeonato inglês, as estimativas do modelo dinâmico para seus parâmetros estão contidas nas Figuras [A.73](#) a [A.87](#). As Figuras [A.73](#) a [A.77](#) contêm as estimativas para os fatores de ataque das equipes do campeonato. Já as Figuras [A.78](#) a [A.82](#) contêm as estimativas para os fatores campo, enquanto as estimativas para os fatores de defesa estão nas Figuras [A.83](#) a [A.87](#).

EBs – Modelo Dinâmico

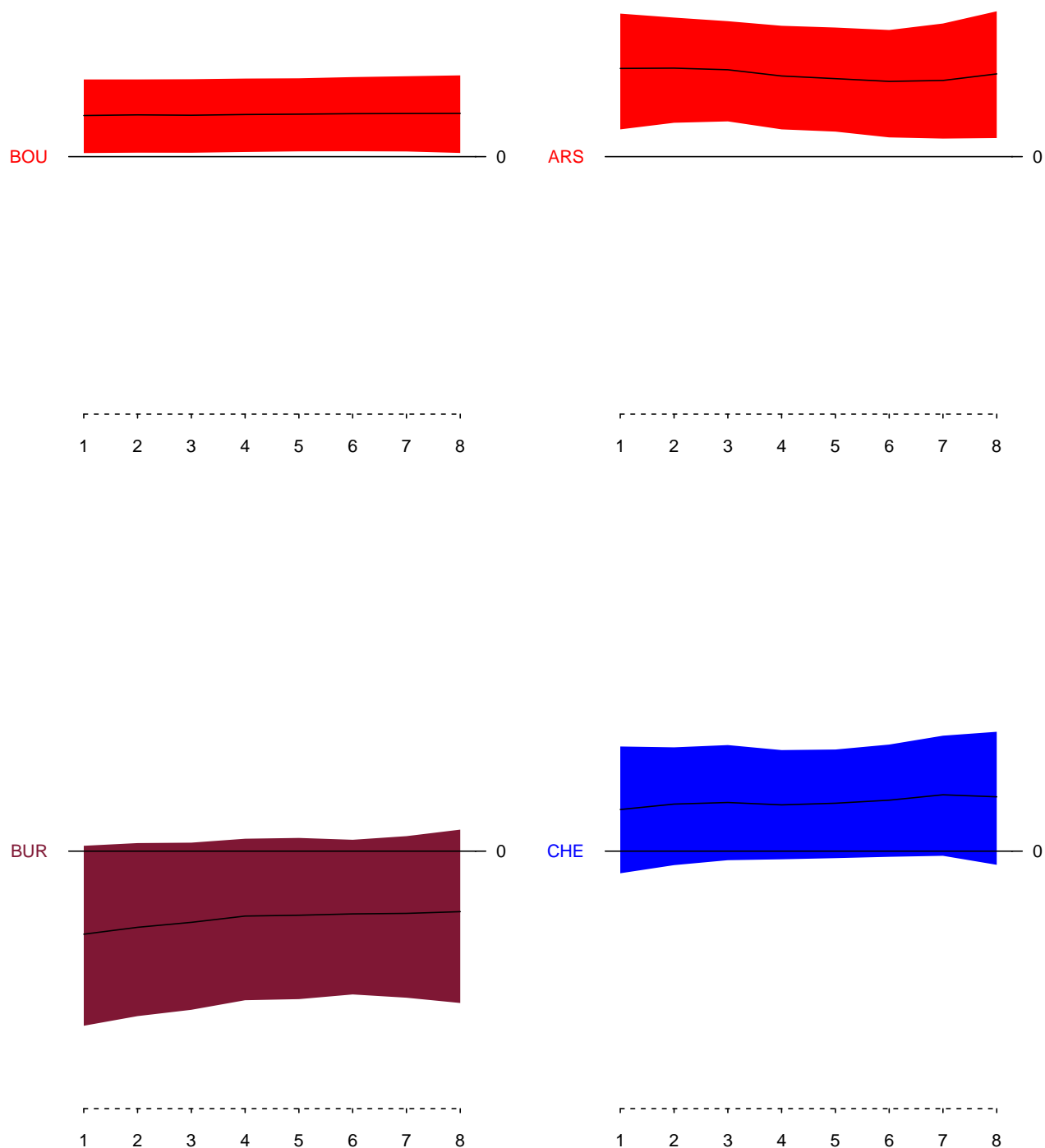


Figura A.73: Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 1.

EBs – Modelo Dinâmico

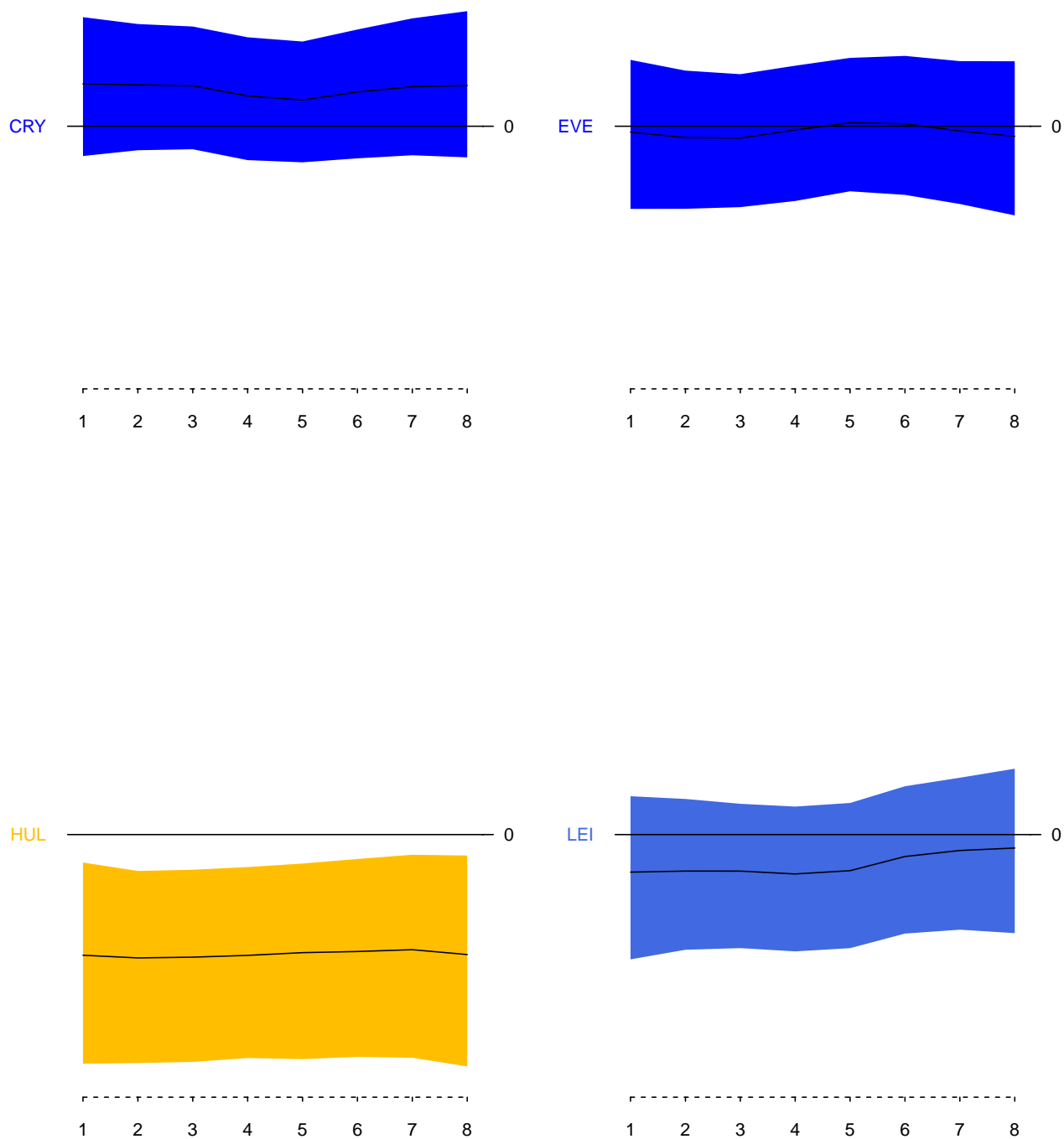


Figura A.74: Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 2.

EBs – Modelo Dinâmico

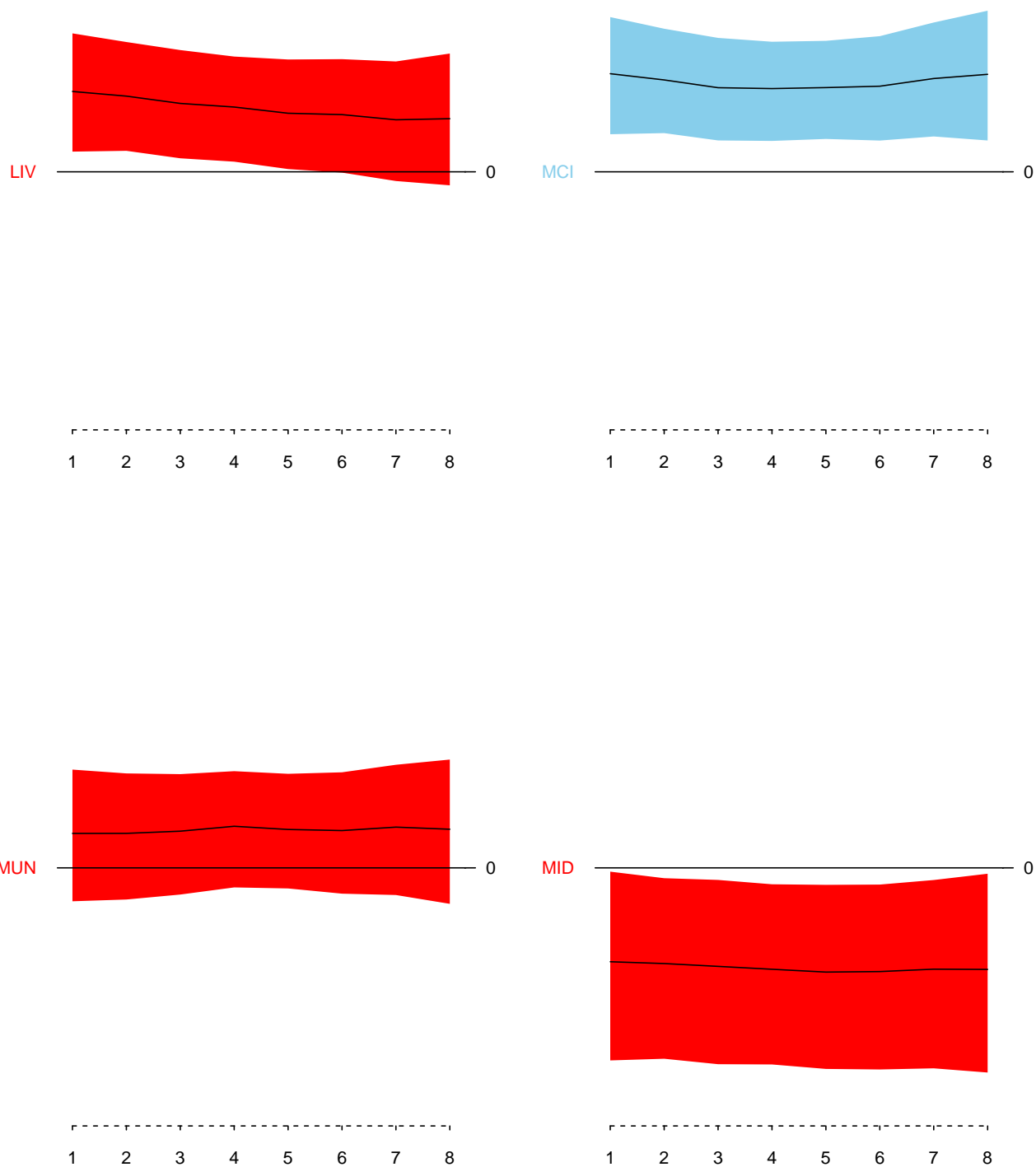


Figura A.75: Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 3.

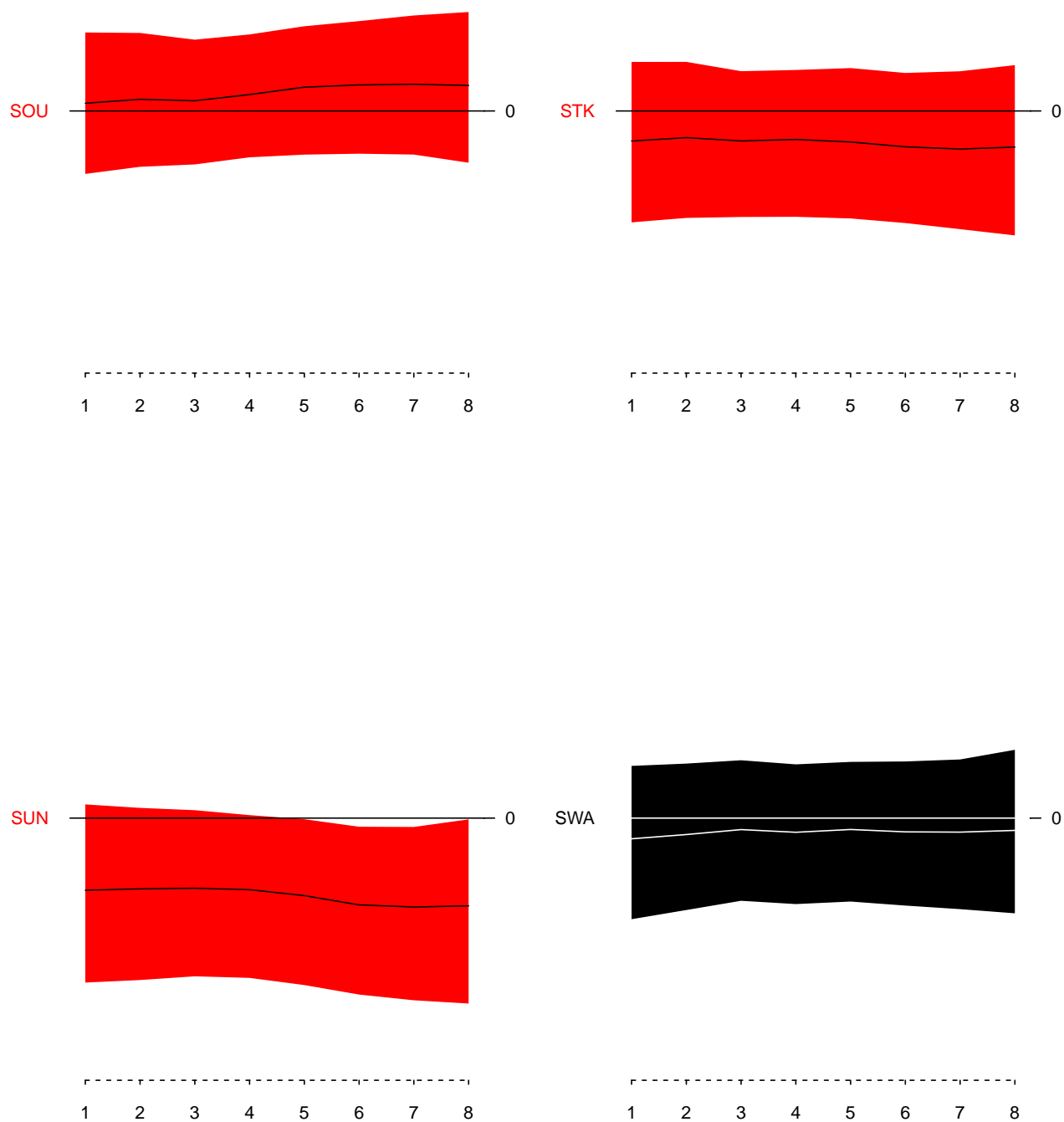


Figura A.76: Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 4.

EBs – Modelo Dinâmico

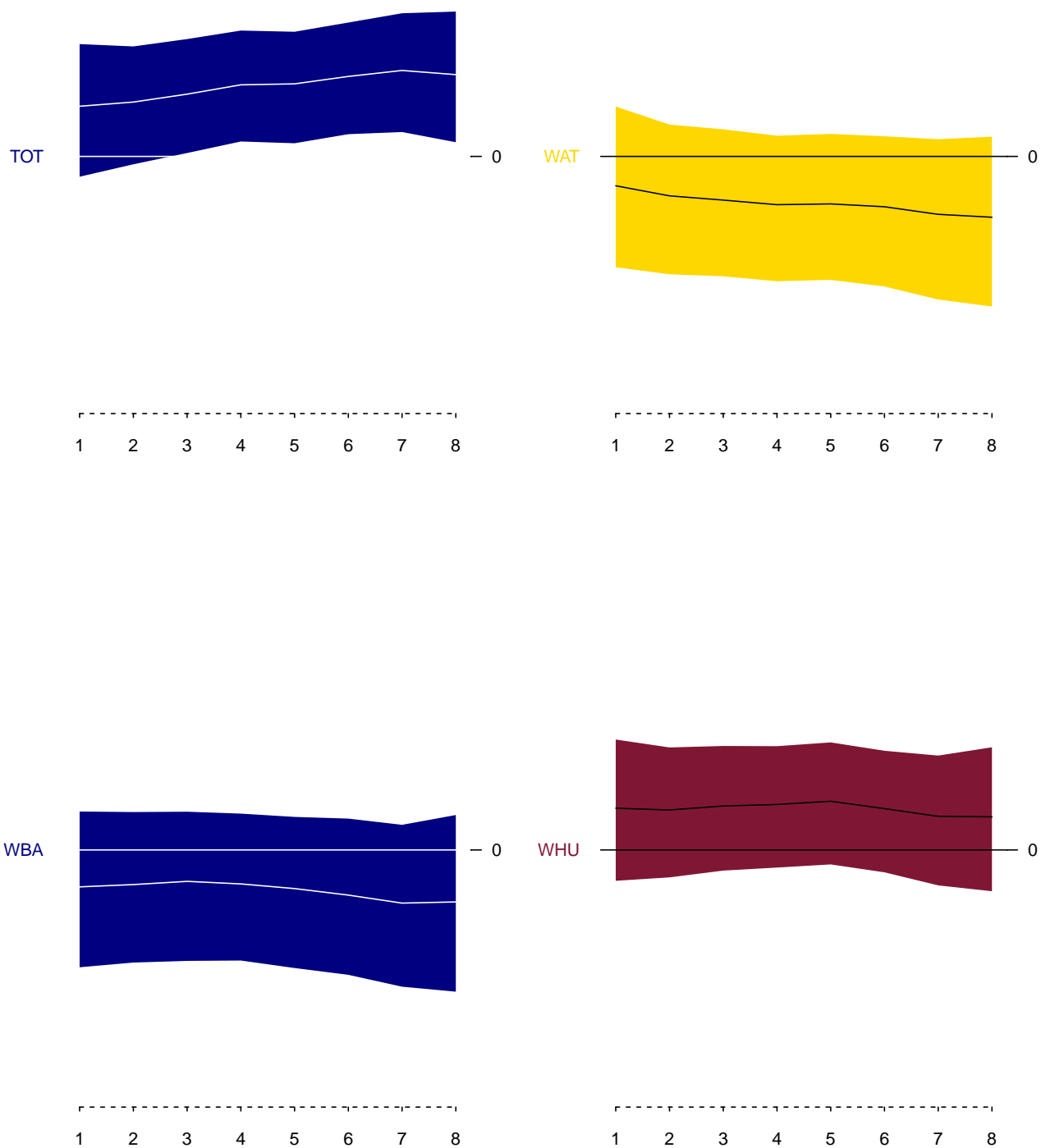


Figura A.77: Estimativas do fator ataque para o Campeonato inglês 2016/17 - Parte 5.

EBs – Modelo Dinâmico

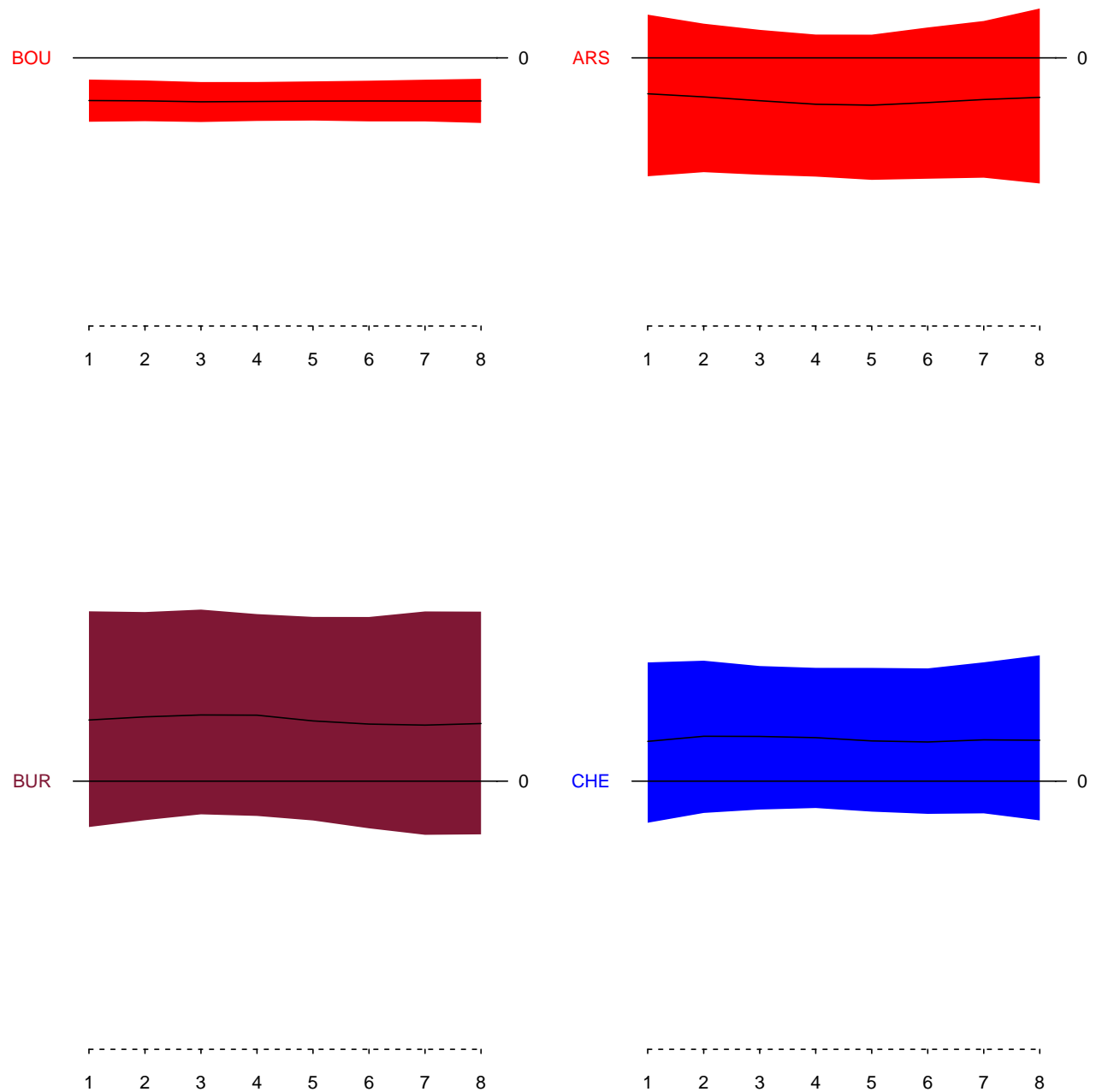


Figura A.78: Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 1.

EBs – Modelo Dinâmico

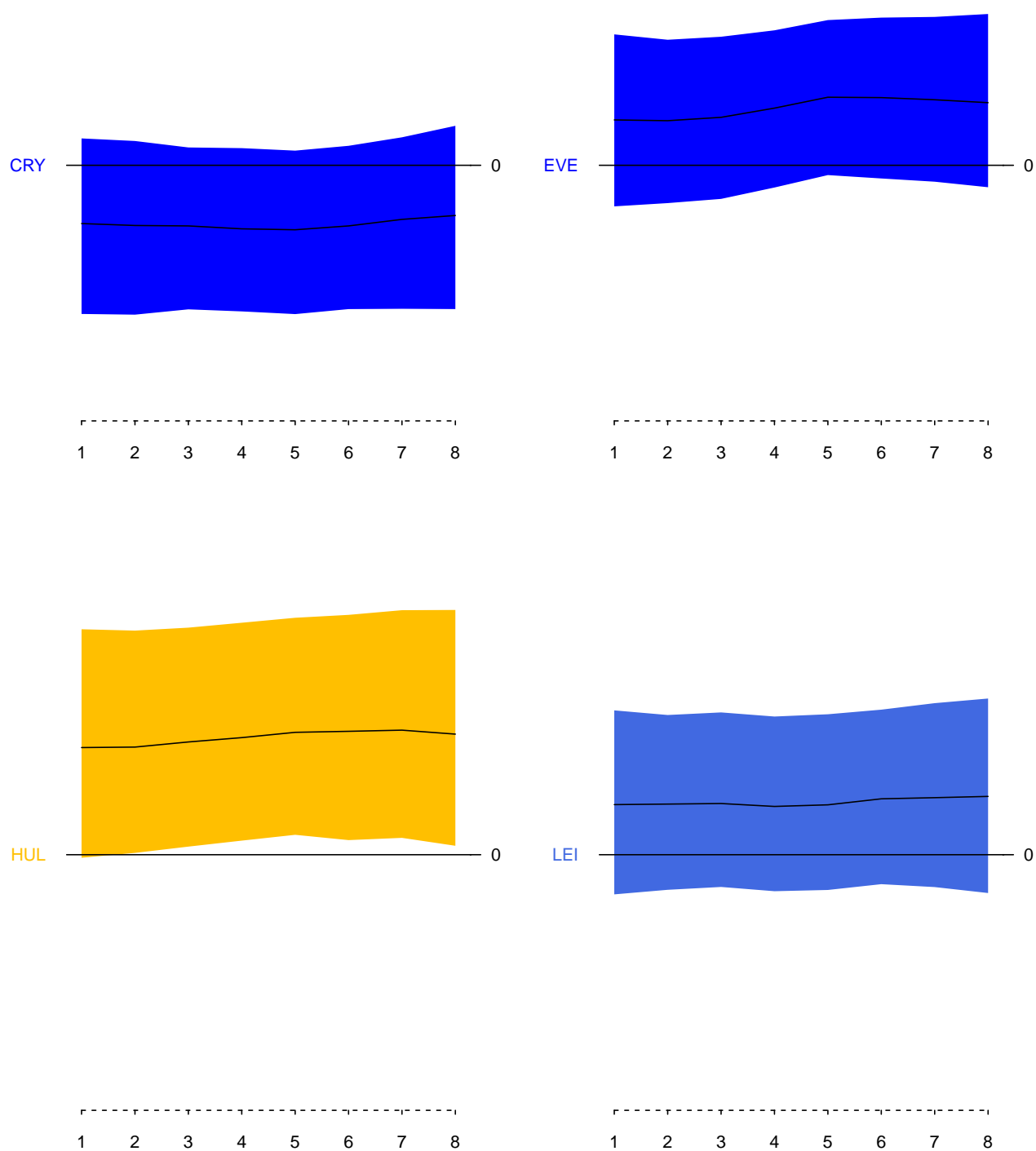


Figura A.79: Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 2.

EBs – Modelo Dinâmico

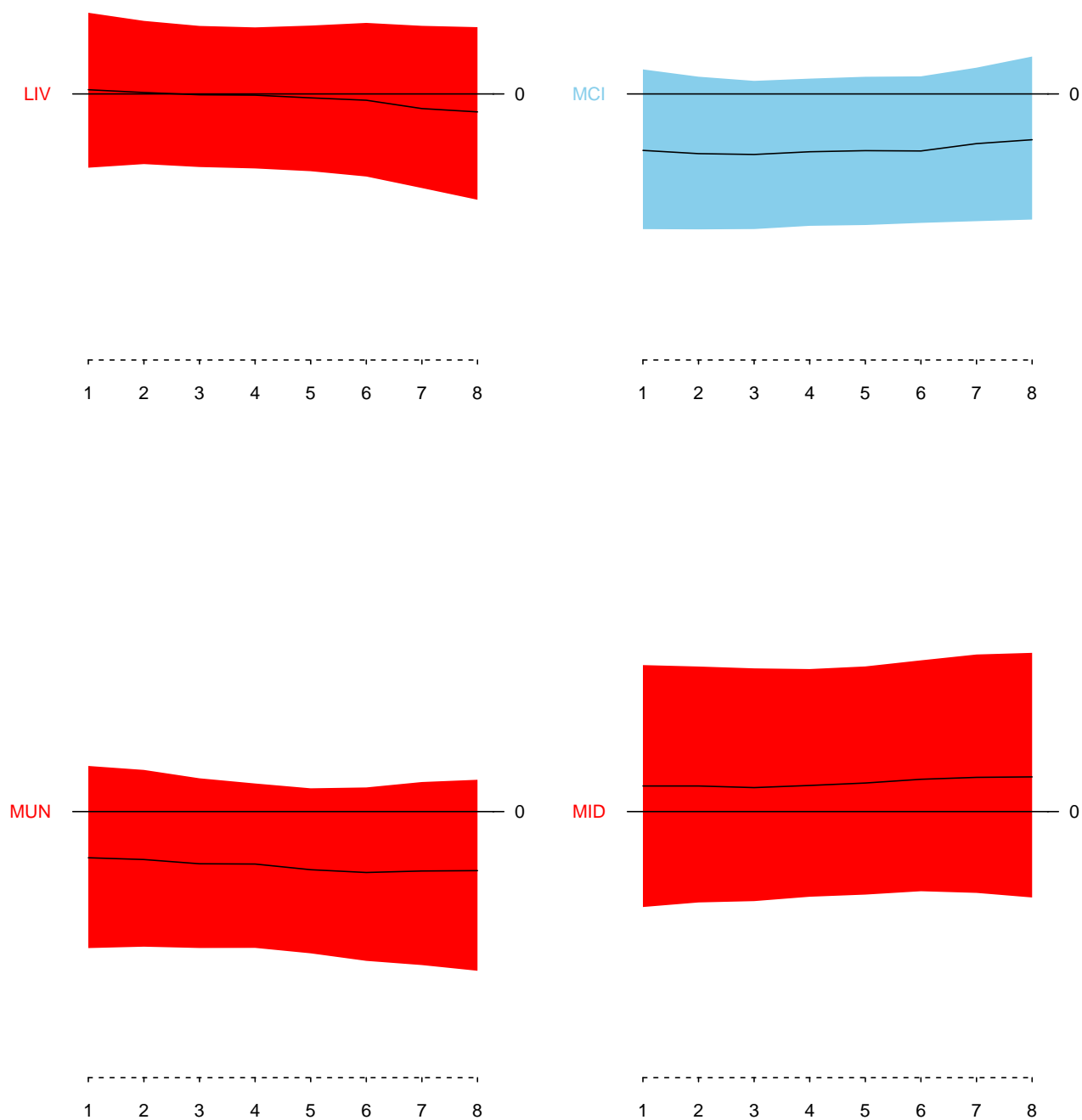


Figura A.80: Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 3.

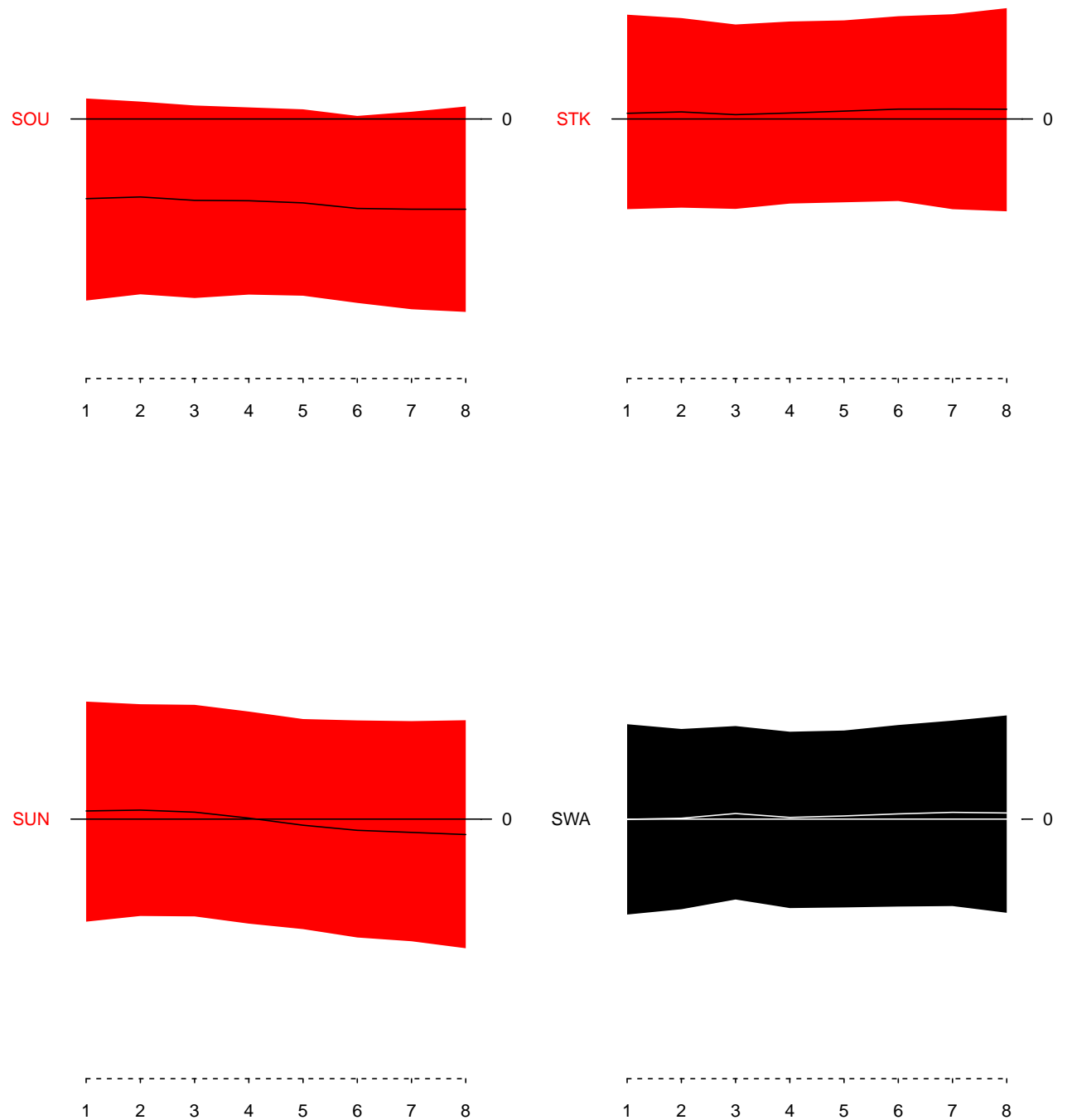


Figura A.81: Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 4.

EBs – Modelo Dinâmico

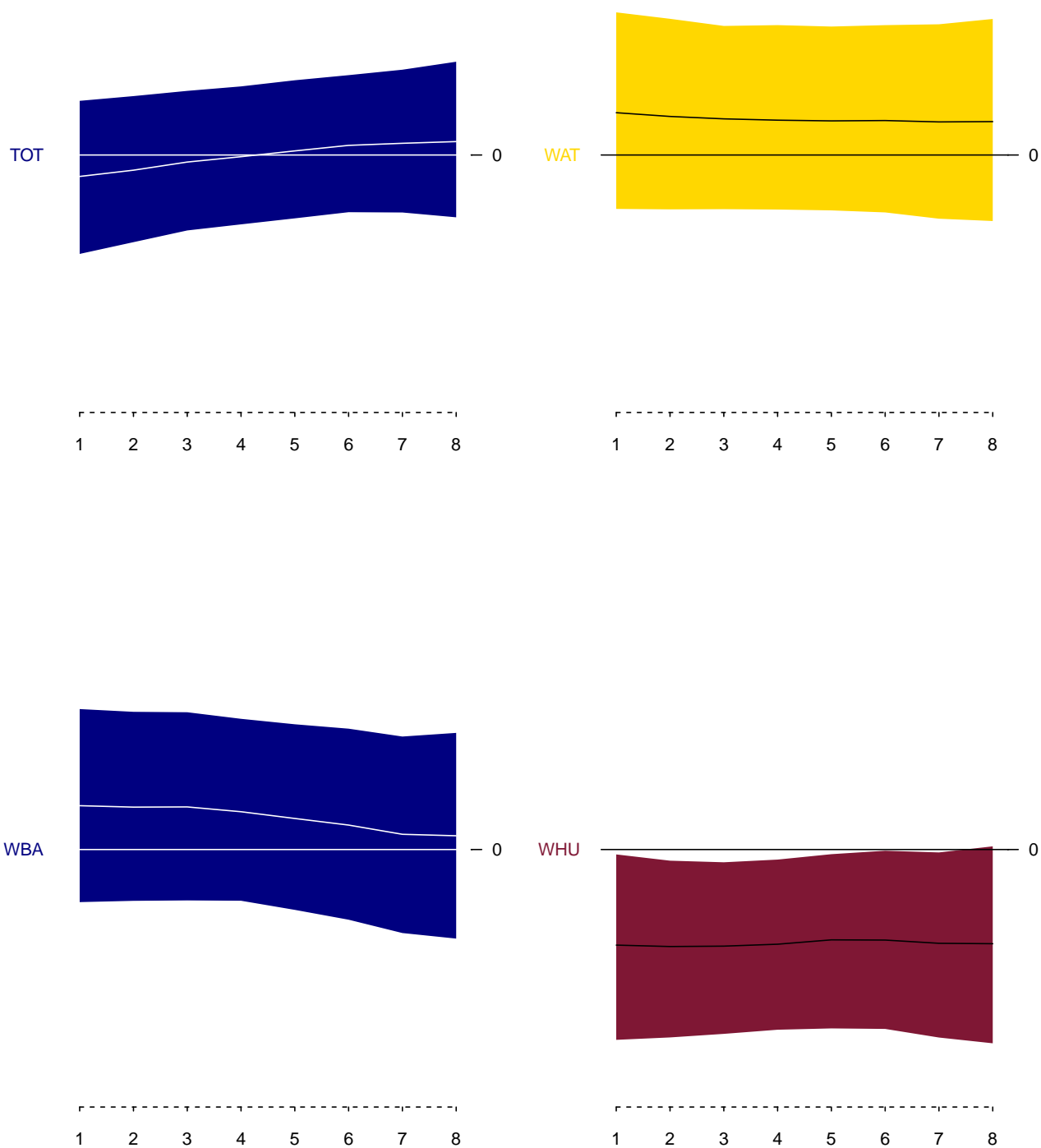


Figura A.82: Estimativas do fator campo para o Campeonato inglês 2016/17 - Parte 5.

EBs – Modelo Dinâmico

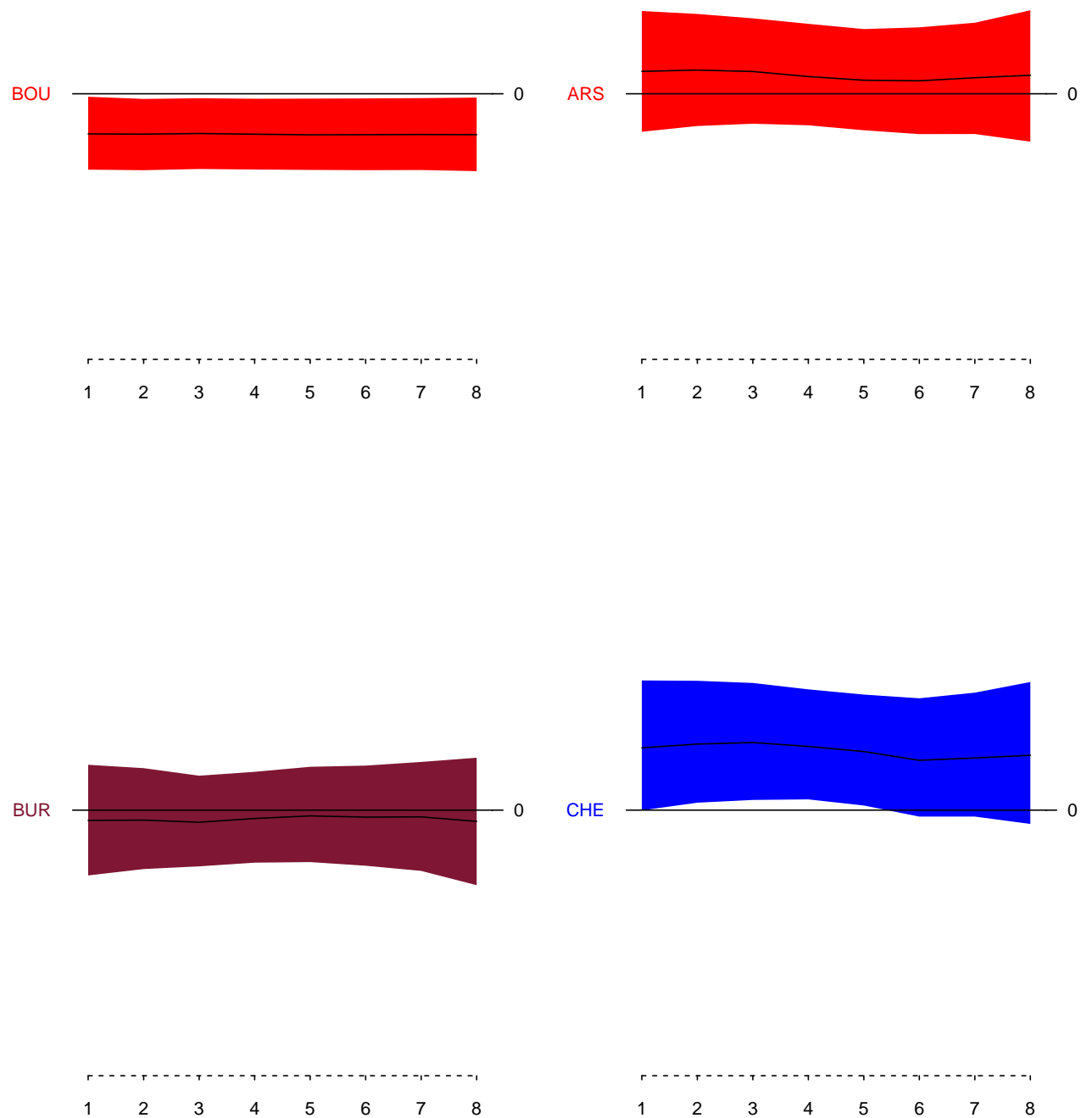


Figura A.83: Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 1.

EBs – Modelo Dinâmico

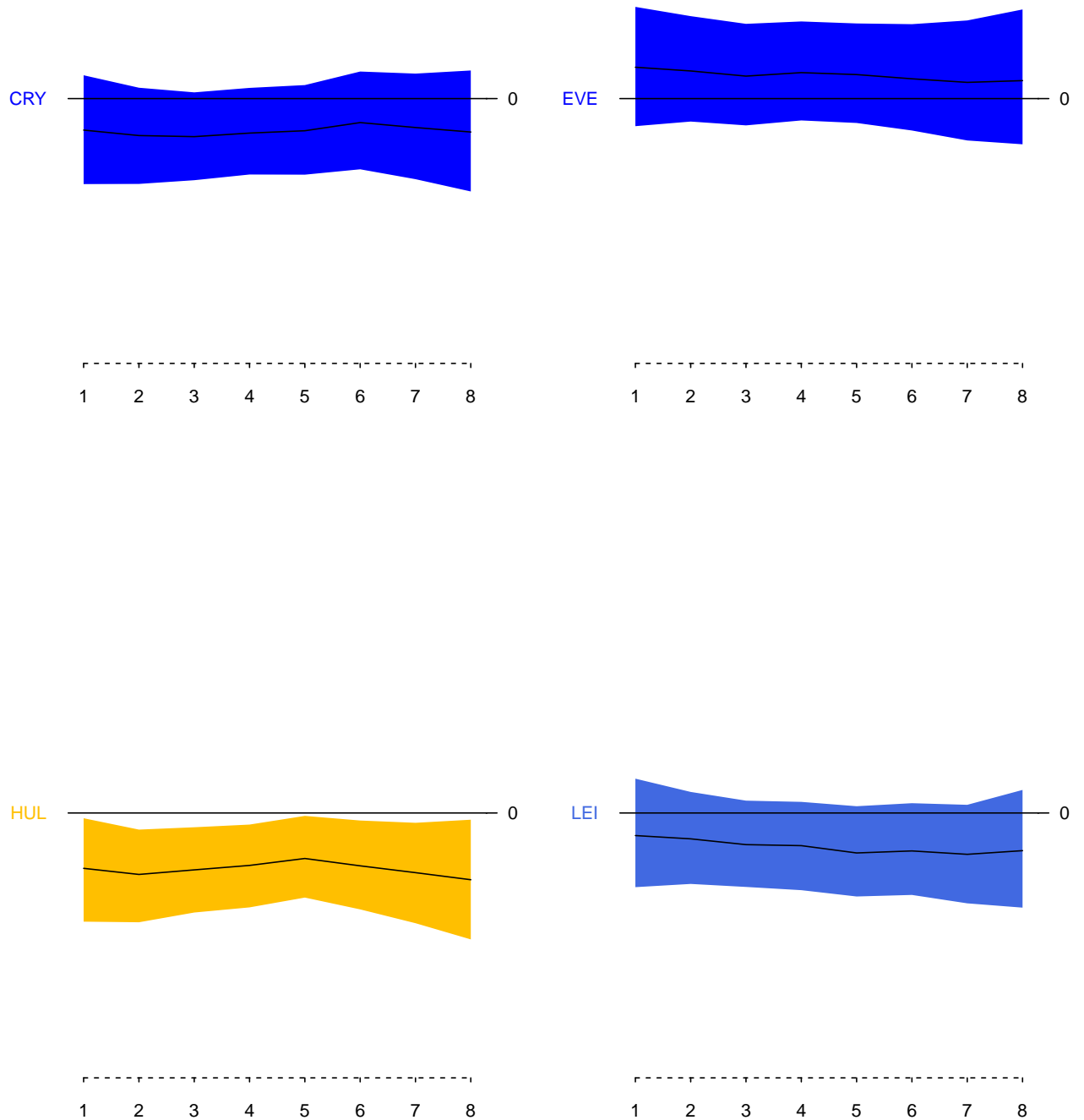


Figura A.84: Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 2.

EBs – Modelo Dinâmico

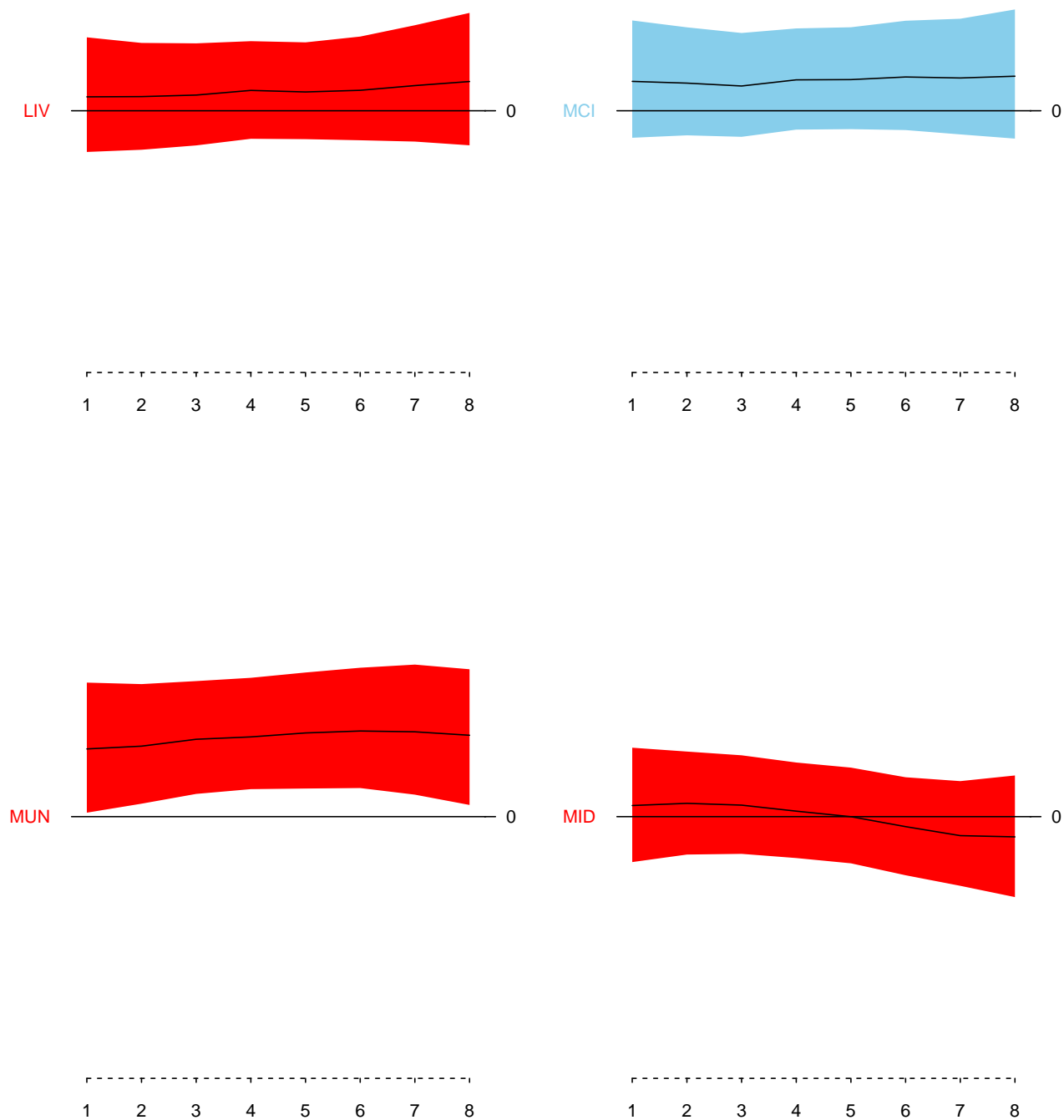


Figura A.85: Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 3.

EBs – Modelo Dinâmico

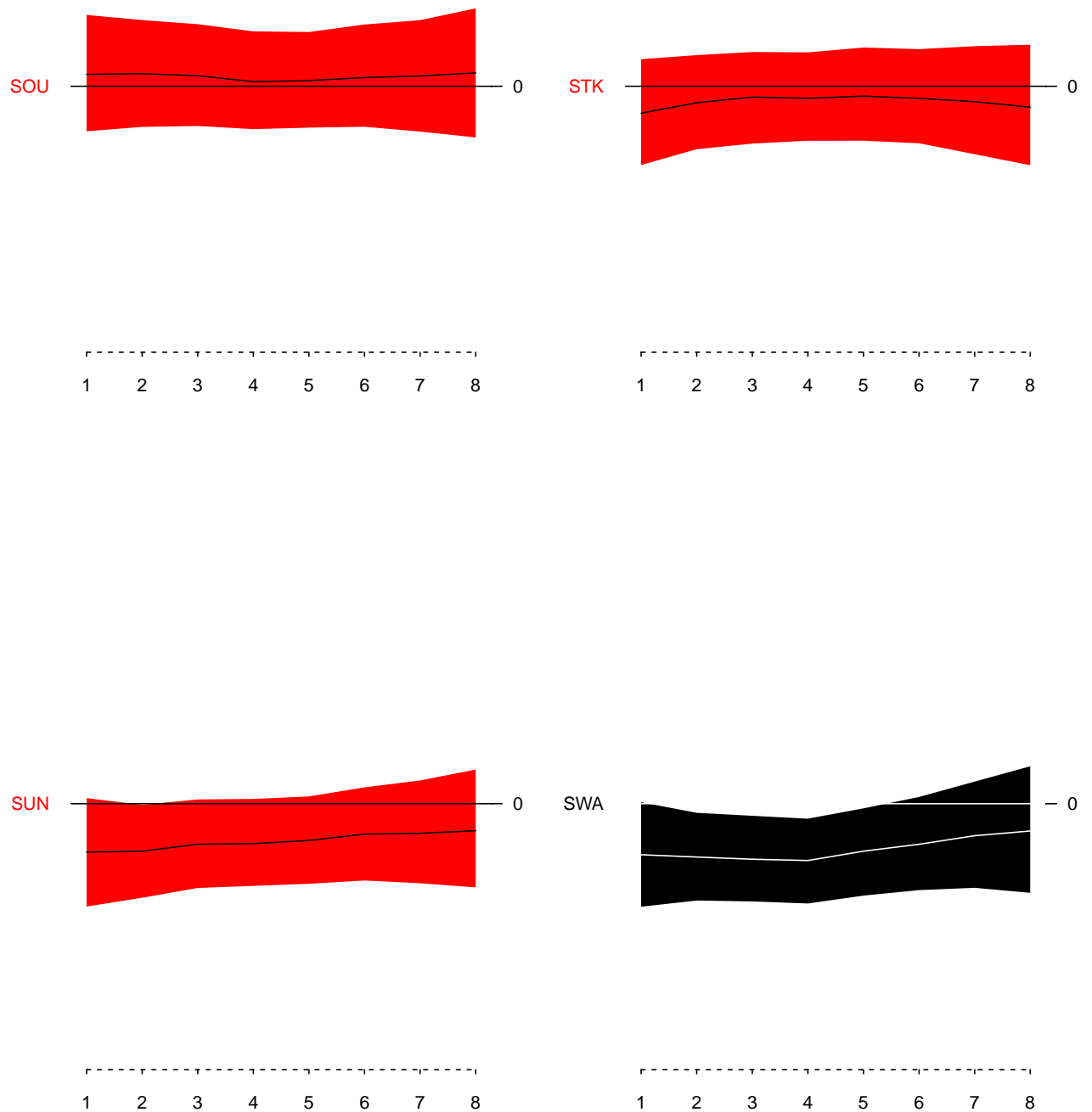


Figura A.86: Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 4.

EBs – Modelo Dinâmico

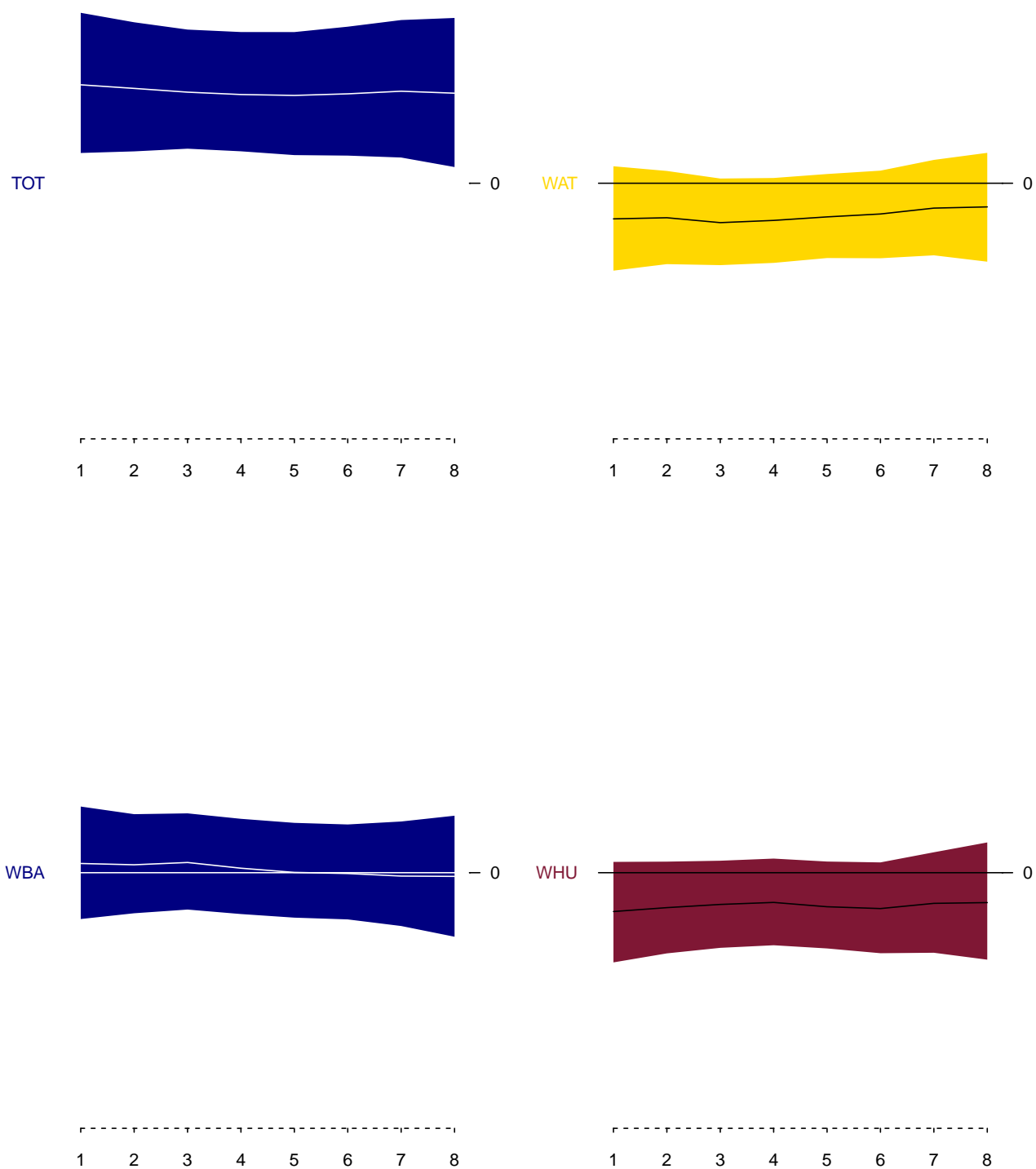


Figura A.87: Estimativas do fator defesa para o Campeonato inglês 2016/17 - Parte 5.

O comentário que pode ser feito sobre essa competição é que o Tottenham melhorou, ao longo do torneio. Sua defesa se manteve entre as mais fortes do campeonato ao longo do tempo, enquanto seu forte ataque e seu decepcionante mando de campo apresentaram melhorias, com o passar das rodadas. Se o aumento de qualidade tivesse começado mais cedo, ou sido mais dramático, a equipe talvez tivesse conseguido ser a vencedora, ao invés do Chelsea.

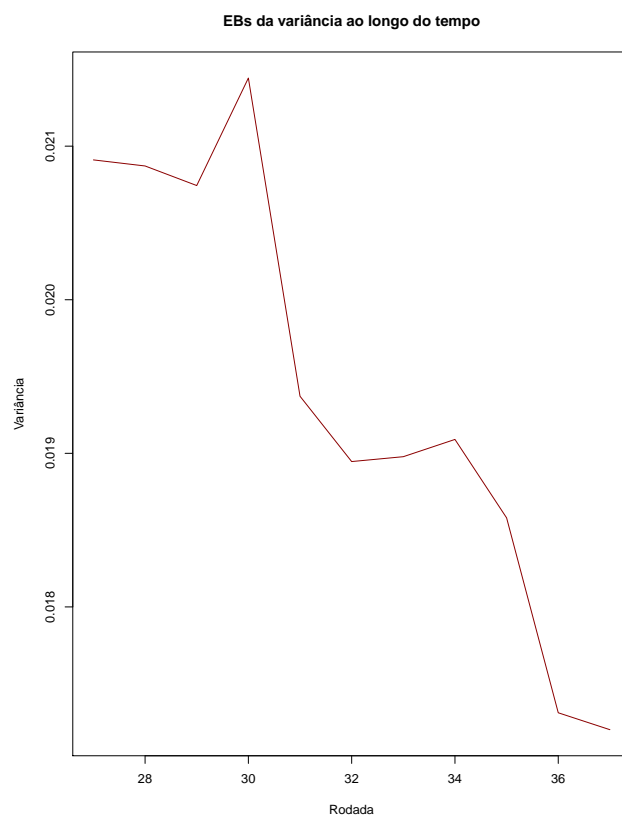


Figura A.88: Estimativas da variância das perturbações para o Campeonato inglês 2016/17.

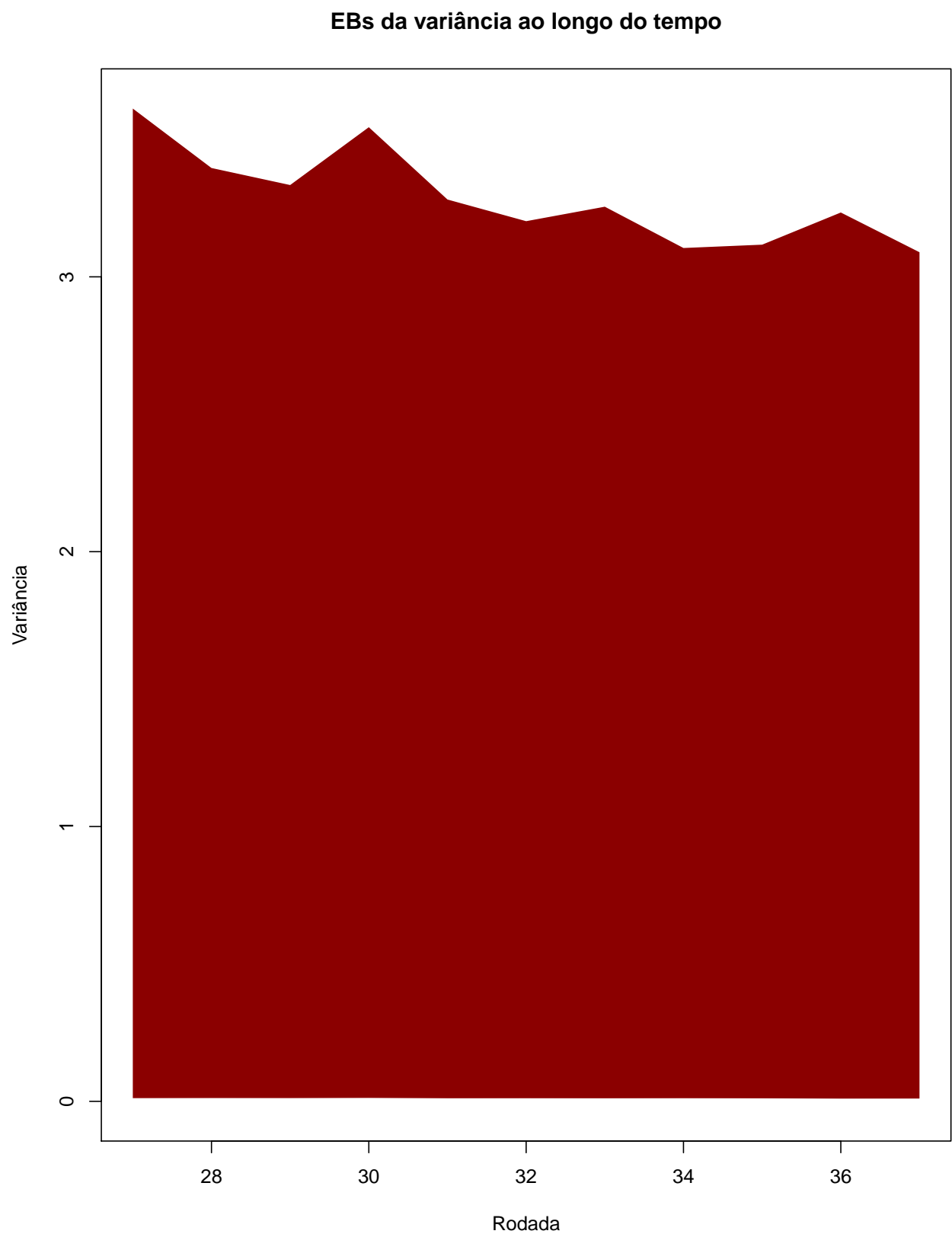


Figura A.89: Estimativas da variância das perturbações para o Campeonato inglês 2016/17.

Histograma da amostra da posteriori da variância

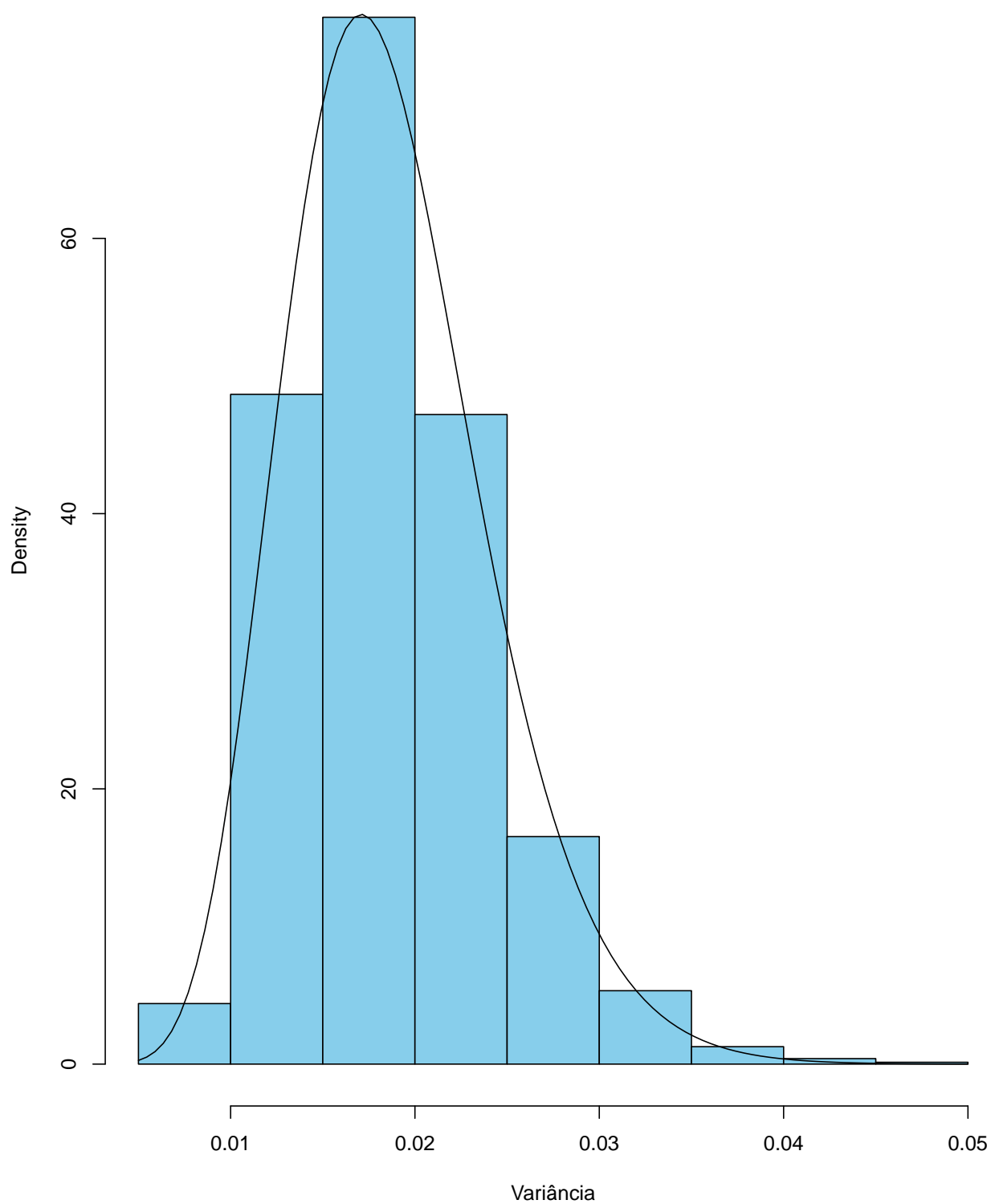


Figura A.90: Estimativas da variância das perturbações para o Campeonato inglês 2016/17.

As conclusões tiradas sobre as Figuras A.88 a A.90 são as mesmas tiradas sobre as Figuras A.22 a A.24, na seção A.3.2.

A.5.3 Verossimilhanças preditivas para os modelos

As Figuras A.91 a A.94 contêm os valores das verossimilhanças preditivas padronizadas do modelo estático e dos modelos usados para comparação neste trabalho. Especificamente, as Figuras A.92 e A.94 contêm as verossimilhanças preditivas padronizadas do modelo estático. A Figura A.92 apresenta as verossimilhanças preditivas para cada jogo do Campeonato brasileiro de 2017, enquanto a Figura A.94 apresenta as verossimilhanças do modelo por rodada.

As Figuras referentes ao modelo dinâmico são as Figuras A.95 a A.98. A única diferença dessas Figuras para as do modelo estático é que o modelo dinâmico passa a ser representado nelas. Elas foram geradas com o propósito de permitir comparações entre os modelos estático e dinâmico. Por isso, só foram coletadas previsões para o modelo dinâmico a partir das rodadas em que se considerou que havia ocorrido a estabilização das previsões do modelo estático.

Para o Campeonato inglês, o ponto de estabilização foi mais difícil de encontrar, em grande parte porque o desempenho do modelo foi excelente para quase todo o torneio. A partir da Figura A.92, notou-se uma diminuição da amplitude do modelo perto do jogo de número 250, e perto do jogo de número 300. Na Figura A.94, de novo há dois pontos que parecem simbolizar essa mudança qualitativa, nas rodadas 24 e 28. A rodada 16 também parece ter sido significativa. Escolheu-se agir de forma conservadora, e considerar a rodada 28 como o ponto de estabilização.

Para o Campeonato inglês, pode-se ver que os modelos dinâmico e estático tiveram qualidade semelhante. Quando se compara as Figuras A.92 e A.96 e as Figuras A.94 e A.98 nota-se que as previsões de ambos os modelos para esse campeonato foram muito semelhantes, em seu desempenho. A Figura A.61 confirma essas conclusões, e fica claro que há uma significativa mudança na qualidade das previsões do modelo, a partir da rodada 24. A partir dessa rodada, as verossimilhanças preditivas começam a apresentar valores mais altos.

Uma análise qualitativa das figuras apresentadas combinada com uma análise das Tabelas A.5 e A.6, aponta fortemente para uma grande mudança qualitativa nas verossimilhanças preditivas do modelo estático, por volta da rodada 28. Após esse ponto de

corte, o modelo estático aproxima-se muito em qualidade do modelo do *FiveThirtyEight*.

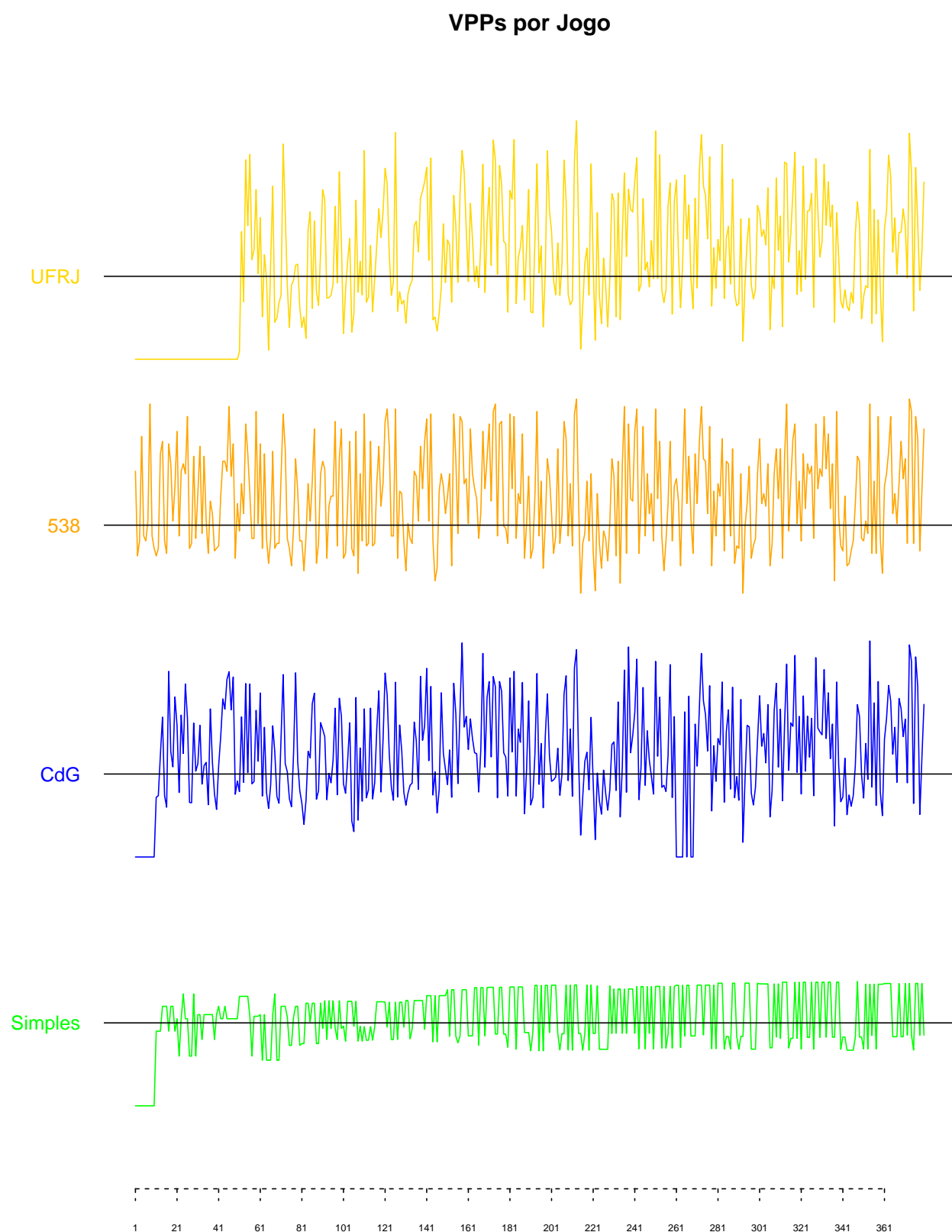


Figura A.91: Verossimilhanças preditivas padronizadas por jogo, para o Campeonato inglês 2016/17.

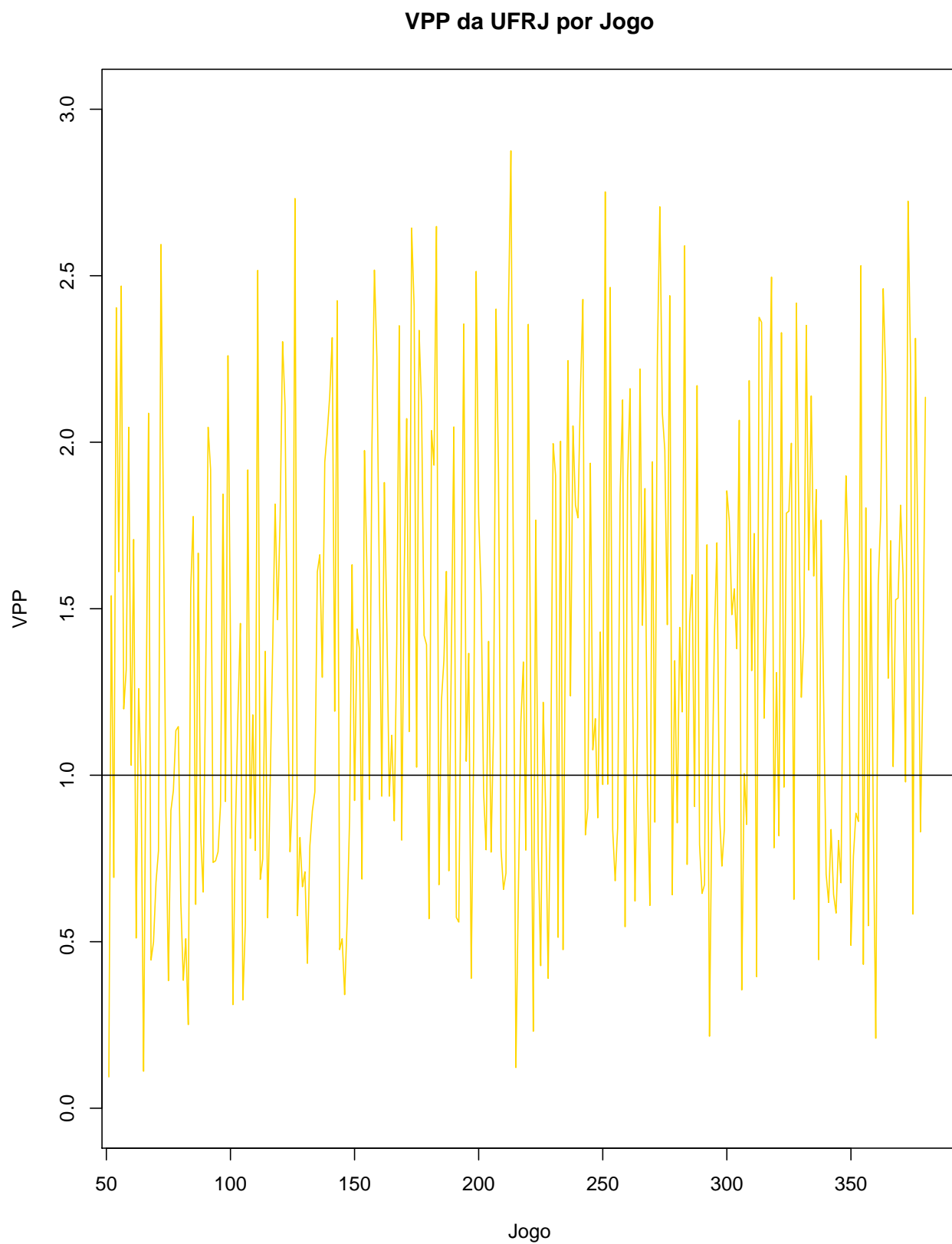


Figura A.92: Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato inglês 2016/17.

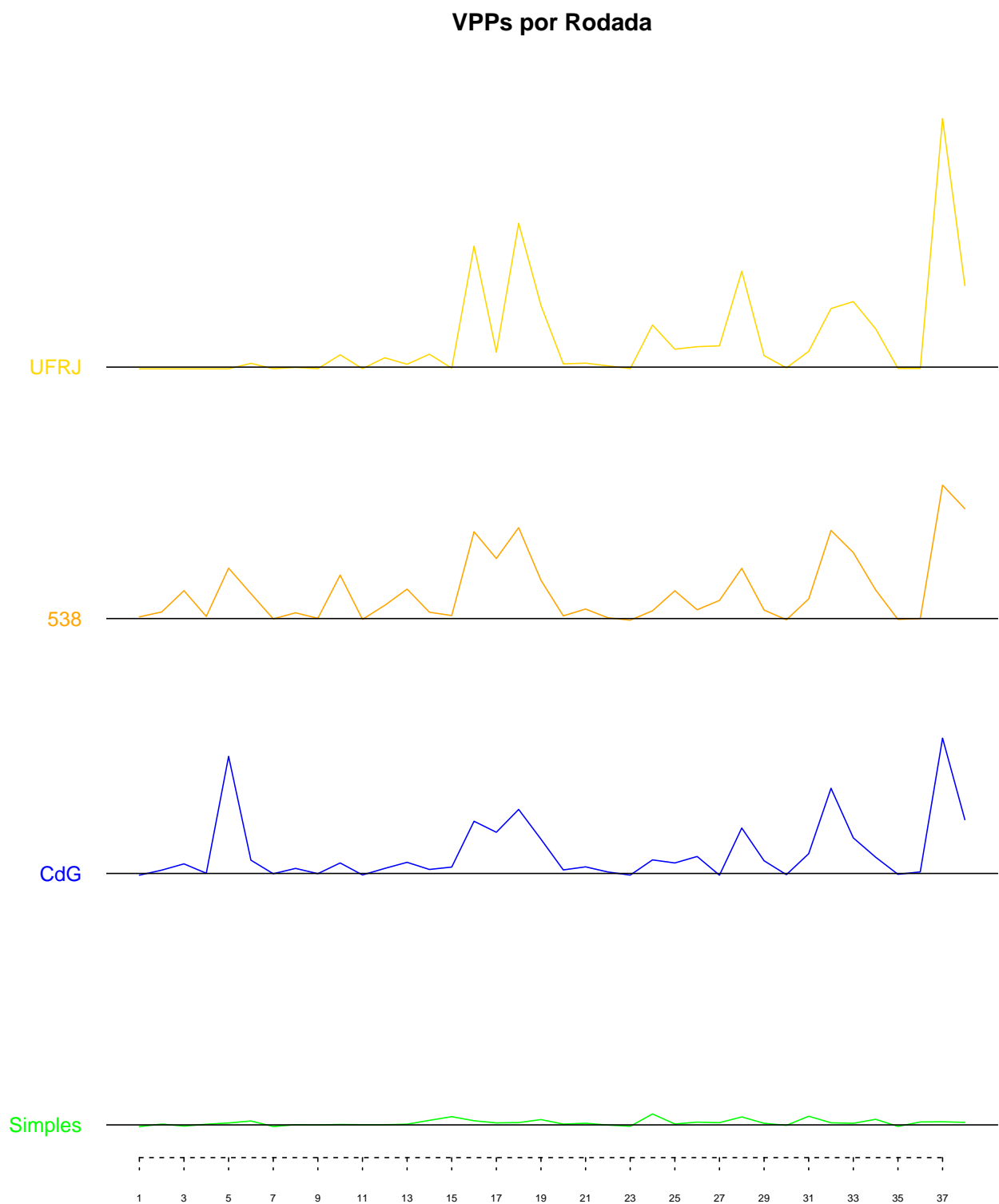


Figura A.93: Verossimilhanças preditivas padronizadas por rodada, para o Campeonato inglês 2016/17.

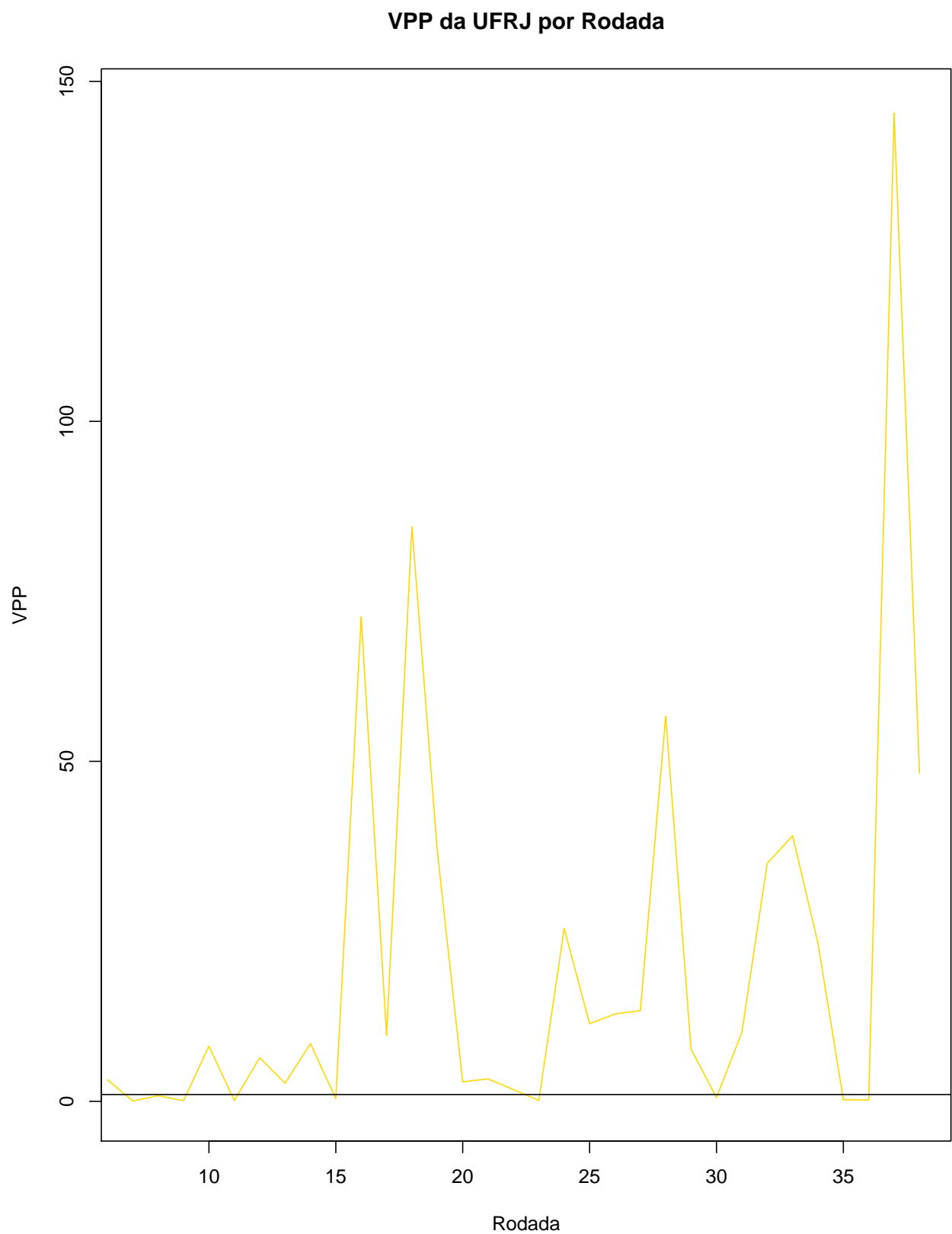


Figura A.94: Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato inglês 2016/17.

VPPs por Jogo

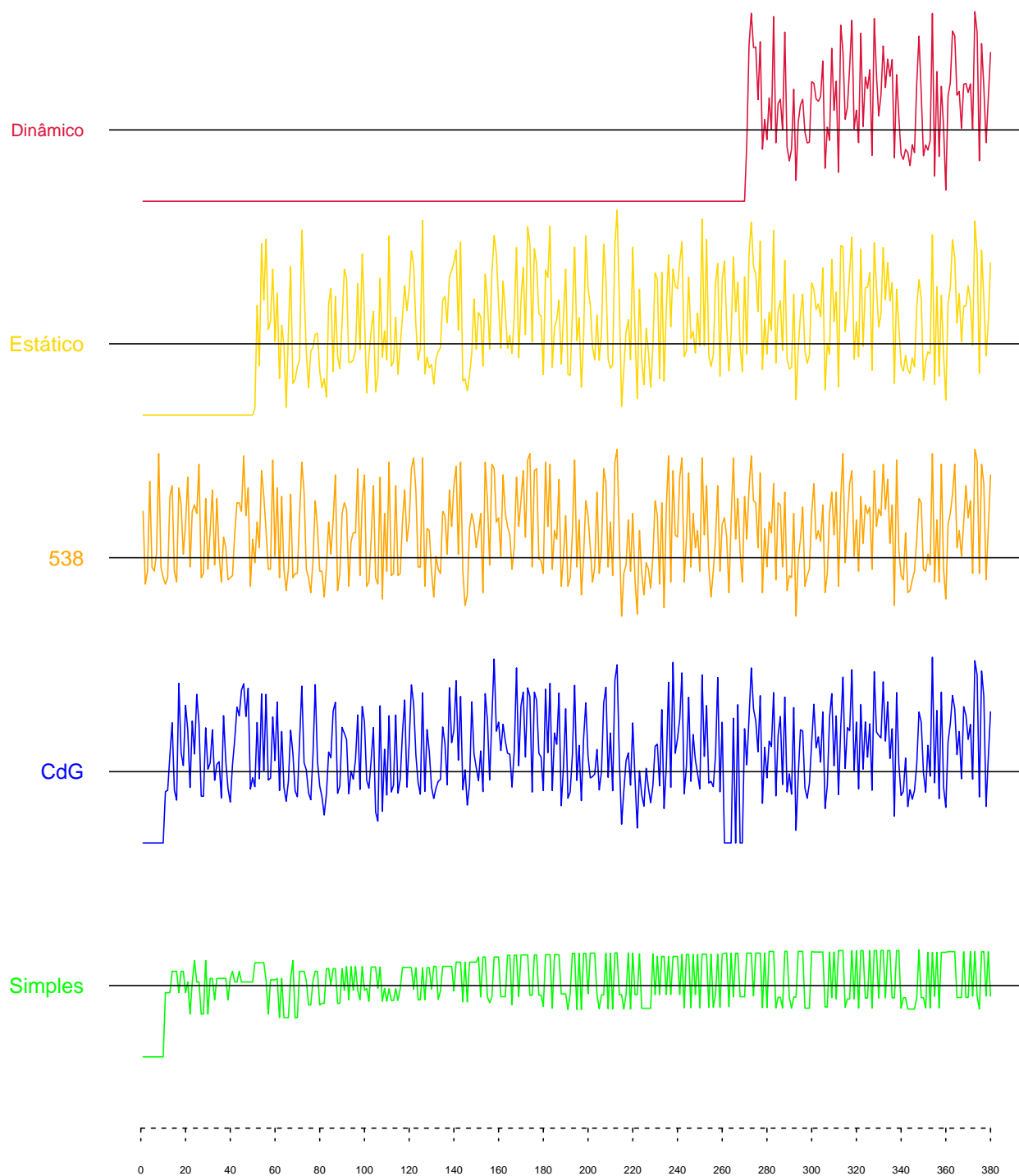


Figura A.95: Verossimilhanças preditivas padronizadas por jogo, para o Campeonato inglês 2016/17.

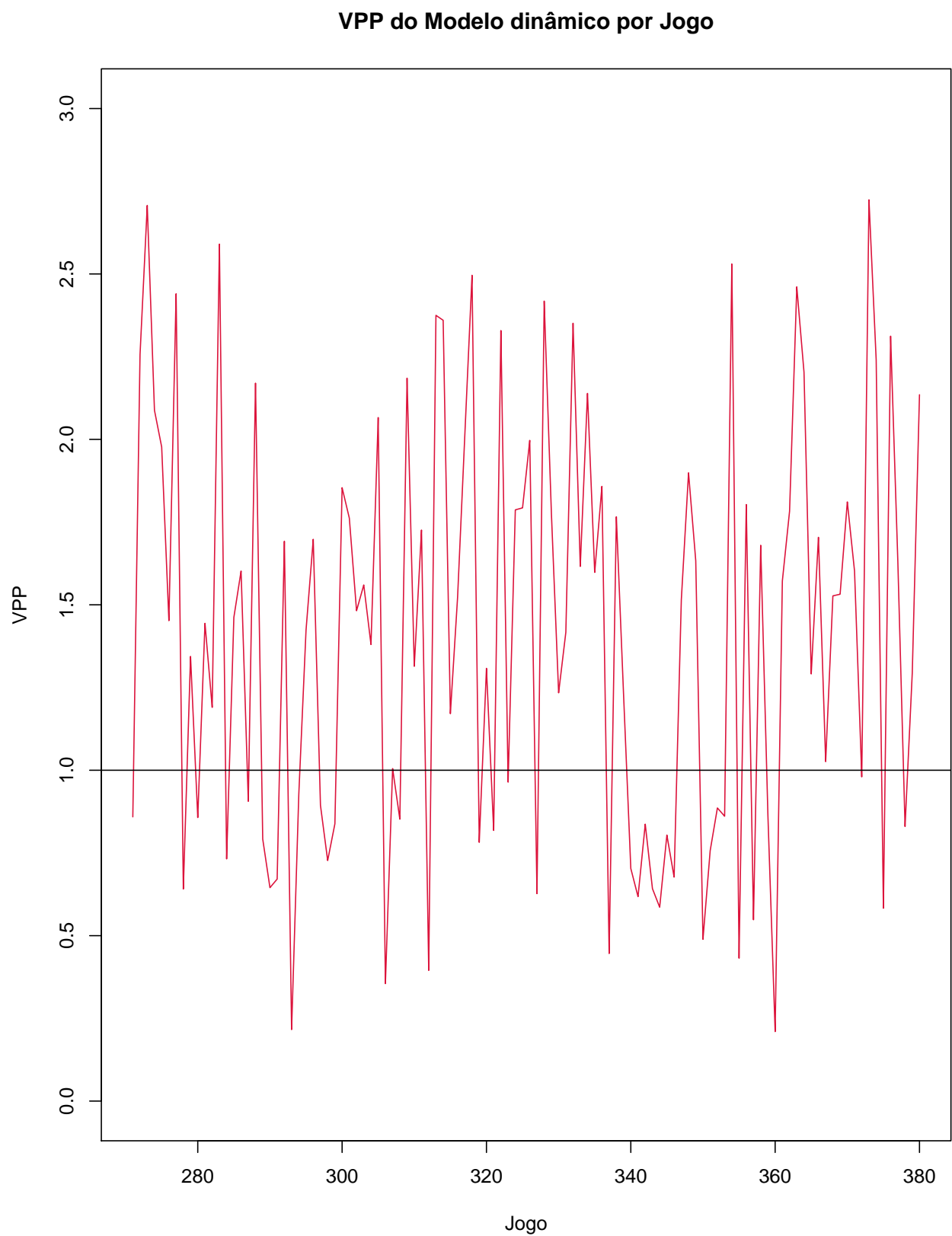


Figura A.96: Verossimilhanças preditivas padronizadas para o modelo estático por jogo, para o Campeonato inglês 2016/17.

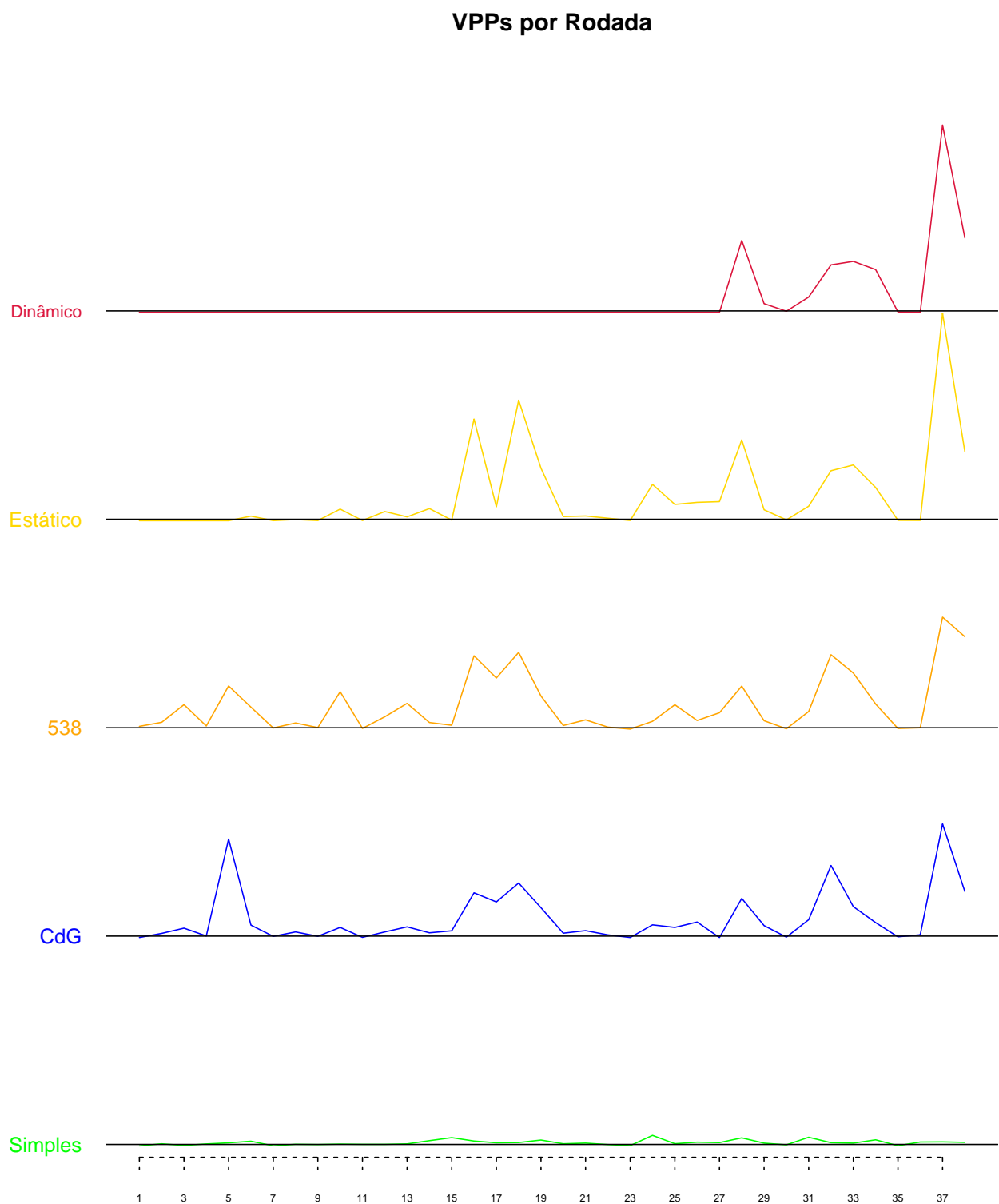


Figura A.97: Verossimilhanças preditivas padronizadas por rodada, para o Campeonato inglês 2016/17.

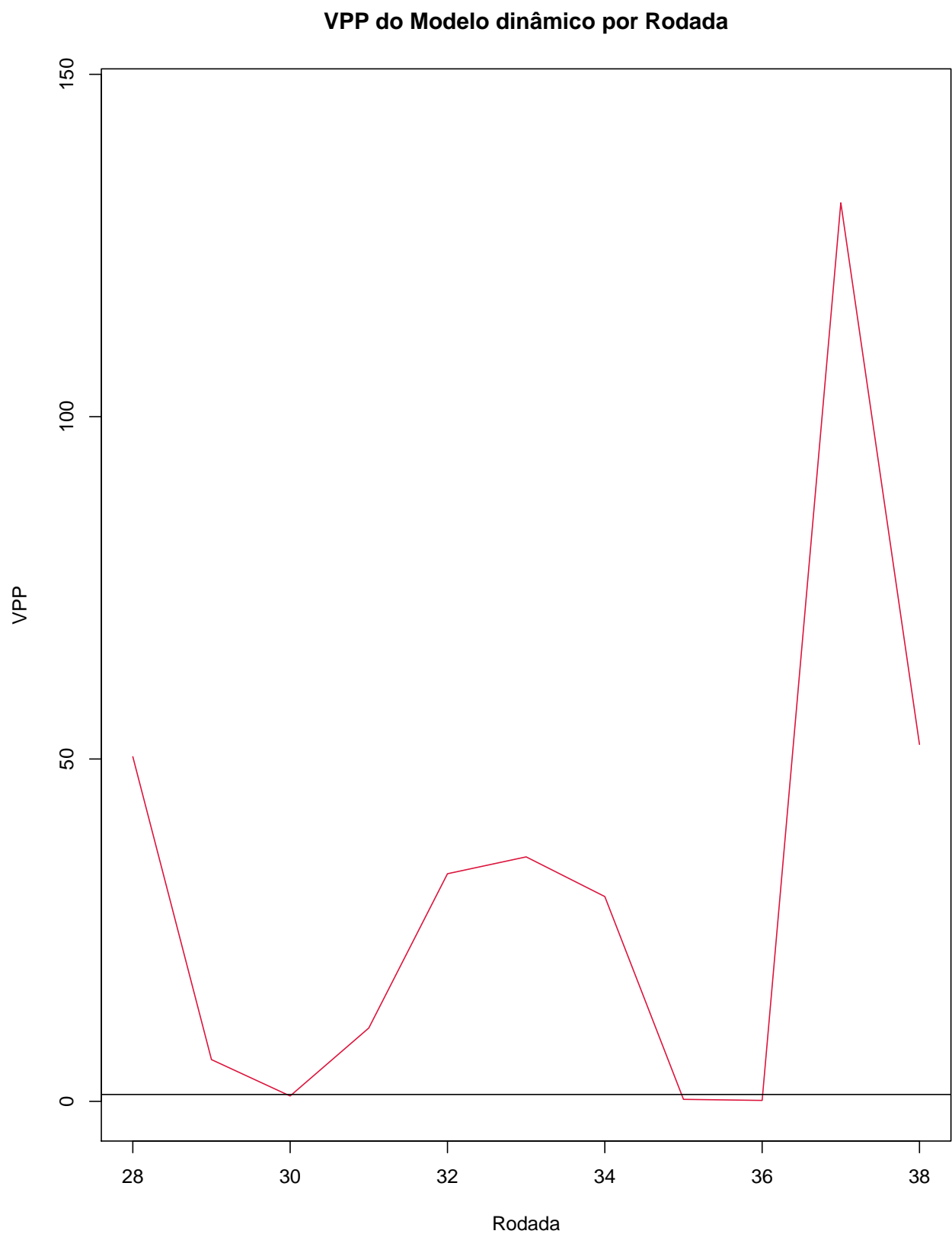


Figura A.98: Verossimilhanças preditivas padronizadas para o modelo estático por rodada, para o Campeonato inglês 2016/17.

Tabela A.5: Verossimilhanças preditivas para o Campeonato inglês 2016/2017.

Rodadas	Dinâmico	Estático	538	CdG	Nulo	Simples
6 a 10	0	0,09	1661,67	0	1	0,21
11 a 20	0	$4,16 \cdot 10^7$	$6,41 \cdot 10^9$	0	1	3515,41
21 a 30	0	$8,86 \cdot 10^6$	$1,88 \cdot 10^5$	0	1	193,1
31 a 38	10^8	$1,04 \cdot 10^8$	$1,14 \cdot 10^9$	$3,49 \cdot 10^8$	1	248,24
Total	10^8	$3,46 \cdot 10^{21}$	$2,29 \cdot 10^{27}$	$3,49 \cdot 10^8$	1	$3,52 \cdot 10^7$

Tabela A.6: Verossimilhanças preditivas para o Campeonato inglês 2016/2017.

Rodadas	Dinâmico	Estático	538	CdG	Nulo	Simples
6 a 27	0	$3,46 \cdot 10^{21}$	$2,29 \cdot 10^{27}$	0	1	$3,52 \cdot 10^7$
28 a 38	$2,4 \cdot 10^{10}$	$2,52 \cdot 10^{10}$	$5,62 \cdot 10^{10}$	$2,47 \cdot 10^{10}$	1	1977,07

As investigações começadas na seção [A.3.3](#) e continuadas na seção [A.4.3](#) terminam nesta seção. As são análogas às [A.33](#) a [A.36](#) e às Figuras [A.66](#) a [A.69](#), tendo mudado apenas o campeonato estudado e os times avaliados.

As conclusões, contudo, continuam sendo as mesmas: depois do ponto de estabilização das previsões, a dispersão das amostras para de diminuir significativamente e as prioris escolhidas parecem ter sido vagas demais.

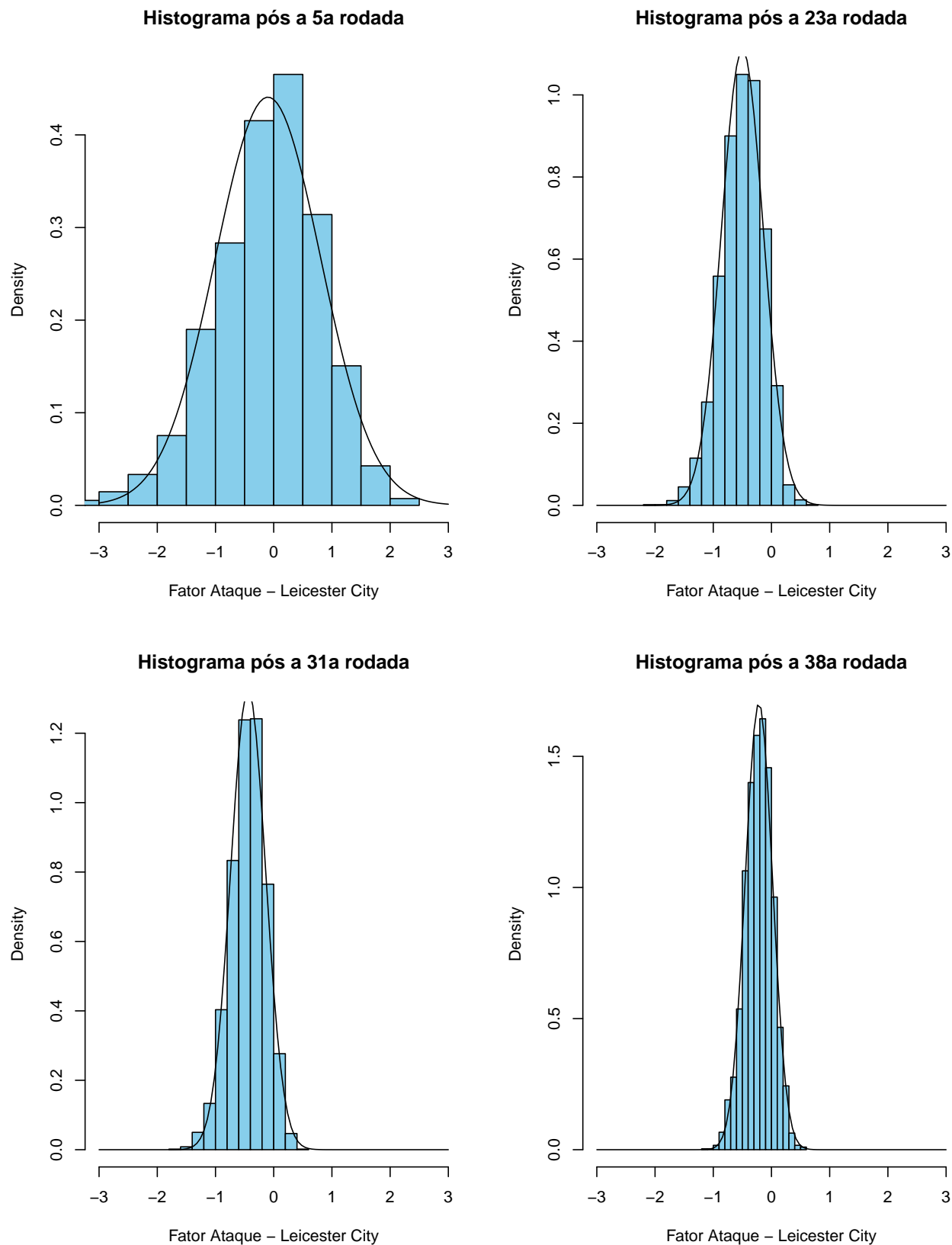


Figura A.99: Amostras aproximadas da posteriori do fator de ataque do Atlético-MG.

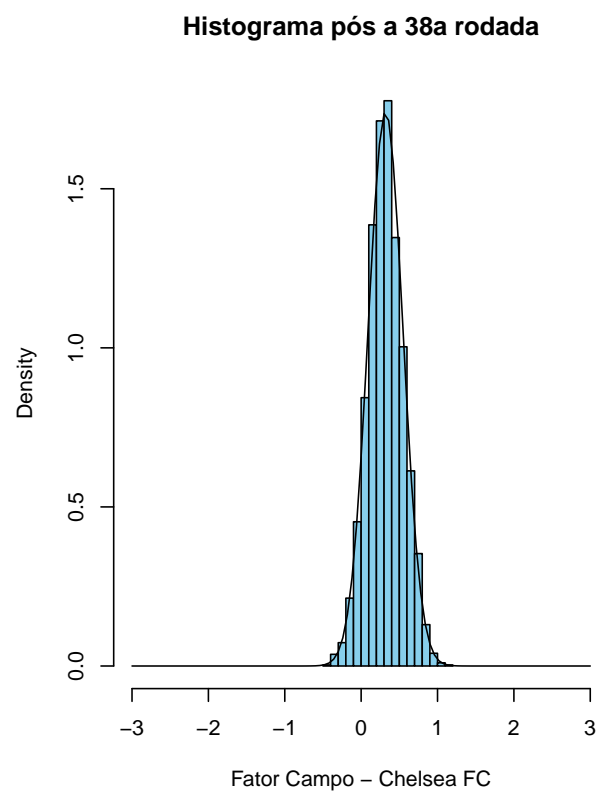
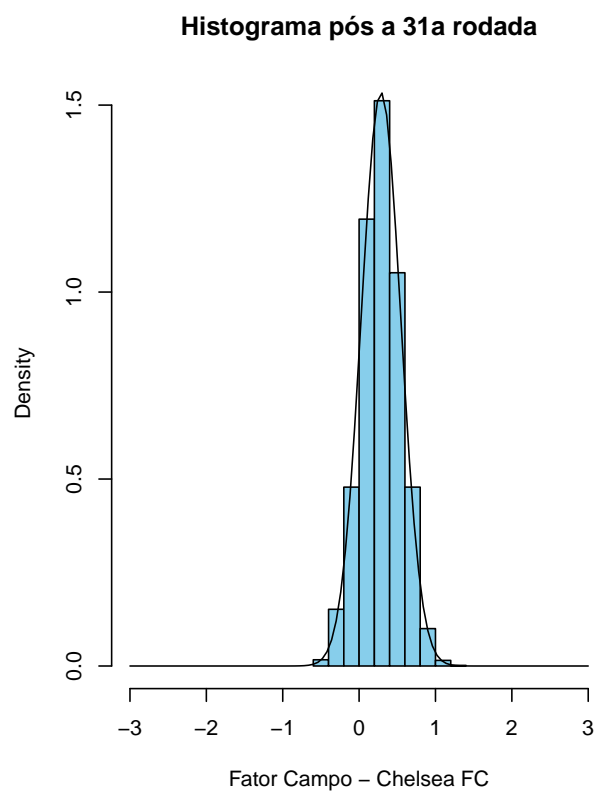
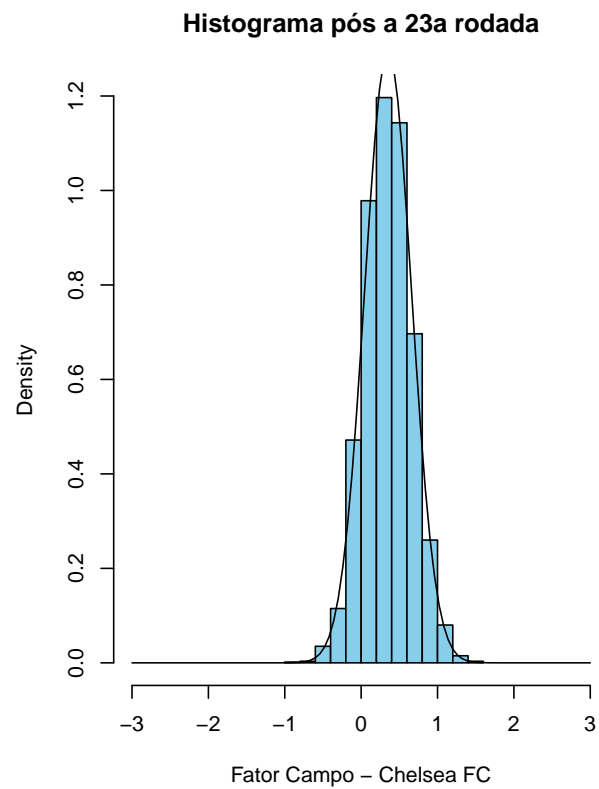
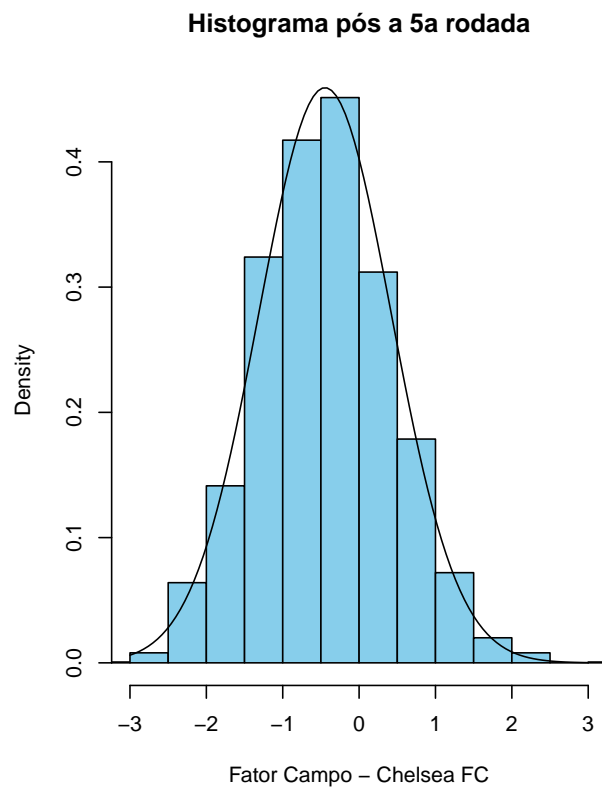


Figura A.100: Amostras aproximadas da posteriori do fator campo do Grêmio.

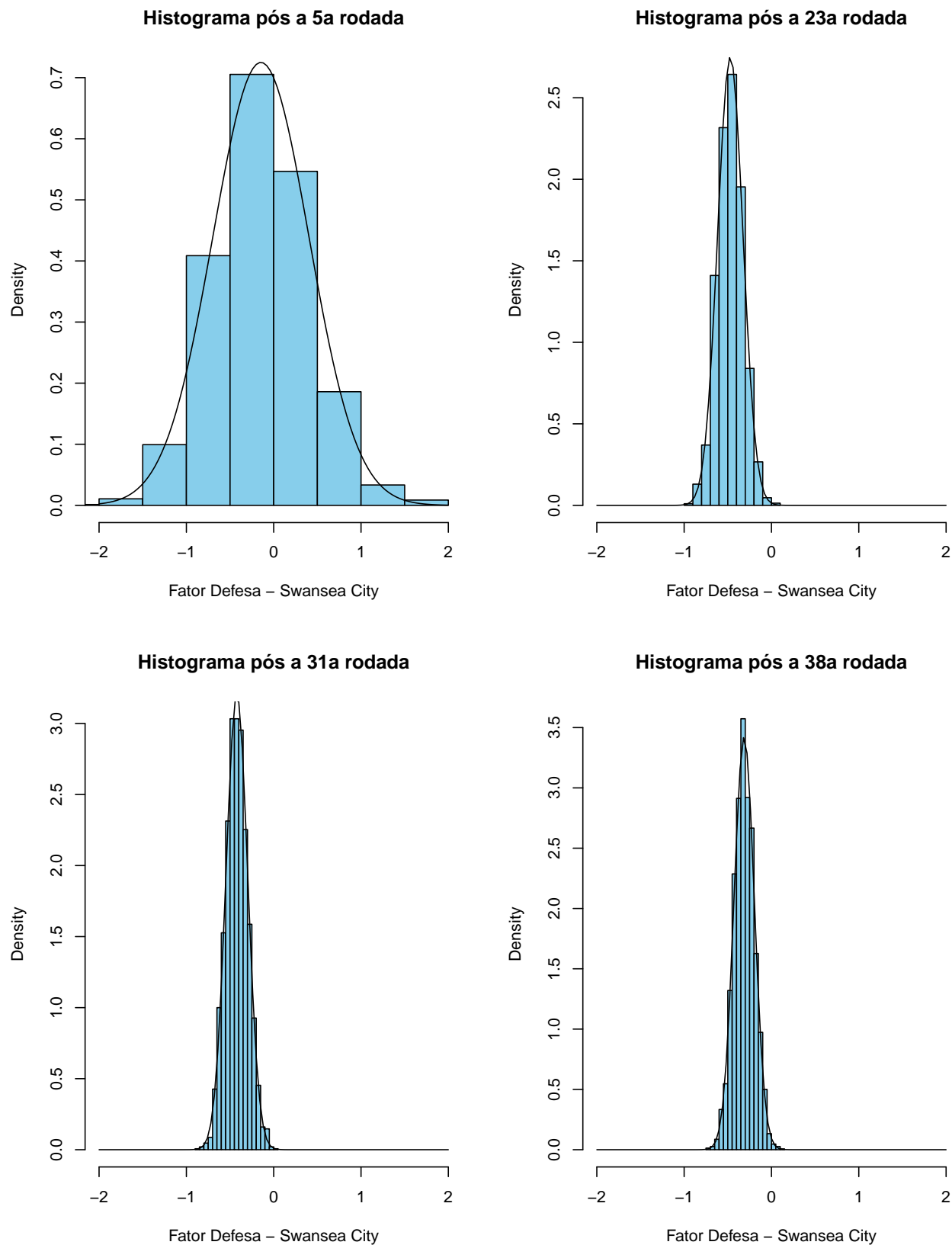


Figura A.101: Amostras aproximadas da posteriori do fator de defesa do Atlético-PR.

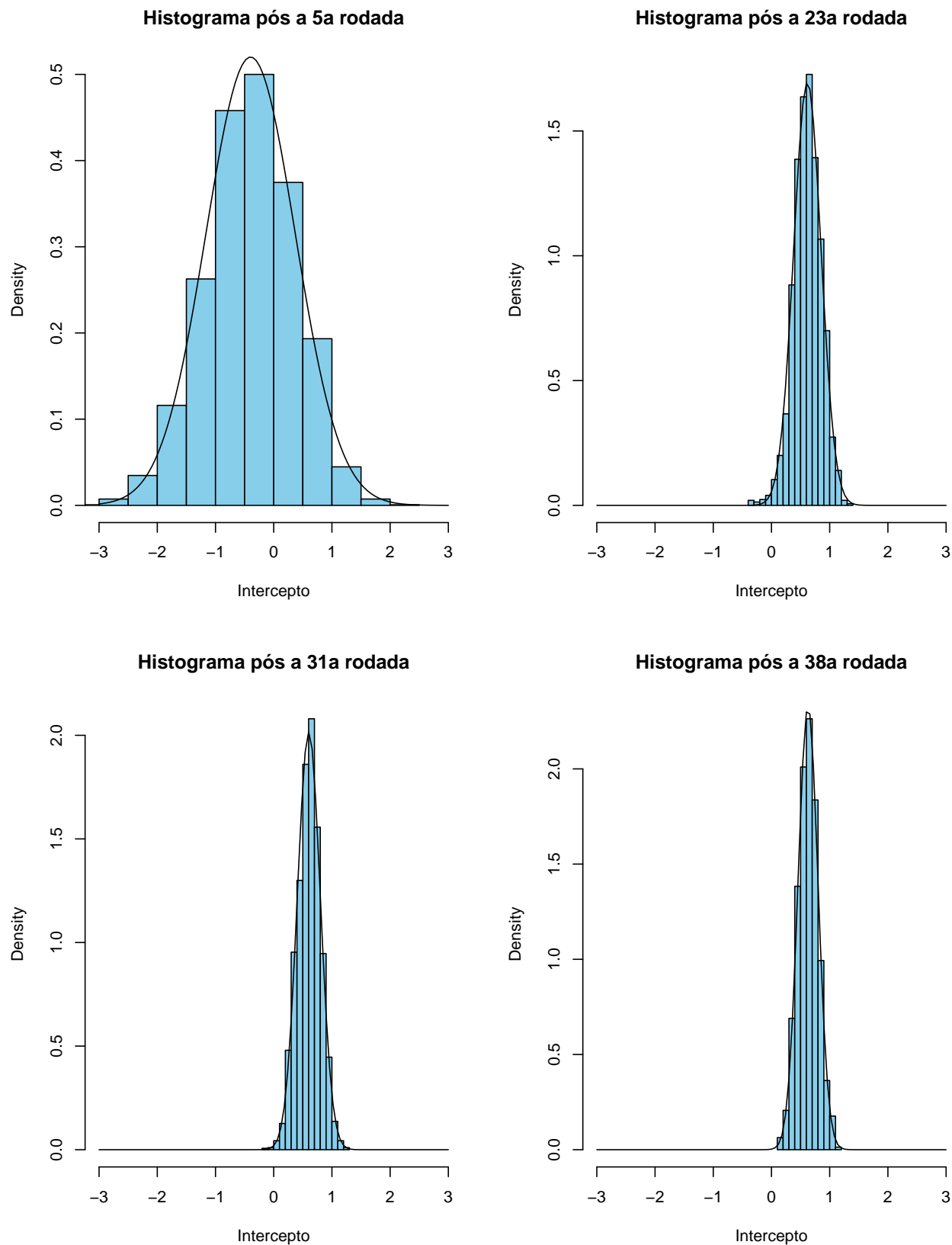


Figura A.102: Amostras aproximadas da posteriori do intercepto do Campeonato brasileiro de 2017.

Apêndice B

Neste apêndice estarão os códigos computacionais usados para gerar as amostras, estimativas, figuras, previsões e tabelas mostradas e usadas no presente trabalho. Foram usados dois programas: o OpenBUGS para o método MCMC e o R para todo o resto.

No caso do OpenBUGS, os códigos para os modelos estão a seguir. Primeiro, o modelo estático:

$$A_1 = 0$$

$$D_1 = 0$$

$$C_1 = 0$$

$$\left. \begin{array}{l} A_i \sim \text{dnorm}(0.0, 0.01) \\ D_i \sim \text{dnorm}(0.0, 0.01) \\ C_i \sim \text{dnorm}(0.0, 0.01) \end{array} \right\} 2 \leq i \leq N$$

$$I \sim \text{dnorm}(0.0, 0.01)$$

$$\left. \begin{array}{l} Y_{i,2,j-1} \sim \text{dpois}(\lambda_{i,2,j-1}) \\ \lambda_{i,2,j-1} = e^{AX_{i,2,j-1} - DX_{i,2,j} + CX_{i,2,j-1} + I} \\ Y_{i,2,j} \sim \text{dpois}(\lambda_{i,2,j}) \\ \lambda_{i,2,j} = e^{AX_{i,2,j} - DX_{i,2,j-1} + I} \end{array} \right\} 1 \leq j \leq N/2 \left. \right\} 1 \leq i \leq K$$

Depois, o dinâmico:

$$\begin{aligned}
I_1 &\sim \text{dnorm}(0.0, 0.01) \\
A_{1,1} &= 0 \\
D_{1,1} &= 0 \\
C_{1,1} &= 0 \\
P &\sim \text{dgamma}(0.1, 0.1) \\
\left. \begin{aligned}
A_{i,1} &\sim \text{dnorm}(0.0, 0.01) \\
D_{i,1} &\sim \text{dnorm}(0.0, 0.01) \\
C_{i,1} &\sim \text{dnorm}(0.0, 0.01)
\end{aligned} \right\} 2 \leq i \leq N \\
\left. \begin{aligned}
A_{1,i} &= 0 \\
D_{1,i} &= 0 \\
C_{1,i} &= 0 \\
\left. \begin{aligned}
wA_{j,i-1} &\sim \text{dnorm}(0.0, P) \\
wD_{j,i-1} &\sim \text{dnorm}(0.0, P) \\
wC_{j,i-1} &\sim \text{dnorm}(0.0, P) \\
A_{j,i} &= A_{j,i-1} + wA_{j,i-1} \\
D_{j,i} &= D_{j,i-1} + wD_{j,i-1} \\
C_{j,i} &= C_{j,i-1} + wC_{j,i-1}
\end{aligned} \right\} 2 \leq j \leq N \\
wI_{i-1} &\sim \text{dnorm}(0.0, P) \\
I_i &= I_{i-1} + wI_{i-1}
\end{aligned} \right\} 2 \leq i \leq K_2 \\
k_i &= \text{trunc}(i/5 + 0.99) \\
\left. \begin{aligned}
Y_{i,2 \cdot j-1} &\sim \text{dpois}(\lambda_{i,2 \cdot j-1}) \\
\lambda_{i,2 \cdot j-1} &= e^{AX_{i,2 \cdot j-1}, k_i - DX_{i,2 \cdot j}, k_i + CX_{i,2 \cdot j-1}, k_i + I_{k_i}} \\
Y_{i,2 \cdot j} &\sim \text{dpois}(\lambda_{i,2 \cdot j}) \\
\lambda_{i,2 \cdot j} &= e^{AX_{i,2 \cdot j}, k_i - DX_{i,2 \cdot j-1}, k_i + I_{k_i}}
\end{aligned} \right\} 1 \leq j \leq N/2 \left. \right\} 1 \leq i \leq K
\end{aligned}$$

Com esses códigos, é possível gerar as amostras aproximadas das posteriores dos parâmetros, e pode-se, então, partir para o R. Nele, há três partes de código. A primeira, para gerar estimativas e precisões a partir dos dados de um campeonato, e das amostras para as posteriores de seus parâmetros, usando o modelo estático:

```
#-----DADOS-----
Probs=matrix(NA,320,5)
POS=list()
linhas=c()
times=file.choose()
n=length(times)
for(r in 6:37){
  file=file.choose()
  a=read.csv(file,header=T)#dados brutos
  fatores=c("Ataque",'Campo','Defesa')
  b=c()#vetor com os dados sem as datas dos jogos
  n3=length(a[[1]])#numero de linhas no arquivo
  for(c in 1:n3){
    d=grep(";",a[[1]][c],value=TRUE)#tira os "levels"
    e=strsplit(d,";")#quebra a linha do arquivo
    e[[1]][2]=gsub(pattern='',replacement='',
                    x=e[[1]][2])#tira o sinal
    #bizarro que pode aparecer no final do nome
    #do mandante
    e[[1]][3]=gsub(pattern='',replacement='',
                    x=e[[1]][3])#tira o sinal
    #bizarro que pode aparecer no final do nome
    #do visitante
    f=length(e[[1]])#quantidade de dados na linha
    b=append(b,e[[1]][2:f])
  }
  c=matrix(b,c,(f-1),byrow=TRUE)#matriz com os dados
  d=strsplit(c[,3],"")#quebra as strings dos placares
  c=c[,-3]#tira a coluna com os placares
```

```

c=cbind(c,unlist(lapply(d,'[[',1)),
unlist(lapply(d,'[[',5))) #acrescenta 0's,
#para que se possa colocar o numero de gols
#feitos pelos times
d=match(c[,1],times) #mandantes como numeros
e=match(c[,2],times) #visitantes como numeros
f=as.numeric(c[,4:5]) #gols num vetor
d=cbind(f,c(d,e),c(e,d)) #matriz com os gols e os
#ataques e defesas responsaveis
for(e in 1:n){
  f=(match(c[,1],times)==e) #variavel indicadora para
  #se o e-esimo time e o mandante
  g=c(f,(f*0)) #acrescenta-se zeros, para os jogos
  #em que o time, se jogou, foi o visitante
  d=cbind(d,g) #adiciona 20 colunas com variaveis
  #indicadoras para cada time como mandante
}

#-----FUNcao GLM-----
e=glm(d[,1]~factor(d[,2])+factor(d[,3])+d[,5]+
  d[,6]+d[,7]+d[,8]+d[,9]+d[,10]+d[,11]+
  d[,12]+d[,13]+d[,14]+d[,15]+d[,16]+
  d[,17]+d[,18]+d[,19]+d[,20]+d[,21]+
  d[,22]+d[,23],family=poisson) #modelo

#-----ESTIMATIVAS GLM-----
f=summary(e)$coef[-1,]
GLM=matrix(f[,1],(n-1),3) #matriz com as estimativas
#da GLM
GLM=rbind(0,GLM)
GLM[,2]=-GLM[,2]
GLM=cbind(GLM,GLM[,2])
GLM=GLM[, -2]
rownames(GLM)=times

```

```
colnames (GLM)=fatores
```

```
#-----LEITURA DO CODA-----
```

```
n1=1000#tamanho das amostras das posteriores
n2=3#numero de cadeias
f=list()#lista com as matrizes das cadeias
g=list()
for (h in 1:n2){
  file=file.choose()
  g[[h]]=read.table(file)[[2]]
}
h=t(matrix(unlist(g),n2,length(g[[1]]),byrow=TRUE))
i=list()
j=(n-1)*n1
for (k in 1:3){
  i[[k]]=h[((j*(k-1)+1):(j*k)),]
}
i[[4]]=h[((3*j+1):length(h[,1])),]
f=i
g=list()#lista com as amostras para cada um dos
#fatores
for (h in 1:3){
  g[[h]]=list()#lista com as amostras de cada
  #time para um fator
  g[[h]][[1]]=rep(0,(n1*n2))#a amostra do primeiro
  #time e composta de zeros
  for (i in 2:n){
    g[[h]][[i]]=as.vector(f[[h]][((i-2)*n1+1):
                                                                    ((i-1)*n1),])
  }
}
g[[4]]=as.vector(f[[4]])
```



```

#-----ESTIMATIVAS DE BAYES-----

EB=matrix(NA,n,3)#matriz com as estimativas de
#Bayes para cada fator
for(h in 1:3){
  i=lapply(g[[h]],FUN=mean)
  EB[,h]=c(do.call("cbind",i))
}
rownames(EB)=times
colnames(EB)=fatores

#-----PREDICOES PARA JOGOS INDIVIDUAIS-----

file=file.choose()
j=read.csv(file,header=T)#jogos que faltam
n4=length(j[[1]])#numero de linhas no arquivo
k=grep(";",j[[1]],value=TRUE)#tira os "levels"
l=strsplit(k,";")#quebra a linha do arquivo
m=matrix(NA,n4,3)
for(o in 2:4){
  p=function(x){
    x[[o]]
  }
  q=lapply(l,FUN=p)
  s=c(do.call("cbind",q))
  m[, (o-1)]=s
}
k=m#matriz com os jogos que faltam
l=match(k[,1],times)#indices dos mandantes
m=match(k[,2],times)#indices dos visitantes
o=paste(k[,1], 'x', k[,2], sep=' ')
k[,1]=l
k[,2]=m
k=matrix(as.numeric(k),n4,length(k[,1]))

```

```

rownames(k)=o
colnames(k)=c('Mandante_(M)', 'Visitante_(V)', 'Rodada')
l=matrix(NA, (n1*n2), (2*n4)) #matriz com os
#resultados preditos para cada jogo
for(m in 1:(n4)){
  o=k[m,1] #mandante
  p=k[m,2] #visitante
  q=g[[1]][[o]]-g[[3]][[p]]+g[[2]][[o]]+g[[4]]
  q=exp(q) #lambda do mandante
  s=g[[1]][[p]]-g[[3]][[o]]+g[[4]]
  s=exp(s) #lambda do visitante
  t=rpois((n1*n2), lambda=q) #gols feitos pelo mandante
  u=rpois((n1*n2), lambda=s) #gols feitos pelo visitante
  l[, (2*m-1)]=t
  l[, (2*m)]=u
}
m=paste(rownames(k), '_M', sep='')
o=paste(rownames(k), '_V', sep='')
p=rbind(m, o)
m=as.vector(p)
colnames(l)=m
probs=matrix(NA, n4, 5) #matriz com as probabilidades
#dos resultados
m=apply(l, FUN=sum, 2)
m=as.vector(m)
m=matrix(m, n4, 2, byrow=TRUE) #numero de gols feitos,
#juntando todos os confrontos ficticios
probs[, 4:5]=m/(n1*n2)
m=seq(from=1, to=(2*n4), by=2) #gols dos mandantes
o=seq(from=2, to=(2*n4), by=2) #gols dos visitantes
p=l[,m]-l[,o] #saldos de gols dos mandantes
q=p>0 #vitorias dos mandantes
s=p==0 #empates
t=p<0 #vitorias dos visitantes

```

```

u=apply(q, FUN=sum, 2)
v=apply(s, FUN=sum, 2)
w=apply(t, FUN=sum, 2)
probs[,1]=u/(n1*n2)
probs[,2]=v/(n1*n2)
probs[,3]=w/(n1*n2)
colnames(probs)=c('V','E','D','GF','GS')
rownames(probs)=rownames(k)

```

#-----PREDICOES PARA O CAMPEONATO-----

```

x=3*q+s#pontuacoes dos mandantes
y=3*t+s#pontuacoes dos visitantes
z=matrix(NA,n,24)#tabela do campeonato
for(a1 in 1:n){
  b1=(d[,2]==a1)*c(1:(length(d[,1])))#linhas
  #dos jogos em que o time participou
  c1=d[b1,1]#gols feitos pelo time
  d1=(d[,3]==a1)*c(1:(length(d[,1])))#linhas
  #dos jogos em que o time participou
  d1=c(d1[(n3+1):(2*n3)],d1[1:n3])
  e1=d[d1,1]#gols sofridos pelo time
  f1=c1-e1#saldos de gols do time
  g1=f1>0#vitorias do time
  h1=f1==0#empates do time
  i1=f1<0#derrotas do time
  z[a1,2]=sum(g1)#numero de vitorias
  z[a1,3]=sum(h1)#numero de empates
  z[a1,4]=sum(i1)#numero de derrotas
  z[a1,8]=z[a1,2]*3+z[a1,3]#numero de pontos
  z[a1,1]=sum(z[a1,2:4])#numero de jogos
  z[a1,5]=sum(c1)#numero de gols feitos
  z[a1,6]=sum(e1)#numero de gols sofridos
  z[a1,7]=sum(f1)#saldo de gols

```

```

j1=b1[1:n3]#linhas dos jogos em que o time
#participou, em casa
k1=d[j1,1]#gols feitos pelo time, em casa
l1=d1[1:n3]#linhas dos jogos em que o time
#participou, em casa
m1=d[l1,1]#gols sofridos pelo time, em casa
o1=k1-m1#saldos de gols do time, em casa
p1=o1>0#vitorias do time, em casa
q1=o1==0#empates do time, em casa
s1=o1<0#derrotas do time, em casa
z[a1,10]=sum(p1)#numero de vitorias em casa
z[a1,11]=sum(q1)#numero de empates em casa
z[a1,12]=sum(s1)#numero de derrotas em casa
z[a1,16]=z[a1,10]*3+z[a1,11]#numero de pontos
#em casa
z[a1,9]=sum(z[a1,10:12])#numero de jogos
#em casa
z[a1,13]=sum(k1)#numero de gols feitos
#em casa
z[a1,14]=sum(m1)#numero de gols sofridos
#em casa
z[a1,15]=sum(o1)#saldo de gols em casa
t1=b1[(n3+1):(2*n3)]#linhas dos jogos em que
#o time participou, fora de casa
u1=d[t1,1]#gols feitos pelo time,
#fora de casa
v1=d1[(n3+1):(2*n3)]#linhas dos jogos em que
#o time participou, fora de casa
w1=d[v1,1]#gols sofridos pelo time,
#fora de casa
x1=u1-w1#saldos de gols do time,
#fora de casa
y1=x1>0#vitorias do time, fora de casa
z1=x1==0#empates do time, fora de casa

```

```

a2=x1<0#derrotas do time, fora de casa
z[a1,18]=sum(y1)#numero de vitorias
#fora de casa
z[a1,19]=sum(z1)#numero de empates
#fora de casa
z[a1,20]=sum(a2)#numero de derrotas
#fora de casa
z[a1,24]=z[a1,18]*3+z[a1,19]#numero
#de pontos fora de casa
z[a1,17]=sum(z[a1,18:20])#numero de
#jogos fora de casa
z[a1,21]=sum(u1)#numero de gols
#feitos fora de casa
z[a1,22]=sum(w1)#numero de gols
#sofridos fora de casa
z[a1,23]=sum(x1)#saldo de gols
#fora de casa
}
rownames(z)=times
a1=c('J','V','E','D','GF','GS','SG','P')
b1=paste(a1,'_C',sep='')
c1=paste(a1,'_F',sep='')
colnames(z)=c(a1,b1,c1)
tabela=z#tabela do campeonato
a1=k[,1]#mandantes das partidas a serem
#previstas
b1=k[,2]#visitantes das partidas a serem
#previstas
c1=seq(1,2*n4,by=2)#numeros impares ate 2*n4
d1=seq(2,2*n4,by=2)#numeros pares ate 2*n4
e1=c()
e1[(c1)]=a1
e1[d1]=b1
z1=list()#lista com as colunas da matriz

```

```

#de gols feitos e sofridos para o resto
#do campeonato, correspondentes a cada time
z2=list()#lista com as colunas da matriz
#de gols feitos e sofridos previstos para
#o campeonato todo, correspondentes a
#cada time
z3=list()#lista com as colunas da matriz
#de resultados previstos para o resto do
#campeonato, correspondentes a cada time
for(h in 1:n){
  z1[[h]]=list()
  z2[[h]]=list()
  z3[[h]]=list()
  a1=(e1==h)*c(1:(2*n4))
  a1=a1[a1!=0]#colunas correspondentes ao
  #h-esimo time
  b1=a1[a1%%2==0]#colunas correspondentes
  #aos jogos em que foi o visitante
  c1=a1[a1%%2==1]#colunas correspondentes
  #aos jogos em que foi o mandante
  b2=b1-1#colunas correspondentes aos
  #mandantes, quando foi o visitante
  c2=c1+1#colunas correspondentes aos
  #visitantes, quando foi o mandante
  if(length(c1)>1){
    z1[[h]][[1]]=1[,c1]#colunas com os
    #gols que fez como mandante
    z2[[h]][[1]]=apply(FUN=sum, z1[[h]]
                        [[1]], 1)+z[h,13]#colunas
    #com os gols que tera feito como mandante,
    #ao final do campeonato
    z1[[h]][[2]]=1[,c2]#colunas com os gols
    #que sofreu como mandante
    z2[[h]][[2]]=apply(FUN=sum, z1[[h]]

```

```

[[2]],1)+z[h,14]#colunas
#com os gols que tera sofrido como mandante,
#ao final do campeonato
}
if(length(c1)==1){
  z1[[h]][[1]]=l[,c1]#colunas com os gols que
  #fez como mandante
  z2[[h]][[1]]=z1[[h]][[1]]+z[h,13]#colunas
  #com os gols que tera feito como mandante,
  #ao final do campeonato
  z1[[h]][[2]]=l[,c2]#colunas com os gols
  #que sofreu como mandante
  z2[[h]][[2]]=z1[[h]][[2]]+z[h,14]#colunas
  #com os gols que tera sofrido como mandante,
  #ao final do campeonato
}
if(length(c1)==0){
  z1[[h]][[1]]=rep(0,n1*n2)#colunas com os
  #gols que fez como mandante
  z2[[h]][[1]]=z1[[h]][[1]]+z[h,13]#colunas
  #com os gols que tera feito como mandante, ao final do
  z1[[h]][[2]]=rep(1,n1*n2)#colunas com os
  #gols que sofreu como mandante
  z2[[h]][[2]]=z1[[h]][[2]]+z[h,14]-1#colunas
  #com os gols que tera sofrido como mandante,
  #ao final do campeonato
}
if(length(b1)>1){
  z1[[h]][[3]]=l[,b1]#colunas com os gols
  #que fez como visitante
  z2[[h]][[3]]=apply(FUN=sum, z1[[h]]
  [[3]],1)+z[h,21]#colunas
  #com os gols que tera feito como visitante,
  #ao final do campeonato

```

```

z1[[h]][[4]]=l[,b2]#colunas com os gols
#que sofreu como visitante
z2[[h]][[4]]=apply(FUN=sum, z1[[h]]
                    [[4]],1)+z[h,22]#colunas
#com os gols que tera sofrido como visitante,
#ao final do campeonato
}
if(length(b1)==1){
  z1[[h]][[3]]=l[,b1]#colunas com os gols
  #que fez como visitante
  z2[[h]][[3]]=z1[[h]][[3]]+z[h,21]#colunas
  #com os gols que tera feito como visitante,
  #ao final do campeonato
  z1[[h]][[4]]=l[,b2]#colunas com os gols
  #que sofreu como visitante
  z2[[h]][[4]]=z1[[h]][[4]]+z[h,22]#colunas
  #com os gols que tera sofrido como visitante,
  #ao final do campeonato
}
if(length(b1)==0){
  z1[[h]][[3]]=rep(0,n1*n2)#colunas com os
  #gols que fez como visitante
  z2[[h]][[3]]=z1[[h]][[3]]+z[h,21]#colunas
  #com os gols que tera feito como visitante,
  #ao final do campeonato
  z1[[h]][[4]]=rep(1,n1*n2)#colunas com os
  #gols que sofreu como visitante
  z2[[h]][[4]]=z1[[h]][[4]]+z[h,22]-1#colunas
  #com os gols que tera sofrido como visitante,
  #ao final do campeonato
}
z3[[h]][[1]]=((z1[[h]][[1]]-z1[[h]][[2]])>0)*3+
  ((z1[[h]][[1]]-z1[[h]][[2]])==0)#colunas com
#os resultados como mandante, ate o final do

```



```

#campeonato
z3[[h]][[2]]=((z1[[h]][[3]]-z1[[h]][[4]])>0)*3+
  ((z1[[h]][[3]]-z1[[h]][[4]])==0)#colunas com
#os resultados como visitante, ate o final do
#campeonato
}
Probs[((r-6)*(n/2)+1):((r-5)*(n/2)),]=probs[1:10,]
linhas=append(linhas,rownames(probs)[1:10])
z4=list()#lista das tabelas para cada valor
#da amostra dos parametros
z5=matrix(0,n,(n1*n2))#matriz com as posicoes
#finais do time para cada valor da amostra
#dos parametros
for(h in 1:(n1*n2)){
  z4[[h]]=z#matriz com a tabela para o valor
#especifico da amostra
  for(f1 in 1:n){
    if(is.matrix(z1[[f1]][[1]]))==T){
      z4[[h]][f1,9]=z4[[h]][f1,9]+
        length(z1[[f1]][[1]][1,])#jogos em
#casa ao final do campeonato
      z4[[h]][f1,10]=z4[[h]][f1,10]+
        sum(z3[[f1]][[1]][h,]==3)#vitorias
#em casa ao final do campeonato
      z4[[h]][f1,11]=z4[[h]][f1,11]+
        sum(z3[[f1]][[1]][h,]==1)#empates
#em casa ao final do campeonato
      z4[[h]][f1,12]=z4[[h]][f1,12]+
        sum(z3[[f1]][[1]][h,]==0)#derrotas
#em casa ao final do campeonato
      z4[[h]][f1,16]=z4[[h]][f1,16]+
        sum(z3[[f1]][[1]][h,])#saldo de gols
#em casa ao final do campeonato
    }
  }
}

```

```

if(is.matrix(z1[[f1]][[1]])==F &
    sum(z1[[f1]][[1]])>0){
    z4[[h]][f1,9]=19#jogos em casa ao
    final do campeonato
    z4[[h]][f1,10]=z4[[h]][f1,10]+
        (z3[[f1]][[1]][h]==3)#vitorias
    #em casa ao final do campeonato
    z4[[h]][f1,11]=z4[[h]][f1,11]+
        (z3[[f1]][[1]][h]==1)#empates
    #em casa ao final do campeonato
    z4[[h]][f1,12]=z4[[h]][f1,12]+
        (z3[[f1]][[1]][h]==0)#derrotas
    #em casa ao final do campeonato
    z4[[h]][f1,16]=z4[[h]][f1,16]+
        z3[[f1]][[1]][h]#saldo de gols
    #em casa ao final do campeonato
}

if(sum(z1[[f1]][[1]])==0){
    z4[[h]][f1,9]=19#jogos em casa ao final
    #do campeonato
    z4[[h]][f1,10]=z4[[h]][f1,10]#vitorias
    #em casa ao final do campeonato
    z4[[h]][f1,11]=z4[[h]][f1,11]#empates
    #em casa ao final do campeonato
    z4[[h]][f1,12]=z4[[h]][f1,12]#derrotas
    #em casa ao final do campeonato
    z4[[h]][f1,16]=z4[[h]][f1,16]#saldo de gols
    #em casa ao final do campeonato
}

z4[[h]][f1,13]=z2[[f1]][[1]][h]#gols feitos
#em casa ao final do campeonato
z4[[h]][f1,14]=z2[[f1]][[2]][h]#gols
#sofridos em casa ao final do campeonato
z4[[h]][f1,15]=z4[[h]][f1,13]-z4[[h]][f1,14]

```

```

#saldo de gols em casa ao final do campeonato
if(is.matrix(z1[[f1]][[3]])==T){
  z4[[h]][f1,17]=z4[[h]][f1,17]+
    length(z1[[f1]][[3]][1,])#jogos
  #fora de casa ao final do campeonato
  z4[[h]][f1,18]=z4[[h]][f1,18]+
    sum(z3[[f1]][[2]][h,]==3)#vitorias
  #fora de casa ao final do campeonato
  z4[[h]][f1,19]=z4[[h]][f1,19]+
    sum(z3[[f1]][[2]][h,]==1)#empates
  #fora de casa ao final do campeonato
  z4[[h]][f1,20]=z4[[h]][f1,20]+
    sum(z3[[f1]][[2]][h,]==0)#derrotas
  #fora de casa ao final do campeonato
  z4[[h]][f1,24]=z4[[h]][f1,24]+
    sum(z3[[f1]][[2]][h,])#saldo de gols
  #fora de casa ao final do campeonato
}
if(is.matrix(z1[[f1]][[3]])==F &
  sum(z1[[f1]][[3]])>0){
  z4[[h]][f1,17]=19#jogos fora
  #de casa ao final do campeonato
  z4[[h]][f1,18]=z4[[h]][f1,18]+
    (z3[[f1]][[2]][h]==3)#vitorias
  #fora de casa ao final do campeonato
  z4[[h]][f1,19]=z4[[h]][f1,19]+
    (z3[[f1]][[2]][h]==1)#empates
  #fora de casa ao final do campeonato
  z4[[h]][f1,20]=z4[[h]][f1,20]+
    (z3[[f1]][[2]][h]==0)#derrotas
  #fora de casa ao final do campeonato
  z4[[h]][f1,24]=z4[[h]][f1,24]+
    z3[[f1]][[2]][h]#saldo de gols
  #fora de casa ao final do campeonato

```

```

}
if (sum (z1 [[f1]] [[3]]) == 0) {
  z4 [[h]] [f1, 17] = 19 # jogos em casa ao final
  # do campeonato
  z4 [[h]] [f1, 18] = z4 [[h]] [f1, 18]
  # vitórias em casa ao final do campeonato
  z4 [[h]] [f1, 19] = z4 [[h]] [f1, 19]
  # empates em casa ao final do campeonato
  z4 [[h]] [f1, 20] = z4 [[h]] [f1, 20]
  # derrotas em casa ao final do campeonato
  z4 [[h]] [f1, 24] = z4 [[h]] [f1, 24]
  # saldo de gols em casa ao final do campeonato
}
z4 [[h]] [f1, 21] = z2 [[f1]] [[3]] [h] # gols
# feitos fora de casa ao final do campeonato
z4 [[h]] [f1, 22] = z2 [[f1]] [[4]] [h] # gols
# sofridos fora de casa ao final do campeonato
z4 [[h]] [f1, 23] = z4 [[h]] [f1, 21] - z4 [[h]] [f1, 22]
# saldo de gols fora de casa ao final do campeonato
for (g1 in 1:8) {
  z4 [[h]] [f1, g1] = z4 [[h]] [f1, g1+8] +
    z4 [[h]] [f1, g1+16]
}
}
f1 = z4 [[h]] [order (z4 [[h]] [, 8],
                      decreasing=T), ]
g1 = match (times, rownames (f1))
z5 [, h] = g1
POS [[r]] = z5
}
}
colnames (Probs) = colnames (probs)
rownames (Probs) = linhas
pos = list ()

```

```

for (r in 5:37) {
  pos[[r]]=matrix(0,n,n)
  colnames(pos[[r]])=1:20
  rownames(pos[[r]])=times
  for (a in 1:n) {
    for (b in 1:n) {
      pos[[r]][a,b]=sum(POS[[r]][a,]==b)
    }
  }
  pos[[r]]=pos[[r]]/(n1*n2)
}

```

Para o modelo dinâmico, os códigos são semelhantes:

```
#-----DADOS-----
Probs=list()
Prec=list()
pos=list()
POS=list()
atu=c(5,8,10,13,19)
times=file.choose()
fatores=c("Ataque",'Campo','Defesa')
n=length(times)
A=list()
B=list()
for(precn in 1:4){
  Probs[[precn]]=matrix(NA,160,5)
  POS[[precn]]=list()
  linhas=c()
  Prec[[precn]]=list()
  for(r in 22:37){
    file=file.choose()
    a=read.csv(file,header=T)#dados brutos
    r1=atu[precn]#numero de rodadas para a atualizacao dos pa
    r2=ceiling((r/r1))
    b=c()#vetor com os dados sem as datas dos jogos
    n3=length(a[[1]])#numero de linhas no arquivo
    for(c in 1:n3){
      d=grep(";",a[[1]][c],value=TRUE)#tira os "levels"
      e=strsplit(d,";")#quebra a linha do arquivo
      e[[1]][2]=gsub(pattern='',replacement='',
                     x=e[[1]][2])#tira o sinal
      #bizarro que pode aparecer no final do nome
      #do mandante
      e[[1]][3]=gsub(pattern='',replacement='',
                     x=e[[1]][3])#tira o sinal
```

```

    #bizarro que pode aparecer no final do nome
    #do visitante
    f=length(e[[1]])#quantidade de dados na linha
    b=append(b,e[[1]][2:f])
}

c=matrix(b,c,(f-1),byrow=TRUE)#matriz com os dados
d=strsplit(c[,3],"")#quebra as strings dos placares
c=c[,-3]#tira a coluna com os placares
c=cbind(c,unlist(lapply(d,'[[',1)),
        unlist(lapply(d,'[[',5)))#acrescenta 0's,
#para que se possa colocar o numero de gols
#feitos pelos times
d=match(c[,1],times)#mandantes como numeros
e=match(c[,2],times)#visitantes como numeros
f=as.numeric(c[,4:5])#gols num vetor
d=cbind(f,c(d,e),c(e,d))#matriz com os gols e os
#ataques e defesas responsaveis
for(e in 1:n){
    f=(match(c[,1],times)==e)#variavel indicadora para
    #se o e-esimo time e o mandante
    g=c(f,(f*0))#acrescenta-se zeros, para os jogos
    #em que o time, se jogou, foi o visitante
    d=cbind(d,g)#adiciona 20 colunas com variaveis
    #indicadoras para cada time como mandante
}

#-----FUNcao GLM-----
e=glm(d[,1]~factor(d[,2])+factor(d[,3])+d[,5]+
      d[,6]+d[,7]+d[,8]+d[,9]+d[,10]+d[,11]+
      d[,12]+d[,13]+d[,14]+d[,15]+d[,16]+
      d[,17]+d[,18]+d[,19]+d[,20]+d[,21]+
      d[,22]+d[,23],family=poisson)#modelo

#-----ESTIMATIVAS GLM-----

```

```

f=summary(e)$coef[-1,]
GLM=matrix(f[,1],(n-1),3)#matriz com as estimativas
#da GLM
GLM=rbind(0, GLM)
GLM[,2]=-GLM[,2]
GLM=cbind(GLM, GLM[,2])
GLM=GLM[, -2]
rownames(GLM)=times
colnames(GLM)=fatores

```

```

#-----LEITURA DO CODA-----

```

```

n1=1000#tamanho das amostras das posteriores
n2=3#numero de cadeias
f=list()#lista com as matrizes das cadeias
g=list()
for(h in 1:n2){
  file=file.choose()
  g[[h]]=read.table(file)[[2]]
}
h=t(matrix(unlist(g), n2, length(g[[1]]),
           byrow=TRUE))
i=list()
j=(n-1)*r2*n1
for(k in 1:3){
  i[[k]]=h[((j*(k-1)+1):(j*k)),]
}
i[[4]]=h[((3*j+1):(3*j+n1*r2)),]
i[[5]]=h[((3*j+n1*r2+1):(length(h[,1]))),]
k=list()
for(l in 1:r2){
  m=seq(1, (length(i[[1]])[,1])/n1), by=r2)
  o=c()
  for(p in 1:length(m)){

```



```

        o=append(o, c(((m[p]-1)*n1+1):(m[p]*n1)))
    }
    k[[1]]=o
}
l=list()
for(m in 1:3){
    l[[m]]=list()
    for(o in 1:r2){
        l[[m]][[o]]=i[[m]][k[[o]],]
    }
}
l[[4]]=list()
for(m in 1:r2){
    l[[4]][[m]]=i[[4]][((m-1)*n1+1):(m*n1)],]
}
l[[5]]=i[[5]][(1:n1),]
f=1
g=list()#lista com as amostras para cada
#um dos fatores
for(h in 1:r2){
    g[[h]]=list()#lista com as amostras para
#cada ponto de atualizacao
    for(i in 1:3){
        g[[h]][[i]]=list()#lista com as amostras
#de cada time para um fator
        g[[h]][[i]][[1]]=rep(0,(n1*n2))#a amostra

        #do primeiro time e composta de zeros
        for(j in 2:n){
            g[[h]][[i]][[j]]=as.vector(f[[i]][[h]]
                                         [((j-2)*n1+1):
                                         ((j-1)*n1)],)
        }
    }
}

```

```

    g[[h]][[4]]=as.vector(f[[4]][[h]])
  }
  g[[r2+1]]=as.vector(f[[5]])
  prec=g[[r2+1]]
  Prec[[precn]][[r]]=prec

```

```

#-----ESTIMATIVAS DE BAYES-----

```

```

EB=list()
for(h in 1:r2){
  EB[[h]]=matrix(NA,n,3)#matriz com
#as estimativas de Bayes para cada fator
  for(i in 1:3){
    j=lapply(g[[h]][[i]],FUN=mean)
    EB[[h]][,i]=c(do.call("cbind",j))
  }
  rownames(EB[[h]])=times
  colnames(EB[[h]])=fatores
}

```

```

#-----PREDICOES PARA JOGOS INDIVIDUAIS-----

```

```

file=file.choose()
j=read.csv(file,header=T)#jogos que faltam
n4=length(j[[1]])#numero de linhas no arquivo
k=grep(";",j[[1]],value=TRUE)#tira os "levels"
l=strsplit(k,";")#quebra a linha do arquivo
m=matrix(NA,n4,3)
for(o in 2:4){
  p=function(x){
    x[[o]]
  }
  q=lapply(l,FUN=p)
  s=c(do.call("cbind",q))
  m[, (o-1)]=s
}

```

```

}
k=m#matriz com os jogos que faltam
l=match(k[,1],times)#indices dos mandantes
m=match(k[,2],times)#indices dos visitantes
o=paste(k[,1], 'x', k[,2], sep='')
k[,1]=l
k[,2]=m
k=matrix(as.numeric(k), n4, length(k[1,]))
rownames(k)=o
colnames(k)=c('Mandante_(M)', 'Visitante_(V)', 'Rodada')
l=matrix(NA, (n1*n2), (2*n4))#matriz com os resultados preo
for(m in 1:(n4)){
  r3=r+ceiling(m*2/n)
  r4=ceiling(r3/r1)
  r5=r4-r2
  o=k[m,1]#mandante
  p=k[m,2]#visitante
  q=g[[r2]][[1]][[o]]-g[[r2]][[3]][[p]]+
    g[[r2]][[2]][[o]]+g[[r2]][[4]]
  s=g[[r2]][[1]][[p]]-g[[r2]][[3]][[o]]+
    g[[r2]][[4]]
  if(r5>0){
    for(r6 in 1:r5){
      q=q+rnorm(n1*n2, mean=0,
                sd=g[[r2+1]]^(-0.5))
      s=s+rnorm(n1*n2, mean=0,
                sd=g[[r2+1]]^(-0.5))
    }
  }
}
q=exp(q)#lambda do mandante
s=exp(s)#lambda do visitante
t=rpois((n1*n2), lambda=q)#gols
#feitos pelo mandante
u=rpois((n1*n2), lambda=s)#gols

```

```

    #feitos pelo visitante
    l[, (2*m-1)] = t
    l[, (2*m)] = u
}
m=paste(rownames(k), ' - M', sep='')
o=paste(rownames(k), ' - V', sep='')
p=rbind(m, o)
m=as.vector(p)
colnames(l) = m
probs=matrix(NA, n4, 5) #matriz com as
#probabilidades dos resultados
m=apply(l, FUN=sum, 2)
m=as.vector(m)
m=matrix(m, n4, 2, byrow=TRUE) #numero de
#gols feitos, juntando todos os confrontos ficticios
probs[, 4:5] = m / (n1*n2)
m=seq(from=1, to=(2*n4), by=2) #gols dos
#mandantes
o=seq(from=2, to=(2*n4), by=2) #gols dos
#visitantes
p=l[, m] - l[, o] #saldos de gols dos mandantes
q=p>0 #vitorias dos mandantes
s=p==0 #empates
t=p<0 #vitorias dos visitantes
u=apply(q, FUN=sum, 2)
v=apply(s, FUN=sum, 2)
w=apply(t, FUN=sum, 2)
probs[, 1] = u / (n1*n2)
probs[, 2] = v / (n1*n2)
probs[, 3] = w / (n1*n2)
colnames(probs) = c('V', 'E', 'D', 'GF', 'GS')
rownames(probs) = rownames(k)

#-----PREDICOES PARA O CAMPEONATO-----

```

```

x=3*q+s#pontuacoes dos mandantes
y=3*t+s#pontuacoes dos visitantes
z=matrix(NA,n,24)#tabela do campeonato
for(a1 in 1:n){
  b1=(d[,2]==a1)*c(1:(length(d[,1])))#linhas
  #dos jogos em que o time participou
  c1=d[b1,1]#gols feitos pelo time
  d1=(d[,3]==a1)*c(1:(length(d[,1])))#linhas
  #dos jogos em que o time participou
  d1=c(d1[(n3+1):(2*n3)],d1[1:n3])
  e1=d[d1,1]#gols sofridos pelo time
  f1=c1-e1#saldos de gols do time
  g1=f1>0#vitorias do time
  h1=f1==0#empates do time
  i1=f1<0#derrotas do time
  z[a1,2]=sum(g1)#numero de vitorias
  z[a1,3]=sum(h1)#numero de empates
  z[a1,4]=sum(i1)#numero de derrotas
  z[a1,8]=z[a1,2]*3+z[a1,3]#numero de pontos
  z[a1,1]=sum(z[a1,2:4])#numero de jogos
  z[a1,5]=sum(c1)#numero de gols feitos
  z[a1,6]=sum(e1)#numero de gols sofridos
  z[a1,7]=sum(f1)#saldo de gols
  j1=b1[1:n3]#linhas dos jogos em que o time
  #participou, em casa
  k1=d[j1,1]#gols feitos pelo time, em casa
  l1=d1[1:n3]#linhas dos jogos em que o time
  #participou, em casa
  m1=d[l1,1]#gols sofridos pelo time, em casa
  o1=k1-m1#saldos de gols do time, em casa
  p1=o1>0#vitorias do time, em casa
  q1=o1==0#empates do time, em casa
  s1=o1<0#derrotas do time, em casa
}

```

```

z[a1,10]=sum(p1)#numero de vitorias em casa
z[a1,11]=sum(q1)#numero de empates em casa
z[a1,12]=sum(s1)#numero de derrotas em casa
z[a1,16]=z[a1,10]*3+z[a1,11]#numero de pontos
#em casa
z[a1,9]=sum(z[a1,10:12])#numero de jogos
#em casa
z[a1,13]=sum(k1)#numero de gols feitos
#em casa
z[a1,14]=sum(m1)#numero de gols sofridos
#em casa
z[a1,15]=sum(o1)#saldo de gols em casa
t1=b1[(n3+1):(2*n3)]#linhas dos jogos em que
#o time participou, fora de casa
u1=d[t1,1]#gols feitos pelo time,
#fora de casa
v1=d1[(n3+1):(2*n3)]#linhas dos jogos em que
#o time participou, fora de casa
w1=d[v1,1]#gols sofridos pelo time,
#fora de casa
x1=u1-w1#saldos de gols do time,
#fora de casa
y1=x1>0#vitorias do time, fora de casa
z1=x1==0#empates do time, fora de casa
a2=x1<0#derrotas do time, fora de casa
z[a1,18]=sum(y1)#numero de vitorias
#fora de casa
z[a1,19]=sum(z1)#numero de empates
#fora de casa
z[a1,20]=sum(a2)#numero de derrotas
#fora de casa
z[a1,24]=z[a1,18]*3+z[a1,19]#numero
#de pontos fora de casa
z[a1,17]=sum(z[a1,18:20])#numero de

```

```

#jogos fora de casa
z[a1,21]=sum(u1)#numero de gols
#feitos fora de casa
z[a1,22]=sum(w1)#numero de gols
#sofridos fora de casa
z[a1,23]=sum(x1)#saldo de gols
#fora de casa
}

rownames(z)=times
a1=c('J','V','E','D','GF','GS','SG','P')
b1=paste(a1,'_C',sep='')
c1=paste(a1,'_F',sep='')
colnames(z)=c(a1,b1,c1)
tabela=z#tabela do campeonato
a1=k[,1]#mandantes das partidas a serem
#previstas
b1=k[,2]#visitantes das partidas a serem
#previstas
c1=seq(1,2*n4,by=2)#numeros impares ate 2*n4
d1=seq(2,2*n4,by=2)#numeros pares ate 2*n4
e1=c()
e1[(c1)]=a1
e1[d1]=b1
z1=list()#lista com as colunas da matriz
#de gols feitos e sofridos para o resto
#do campeonato, correspondentes a cada time
z2=list()#lista com as colunas da matriz
#de gols feitos e sofridos previstos para
#o campeonato todo, correspondentes a
#cada time
z3=list()#lista com as colunas da matriz
#de resultados previstos para o resto do
#campeonato, correspondentes a cada time
for(h in 1:n){

```

```

z1[[h]]=list()
z2[[h]]=list()
z3[[h]]=list()
a1=(e1==h)*c(1:(2*n4))
a1=a1[a1!=0]#colunas correspondentes ao
#h-esimo time
b1=a1[a1%%2==0]#colunas correspondentes
#aos jogos em que foi o visitante
c1=a1[a1%%2==1]#colunas correspondentes
#aos jogos em que foi o mandante
b2=b1-1#colunas correspondentes aos
#mandantes, quando foi o visitante
c2=c1+1#colunas correspondentes aos
#visitantes, quando foi o mandante
if(length(c1)>1){
  z1[[h]][[1]]=l[,c1]#colunas com os
#gols que fez como mandante
  z2[[h]][[1]]=apply(FUN=sum, z1[[h]]
    [[1]], 1)+z[h,13]#colunas
#com os gols que tera feito como mandante,
#ao final do campeonato
  z1[[h]][[2]]=l[,c2]#colunas com os gols
#que sofreu como mandante
  z2[[h]][[2]]=apply(FUN=sum, z1[[h]]
    [[2]], 1)+z[h,14]#colunas
#com os gols que tera sofrido como mandante,
#ao final do campeonato
}
if(length(c1)==1){
  z1[[h]][[1]]=l[,c1]#colunas com os gols que
#fez como mandante
  z2[[h]][[1]]=z1[[h]][[1]]+z[h,13]#colunas
#com os gols que tera feito como mandante,
#ao final do campeonato

```



```

z1[[h]][[2]]=l[,c2]#colunas com os gols
#que sofreu como mandante
z2[[h]][[2]]=z1[[h]][[2]]+z[h,14]#colunas
#com os gols que tera sofrido como mandante,
#ao final do campeonato
}
if(length(c1)==0){
  z1[[h]][[1]]=rep(0,n1*n2)#colunas com os
  #gols que fez como mandante
  z2[[h]][[1]]=z1[[h]][[1]]+z[h,13]#colunas
  #com os gols que tera feito como mandante, ao final o
  z1[[h]][[2]]=rep(1,n1*n2)#colunas com os
  #gols que sofreu como mandante
  z2[[h]][[2]]=z1[[h]][[2]]+z[h,14]-1#colunas
  #com os gols que tera sofrido como mandante,
  #ao final do campeonato
}
if(length(b1)>1){
  z1[[h]][[3]]=l[,b1]#colunas com os gols
  #que fez como visitante
  z2[[h]][[3]]=apply(FUN=sum,z1[[h]]
                      [[3]],1)+z[h,21]#colunas
  #com os gols que tera feito como visitante,
  #ao final do campeonato
  z1[[h]][[4]]=l[,b2]#colunas com os gols
  #que sofreu como visitante
  z2[[h]][[4]]=apply(FUN=sum,z1[[h]]
                      [[4]],1)+z[h,22]#colunas
  #com os gols que tera sofrido como visitante,
  #ao final do campeonato
}
if(length(b1)==1){
  z1[[h]][[3]]=l[,b1]#colunas com os gols
  #que fez como visitante

```

```

z2[[h]][[3]]=z1[[h]][[3]]+z[h,21]#colunas
#com os gols que tera feito como visitante,
#ao final do campeonato
z1[[h]][[4]]=1[,b2]#colunas com os gols
#que sofreu como visitante
z2[[h]][[4]]=z1[[h]][[4]]+z[h,22]#colunas
#com os gols que tera sofrido como visitante,
#ao final do campeonato
}
if(length(b1)==0){
  z1[[h]][[3]]=rep(0,n1*n2)#colunas com os
  #gols que fez como visitante
  z2[[h]][[3]]=z1[[h]][[3]]+z[h,21]#colunas
  #com os gols que tera feito como visitante,
  #ao final do campeonato
  z1[[h]][[4]]=rep(1,n1*n2)#colunas com os
  #gols que sofreu como visitante
  z2[[h]][[4]]=z1[[h]][[4]]+z[h,22]-1#colunas
  #com os gols que tera sofrido como visitante,
  #ao final do campeonato
}
z3[[h]][[1]]=((z1[[h]][[1]]-z1[[h]][[2]])>0)*3+
  ((z1[[h]][[1]]-z1[[h]][[2]])==0)#colunas com
#os resultados como mandante, ate o final do
#campeonato
z3[[h]][[2]]=((z1[[h]][[3]]-z1[[h]][[4]])>0)*3+
  ((z1[[h]][[3]]-z1[[h]][[4]])==0)#colunas com
#os resultados como visitante, ate o final do
#campeonato
}
Probs[[precn]][((r-22)*(n/2)+1):
  ((r-21)*(n/2)),]=
  probs[1:10,]
linhas=append(linhas,rownames(probs)[1:10])

```

```

z4=list()#lista das tabelas para cada
#valor da amostra dos parametros
z5=matrix(0,n,(n1*n2))#matriz com as
#posicoes finais do time para cada valor
#da amostra dos parametros
for(h in 1:(n1*n2)){
  z4[[h]]=z#matriz com a tabela para o valor
  #especifico da amostra
  for(f1 in 1:n){
    if(is.matrix(z1[[f1]][[1]]))==T){
      z4[[h]][f1,9]=z4[[h]][f1,9]+
        length(z1[[f1]][[1]][1,])#jogos em
#casa ao final do campeonato
      z4[[h]][f1,10]=z4[[h]][f1,10]+
        sum(z3[[f1]][[1]][h,]==3)#vitorias
#em casa ao final do campeonato
      z4[[h]][f1,11]=z4[[h]][f1,11]+
        sum(z3[[f1]][[1]][h,]==1)#empates
#em casa ao final do campeonato
      z4[[h]][f1,12]=z4[[h]][f1,12]+
        sum(z3[[f1]][[1]][h,]==0)#derrotas
#em casa ao final do campeonato
      z4[[h]][f1,16]=z4[[h]][f1,16]+
        sum(z3[[f1]][[1]][h,])#saldo de gols
#em casa ao final do campeonato
    }
    if(is.matrix(z1[[f1]][[1]]))==F &
      sum(z1[[f1]][[1]])>0){
      z4[[h]][f1,9]=19#jogos em casa ao
final do campeonato
      z4[[h]][f1,10]=z4[[h]][f1,10]+
        (z3[[f1]][[1]][h]==3)#vitorias
#em casa ao final do campeonato
      z4[[h]][f1,11]=z4[[h]][f1,11]+

```

```

        (z3[[f1]][[1]][h]==1) #empates
#em casa ao final do campeonato
z4[[h]][f1,12]=z4[[h]][f1,12]+
        (z3[[f1]][[1]][h]==0) #derrotas
#em casa ao final do campeonato
z4[[h]][f1,16]=z4[[h]][f1,16]+
        z3[[f1]][[1]][h] #saldo de gols
#em casa ao final do campeonato
}
if(sum(z1[[f1]][[1]])==0){
    z4[[h]][f1,9]=19 #jogos em casa ao final
#do campeonato
    z4[[h]][f1,10]=z4[[h]][f1,10] #vitorias
#em casa ao final do campeonato
    z4[[h]][f1,11]=z4[[h]][f1,11] #empates
#em casa ao final do campeonato
    z4[[h]][f1,12]=z4[[h]][f1,12] #derrotas
#em casa ao final do campeonato
    z4[[h]][f1,16]=z4[[h]][f1,16] #saldo de gols
#em casa ao final do campeonato
}
z4[[h]][f1,13]=z2[[f1]][[1]][h] #gols feitos
#em casa ao final do campeonato
z4[[h]][f1,14]=z2[[f1]][[2]][h] #gols
#sofridos em casa ao final do campeonato
z4[[h]][f1,15]=z4[[h]][f1,13]-z4[[h]][f1,14]
#saldo de gols em casa ao final do campeonato
if(is.matrix(z1[[f1]][[3]])==T){
    z4[[h]][f1,17]=z4[[h]][f1,17]+
        length(z1[[f1]][[3]][1,]) #jogos
#fora de casa ao final do campeonato
    z4[[h]][f1,18]=z4[[h]][f1,18]+
        sum(z3[[f1]][[2]][h,]==3) #vitorias
#fora de casa ao final do campeonato

```

```

z4[[h]][f1,19]=z4[[h]][f1,19]+
  sum(z3[[f1]][[2]][h,]==1) #empates
#fora de casa ao final do campeonato
z4[[h]][f1,20]=z4[[h]][f1,20]+
  sum(z3[[f1]][[2]][h,]==0) #derrotas
#fora de casa ao final do campeonato
z4[[h]][f1,24]=z4[[h]][f1,24]+
  sum(z3[[f1]][[2]][h,]) #saldo de gols
#fora de casa ao final do campeonato
}
if(is.matrix(z1[[f1]][[3]]))==F &
  sum(z1[[f1]][[3]])>0){
z4[[h]][f1,17]=19 #jogos fora
#de casa ao final do campeonato
z4[[h]][f1,18]=z4[[h]][f1,18]+
  (z3[[f1]][[2]][h]==3) #vitorias
#fora de casa ao final do campeonato
z4[[h]][f1,19]=z4[[h]][f1,19]+
  (z3[[f1]][[2]][h]==1) #empates
#fora de casa ao final do campeonato
z4[[h]][f1,20]=z4[[h]][f1,20]+
  (z3[[f1]][[2]][h]==0) #derrotas
#fora de casa ao final do campeonato
z4[[h]][f1,24]=z4[[h]][f1,24]+
  z3[[f1]][[2]][h] #saldo de gols
#fora de casa ao final do campeonato
}
if(sum(z1[[f1]][[3]])==0){
z4[[h]][f1,17]=19 #jogos em casa ao final
#do campeonato
z4[[h]][f1,18]=z4[[h]][f1,18]
#vitorias em casa ao final do campeonato
z4[[h]][f1,19]=z4[[h]][f1,19]
#empates em casa ao final do campeonato

```

```

        z4[[h]][f1,20]=z4[[h]][f1,20]
        #derrotas em casa ao final do campeonato
        z4[[h]][f1,24]=z4[[h]][f1,24]
        #saldo de gols em casa ao final do campeonato
    }
    z4[[h]][f1,21]=z2[[f1]][[3]][h]#gols
    #feitos fora de casa ao final do campeonato
    z4[[h]][f1,22]=z2[[f1]][[4]][h]#gols
    #sofridos fora de casa ao final do campeonato
    z4[[h]][f1,23]=z4[[h]][f1,21]-z4[[h]][f1,22]
    #saldo de gols fora de casa ao final do campeonato
    for(g1 in 1:8){
        z4[[h]][f1,g1]=z4[[h]][f1,g1+8]+
            z4[[h]][f1,g1+16]
    }
}
f1=z4[[h]][order(z4[[h]][,8],
                    decreasing=T),]
g1=match(times,rownames(f1))
z5[,h]=g1
POS[[precn]][[r]]=z5
}
}
rownames(Probs[[precn]])=linhas
colnames(Probs[[precn]])=colnames(probs)
pos[[precn]]=list()
for(R in 19:37){
    pos[[precn]][[R]]=matrix(0,n,n)
    colnames(pos[[precn]][[R]])=1:20
    rownames(pos[[precn]][[R]])=times
    for(a in 1:n){
        for(b in 1:n){
            pos[[precn]][[R]][a,b]=
                sum(POS[[precn]][[R]][a,]==b)
        }
    }
}

```

```

    }
}
pos[[precn]][[R]]=pos[[precn]]
[[R]]/(n1*n2)
}
}

```

Finalmente, os códigos para gerar as figuras e tabelas presentes neste estão a seguir:

```
##### GRAFICOS MODELO ESTATICO#####
```

```
times=file.choose()
siglas=file.choose()
n=length(times)
fatores=c('Ataque','Campo','Defesa')
r=38
n1=1000#tamanho das amostras das posteriores
n2=3#numero de cadeias
f=list()#lista com as matrizes das cadeias
g=list()
for(h in 1:n2){
  file=file.choose()
  g[[h]]=read.table(file)[[2]]
}
h=t(matrix(unlist(g),n2,length(g[[1]]),byrow=TRUE))
i=list()
j=(n-1)*n1
for(k in 1:3){
  i[[k]]=h[((j*(k-1)+1):(j*k)),]
}
i[[4]]=h[((3*j+1):length(h[,1])),]
f=i
g=list()#lista com as amostras para cada um dos fatores
for(h in 1:3){
  g[[h]]=list()#lista com as amostras de cada time para
  #um fator
  g[[h]][[1]]=rep(0,(n1*n2))#a amostra do primeiro time
  #e composta de zeros
  for(i in 2:n){
    g[[h]][[i]]=as.vector(f[[h]][((i-2)*n1+1):
                                ((i-1)*n1),])
```



```

    }
  }
g[[4]]=as.vector(f[[4]])
h=g
for(j in 1:3){
  k=h[[j]][[1]]
  for(l in 1:n){
    k=k+h[[j]][[1]]
  }
  k=k/n
  for(l in 1:n){
    h[[j]][[1]]=h[[j]][[1]]-k
  }
}
g=h
EB=matrix(NA,n,3)#matriz com as estimativas de Bayes
#para cada fator
for(h in 1:3){
  i=lapply(g[[h]],FUN=mean)
  EB[,h]=c(do.call("cbind",i))
}
rownames(EB)=times
colnames(EB)=fatores
LI=matrix(NA,n,3)#matriz com os limites inferiores
#das estimativas de Bayes para cada fator
a=function(x){
  return(sort(x)[75])
}
for(b in 1:3){
  i=unlist(lapply(g[[b]],FUN=a))
  LI[,b]=i
}
rownames(LI)=times
colnames(LI)=fatores

```

```

LS=matrix(NA,n,3)#matriz com os limites superiores
#das estimativas de Bayes para cada fator
b=function(x){
  return(sort(x)[2925])
}
for(c in 1:3){
  i=unlist(lapply(g[[c]],FUN=b))
  LS[,c]=i
}
rownames(LS)=times
colnames(LS)=fatores
col=file.choose()
ata=cbind(EB[,1],LI[,1],LS[,1],siglas,col)
cam=cbind(EB[,2],LI[,2],LS[,2],siglas,col)
def=cbind(EB[,3],LI[,3],LS[,3],siglas,col)
ata=ata[order(as.numeric(ata[,1])),]
cam=cam[order(as.numeric(cam[,1])),]
def=def[order(as.numeric(def[,1])),]
#Grafico dos EMVs do ataque
plot(as.numeric(ata[,1]),(n:1),main='EBs- Rodada 38',
      xlab='',ylab='',axes=F,ylim=c(0,20),xlim=c(-2,2),
      pch=21,col=ata[,5],bg=ata[,5])
for(c in 1:n){
  Axis(side=2,labels=ata[(n+1-c),4],
        col.axis=ata[(n+1-c),5],at=c,tick=F,las=1)
  d=as.numeric(ata[c,2])
  e=as.numeric(ata[c,3])
  lines(c(d,e),c((n+1-c),(n+1-c)),col=ata[c,5])
}
Axis(side=1,line=T,pos=-1,at=seq(-1.5,1.5,0.2),lty=8,
      cex.axis=0.5)
#Grafico dos EMVs do campo
plot(as.numeric(cam[,1]),(n:1),main='EBs- Rodada 38',
      xlab='',ylab='',axes=F,ylim=c(0,20),xlim=c(-2,2),

```

```

        pch=21, col=cam[, 5], bg=cam[, 5])
for(c in 1:n){
  Axis(side=2, labels=cam[(n+1-c), 4],
        col.axis=cam[(n+1-c), 5], at=c, tick=F, las=1)
  d=as.numeric(cam[c, 2])
  e=as.numeric(cam[c, 3])
  lines(c(d, e), c((n+1-c), (n+1-c)), col=cam[c, 5])
}
Axis(side=1, line=T, pos=-1, at=seq(-1.5, 1.5, 0.2),
      lty=8, cex.axis=0.5)
#Grafico dos EMVs da defesa
plot(as.numeric(def[, 1]), (n:1), main='EBs - Rodada 38',
      xlab='', ylab='', axes=F, ylim=c(0, 20), xlim=c(-2, 2),
      pch=21, col=def[, 5], bg=def[, 5])
for(c in 1:n){
  Axis(side=2, labels=def[(n+1-c), 4],
        col.axis=def[(n+1-c), 5], at=c, tick=F, las=1)
  d=as.numeric(def[c, 2])
  e=as.numeric(def[c, 3])
  lines(c(d, e), c((n+1-c), (n+1-c)), col=def[c, 5])
}
Axis(side=1, line=T, pos=-1, at=seq(-1.5, 1.5, 0.2),
      lty=8, cex.axis=0.5)

file=file.choose()
a=read.csv(file=file, header=T) #dados brutos
b=grep(";", a[[1]], value=TRUE) #vetor com os dados sem
#as datas dos jogos
c=strsplit(b, ";")
d=unlist(c)
e=matrix(d, length(b), length(c[[1]]), byrow=T)
f=paste(e[, 1], e[, 2], sep=' x ')

```

```

e=e[,c(-1,-2)]
g=as.numeric(e)
b=matrix(g,length(b),length(c[[1]])-2,byrow=F)
c=c("GFM","GFV","R","PVM┐┐CdG","PE┐┐CdG","PVV┐┐CdG",
     "PVM┐┐UFRJ","PE┐┐UFRJ","PVV┐┐UFRJ","PVM┐┐UFMG",
     "PE┐┐UFMG","PVV┐┐UFMG","PVM┐┐Simples",
     "PE┐┐Simples","PVV┐┐Simples")
colnames(b)=c
rownames(b)=f
c=b[,1:5]#matriz com as VPs por jogo
d=(b[,1]-b[,2]>0)#Vitorias do mandante
e=(b[,1]-b[,2]==0)#Empates
f=(b[,1]-b[,2]<0)#Vitorias do visitante
for(g in 1:5){
  if(g<4){
    c[,g]=d*b[, (g*3+1)]+e*b[, (g*3+2)]+f*b[, (g*3+3)]
  }
  if(g==4){
    c[,g]=1/3
  }
  if(g==5){
    c[,g]=d*b[, (g*3-2)]+e*b[, (g*3-1)]+f*b[, (g*3)]
  }
}
colnames(c)=c('VP-CdG','VP-UFRJ','VP-UFMG',
              'VP-Nulo','VP-Simples')
d=c*3#matriz com as VPs padronizadas por jogo
e=apply(c[1:10,],FUN=prod,2)#matriz com as VPs por rodada
for(r in 2:37){
  f=apply(c[(10*r-9):(10*r)],FUN=prod,2)
  e=rbind(e,f)
}
f=apply(c[371:379,],FUN=prod,2)
e=rbind(e,f)

```

```

rownames (e)=1:38
f=e#matriz com as VPs padronizadas por rodada
for (g in 1:5){
  f[,g]=e[,g]/e[,4]
}
mod=c('Simples','CdG','UFMG','UFRJ')
cor=c('green','blue','red','gold')
#Grafico das VPPs de todos os modelos, por jogo
plot (0,0,main='VPPs por Jogo',xlab='',ylab='',
      axes=F,ylim=c(0,12),xlim=c(1,379),type='l')
Axis (side=2,labels=mod[1],col.axis=cor[1],at=1,
      tick=F,las=1)
lines (1:379,d[,5],col=cor[1])
abline (a=1,b=0)
Axis (side=2,labels=mod[2],col.axis=cor[2],at=4,
      tick=F,las=1)
lines (1:379,d[,1]+3,col=cor[2])
abline (a=4,b=0)
Axis (side=2,labels=mod[3],col.axis=cor[3],at=7,
      tick=F,las=1)
lines (1:379,d[,3]+6,col=cor[3])
abline (a=7,b=0)
Axis (side=2,labels=mod[4],col.axis=cor[4],at=10,
      tick=F,las=1)
lines (1:379,d[,2]+9,col=cor[4])
abline (a=10,b=0)
Axis (side=1,line=T,pos=-1,at=seq(1,379,20),
      lty=8,cex.axis=0.5)
#Grafico das VPPs do modelo estatico, por jogo
plot (c(61:379),d[61:379,2],type='l',xlim=c(61,379),
      ylim=c(0,3),col=cor[4],ylab='VPP',xlab='Jogo',
      main='VPP da UFRJ por Jogo')
abline (a=1,b=0)
#Grafico das VPPs de todos os modelos, por rodada

```

```

plot (0,0,main='VPPs por Rodada',xlab='',ylab='',
      axes=F,ylim=c(0,168),xlim=c(1,38),type='l')
Axis(side=2,labels=mod[1],col.axis=cor[1],at=10,
      tick=F,las=1)
lines (1:38,f[,5]+9,col=cor[1])
abline (a=10,b=0)
Axis(side=2,labels=mod[2],col.axis=cor[2],at=52,
      tick=F,las=1)
lines (1:38,f[,1]+51,col=cor[2])
abline (a=52,b=0)
Axis(side=2,labels=mod[3],col.axis=cor[3],at=94,
      tick=F,las=1)
lines (1:38,f[,3]+93,col=cor[3])
abline (a=94,b=0)
Axis(side=2,labels=mod[4],col.axis=cor[4],at=136,
      tick=F,las=1)
lines (1:38,f[,2]+135,col=cor[4])
abline (a=136,b=0)
Axis(side=1,line=T,pos=-1,at=seq(1,38,2),
      lty=8,cex.axis=0.5)
# Grafico das VPPs do modelo estatico, por rodada
plot (c(7:38),f[7:38,2],type='l',xlim=c(7,38),
      ylim=c(0,42),col=cor[4],ylab='VPP',
      xlab='Rodada',main='VPP da UFRJ por Rodada')
abline (a=1,b=0)

file=file.choose()
a=read.csv(file=file,header=T)# dados brutos
b=grep(";",a[[1]],value=TRUE)# vetor com os dados sem
# as datas dos jogos
c=strsplit(b,";")

```

```

d=unlist(c)
e=matrix(d, length(b), length(c[[1]]), byrow=T)
f=paste(e[,1], e[,2], sep='x')
e=e[, c(-1, -2)]
g=as.numeric(e)
b=matrix(g, length(b), length(c[[1]])-2, byrow=F)
c=file.choose() #modelos dinamicos
colnames(b)=c
rownames(b)=f
c=b[, 1:11] #matriz com as VPs por jogo
d=(b[,1]-b[,2]>0) #Vitorias do mandante
e=(b[,1]-b[,2]==0) #Empates
f=(b[,1]-b[,2]<0) #Vitorias do visitante
for(g in 1:11){
  if(g<10){
    c[,g]=d*b[, (g*3+1)]+e*b[, (g*3+2)]+f*b[, (g*3+3)]
  }
  if(g==10){
    c[,g]=1/3
  }
  if(g==11){
    c[,g]=d*b[, (g*3-2)]+e*b[, (g*3-1)]+f*b[, (g*3)]
  }
}
colnames(c)=file.choose() #modelos dinamicos
d=c*3 #matriz com as VPs padronizadas por jogo
e=apply(c[1:10,], FUN=prod, 2) #matriz com as VPs por rodada
for(r in 2:7){
  f=apply(c[(10*r-9):(10*r)], FUN=prod, 2)
  e=rbind(e, f)
}
f=apply(c[71:79,], FUN=prod, 2)
e=rbind(e, f)
rownames(e)=31:38

```

```

f=e#matriz com as VPs padronizadas por rodada
for(g in 1:11){
  f[,g]=e[,g]/e[,10]
}
mod=file.choose()#modelos dinamicos
cor=c(rainbow(9),'gold')
#Grafico das VPPs de todos os modelos, por jogo
plot(0,0,main='VPPs por Jogo',xlab='',ylab='',
      axes=F,ylim=c(0,30),xlim=c(301,379),type='l')
for(g in 1:9){
  Axis(side=2,labels=mod[g],col.axis=cor[g],
        at=1+3*(g-1),tick=F,las=1,cex.axis=0.5)
  lines(301:379,d[,10-g]+3*(g-1),col=cor[g])
  abline(a=1+3*(g-1),b=0)
}
Axis(side=2,labels=mod[10],col.axis=cor[10],
      at=28,tick=F,las=1)
lines(301:379,d[,11]+27,col=cor[10])
abline(a=28,b=0)
Axis(side=1,line=T,pos=-1,at=seq(301,379,10),
      lty=8,cex.axis=0.5)
#Grafico das VPPs de todos os modelos, por rodada
plot(0,0,main='VPPs por Rodada',xlab='',ylab='',
      axes=F,ylim=c(0,240),xlim=c(31,38),type='l')
for(g in 1:9){
  Axis(side=2,labels=mod[g],col.axis=cor[g],
        at=1+23*(g-1)+10,tick=F,las=1,cex.axis=0.5)
  lines(31:38,f[,10-g]+23*(g-1)+10,col=cor[g])
  abline(a=1+23*(g-1)+10,b=0)
}
Axis(side=2,labels=mod[10],col.axis=cor[10],
      at=218,tick=F,las=1)
lines(31:38,f[,11]+217,col=cor[10])
abline(a=218,b=0)

```



```

Axis(side=1, line=T, pos=-1, at=seq(31, 38, 1),
      lty=8, cex.axis=0.5)

times=file.choose()
siglas=file.choose()
n=length(times)
fatores=c('Ataque', 'Campo', 'Defesa')
r1=c(6, 20, 27, 38)
G=list()
contador=0
for(r in r1){
  n1=1000#tamanho das amostras das posteriores
  n2=3#numero de cadeias
  f=list()#lista com as matrizes das cadeias
  g=list()
  for(h in 1:n2){
    file=file.choose()
    g[[h]]=read.table(file)[[2]]
  }
  h=t(matrix(unlist(g), n2, length(g[[1]]), byrow=TRUE))
  i=list()
  j=(n-1)*n1
  for(k in 1:3){
    i[[k]]=h[((j*(k-1)+1):(j*k)), ]
  }
  i[[4]]=h[((3*j+1):length(h[,1])), ]
  f=i
  g=list()#lista com as amostras para cada um dos fatores
  for(h in 1:3){
    g[[h]]=list()#lista com as amostras de cada time
    #para um fator
    g[[h]][[1]]=rep(0, (n1*n2))#a amostra do primeiro
    #time e composta de zeros
  }
}

```

```

    for(i in 2:n){
        g[[h]][[i]]=as.vector(f[[h]][((i-2)*n1+1):
                                ((i-1)*n1),])
    }
}
g[[4]]=as.vector(f[[4]])
h=g
for(j in 1:3){
    k=h[[j]][[1]]
    for(l in 1:n){
        k=k+h[[j]][[1]]
    }
    k=k/n
    for(l in 1:n){
        h[[j]][[1]]=h[[j]][[1]]-k
    }
}
g=h
contador=contador+1
G[[contador]]=g
}

par(mfrow=c(2,2))
i=3
j=1
for(h in 1:4){
    hist(G[[h]][[i]][[j]], xlim=c(-3.5,3.5),
        xlab=paste('Fator_', fatores[i], '_-',
                    times[j], sep=''),
        main=paste('Histograma_pos_a_',
                    as.numeric(r1[h]),
                    'a_rodada', sep=''),
        prob=TRUE, col='skyblue')
    curve(dnorm(x, mean=mean(G[[h]][[i]][[j]]),

```

```

        sd=sd(G[[h]][[i]][[j]])),
        add=TRUE)
}
for(h in 1:4){
    hist(G[[h]][[4]], xlim=c(-5,5),
        xlab='Intercepto',
        main=paste('Histograma_pos_a_',
                    as.numeric(r1[h]),
                    'a_rodada', sep=' '),
        prob=TRUE, col='skyblue')
    curve(dnorm(x, mean=mean(G[[h]][[4]]),
              sd=sd(G[[h]][[4]])), add=TRUE)
}

```

GRAFICOS MODELO DINAMICO#####

```

times=file.choose()
siglas=file.choose()
n=length(times)
fatores=c("Ataque", 'Campo', 'Defesa')
atu=c(5,8,10,13,19)
precn=1
r=37
r1=atu[precn]#numero de rodadas para a atualizacao dos parame
r2=ceiling((r/r1))
#-----LEITURA DO CODA-----

n1=1000#tamanho das amostras das posteriors
n2=3#numero de cadeias

```

```

f=list()#lista com as matrizes das cadeias
g=list()
for(h in 1:n2){
  file=file.choose()
  g[[h]]=read.table(file)[[2]]
}
h=t(matrix(unlist(g),n2,length(g[[1]]),byrow=TRUE))
i=list()
j=(n-1)*r2*n1
for(k in 1:3){
  i[[k]]=h[((j*(k-1)+1):(j*k)),]
}
i[[4]]=h[((3*j+1):(3*j+n1*r2)),]
i[[5]]=h[((3*j+n1*r2+1):(length(h[,1]))),]
k=list()
for(l in 1:r2){
  m=seq(l,(length(i[[1]][,1])/n1),by=r2)
  o=c()
  for(p in 1:length(m)){
    o=append(o,c(((m[p]-1)*n1+1):(m[p]*n1)))
  }
  k[[l]]=o
}
l=list()
for(m in 1:3){
  l[[m]]=list()
  for(o in 1:r2){
    l[[m]][[o]]=i[[m]][k[[o]],]
  }
}
l[[4]]=list()
for(m in 1:r2){
  l[[4]][[m]]=i[[4]][(((m-1)*n1+1):(m*n1)),]
}

```

```

l[[5]]=i[[5]][(1:n1),]
f=1
g=list()#lista com as amostras para cada um
#dos fatores
for(h in 1:r2){
  g[[h]]=list()#lista com as amostras para
#cada ponto de atualizacao
  for(i in 1:3){
    g[[h]][[i]]=list()#lista com as amostras
#de cada time para um fator
    g[[h]][[i]][[1]]=rep(0,(n1*n2))#a amostra
#do primeiro time e composta de zeros
    for(j in 2:n){
      g[[h]][[i]][[j]]=as.vector(f[[i]][[h]]
                                   [((j-2)*n1+1):
                                   ((j-1)*n1),])
    }
  }
  g[[h]][[4]]=as.vector(f[[4]][[h]])
}
g[[r2+1]]=as.vector(f[[5]])
h=g
for(i in 1:r2){
  for(j in 1:3){
    k=h[[i]][[j]][[1]]
    for(l in 1:n){
      k=k+h[[i]][[j]][[l]]
    }
    k=k/n
    for(l in 1:n){
      h[[i]][[j]][[l]]=h[[i]][[j]][[l]]-k
    }
  }
}
}

```

g=h

```
#-----ESTIMATIVAS DE BAYES-----
EB=list()
LI=list()
LS=list()
for(h in 1:r2){
  EB[[h]]=matrix(NA,n,3)#matriz com as
  #estimativas de Bayes para cada fator
  for(i in 1:3){
    j=lapply(g[[h]][[i]],FUN=mean)
    EB[[h]][,i]=c(do.call("cbind",j))
  }
  rownames(EB[[h]])=times
  colnames(EB[[h]])=fatores
  LI[[h]]=matrix(NA,n,3)#matriz com os
  #limites inferiores de 95% de credibilidade
  #para as estimativas de Bayes para cada fator
  LS[[h]]=matrix(NA,n,3)#matriz com os
  #limites superiores de 95% de credibilidade
  #para as estimativas de Bayes para cada fator
  for(i in 1:3){
    j=function(x){
      return(sort(x)[75])
    }
    k=lapply(g[[h]][[i]],FUN=j)
    LI[[h]][,i]=c(do.call("cbind",k))
    j=function(x){
      return(sort(x)[2925])
    }
    k=lapply(g[[h]][[i]],FUN=j)
    LS[[h]][,i]=c(do.call("cbind",k))
  }
  rownames(LI[[h]])=times
```

```

colnames (LI[[h]])=fatores
rownames (LS[[h]])=times
colnames (LS[[h]])=fatores
for (i in 1:3) {
    LI[[h]][,i]=LI[[h]][,i]-mean (EB[[h]][,i])
    LS[[h]][,i]=LS[[h]][,i]-mean (EB[[h]][,i])
    EB[[h]][,i]=EB[[h]][,i]-mean (EB[[h]][,i])
}
}
ata=EB[[1]][,1]
cam=EB[[1]][,2]
def=EB[[1]][,3]
liminfata=LI[[1]][,1]
liminfcam=LI[[1]][,2]
liminfdef=LI[[1]][,3]
limsupata=LS[[1]][,1]
limsupcam=LS[[1]][,2]
limsupdef=LS[[1]][,3]
for (a in 2:r2) {
    ata=cbind (ata, EB[[a]][,1])
    cam=cbind (cam, EB[[a]][,2])
    def=cbind (def, EB[[a]][,3])
    liminfata=cbind (liminfata, LI[[a]][,1])
    liminfcam=cbind (liminfcam, LI[[a]][,2])
    liminfdef=cbind (liminfdef, LI[[a]][,3])
    limsupata=cbind (limsupata, LS[[a]][,1])
    limsupcam=cbind (limsupcam, LS[[a]][,2])
    limsupdef=cbind (limsupdef, LS[[a]][,3])
}
col=file.choose ()
col2=file.choose ()
ata=cbind (ata, siglas, col, col2)
cam=cbind (cam, siglas, col, col2)
def=cbind (def, siglas, col, col2)

```

```

liminfata=cbind(liminfata, siglas, col, col2)
liminfcam=cbind(liminfcam, siglas, col, col2)
liminfdef=cbind(liminfdef, siglas, col, col2)
limsupata=cbind(limsupata, siglas, col, col2)
limsupcam=cbind(limsupcam, siglas, col, col2)
limsupdef=cbind(limsupdef, siglas, col, col2)
maxls=max(unlist(lapply(LS, FUN=max)))
minli=max(unlist(lapply(LI, FUN=min)))
#Grafico dos EMVs do ataque
par(mfrow=c(2,2))
for(a in 20:17){
  plot(as.numeric(ata[,1]), (n:1), main='',
    xlab='', ylab='', axes=F,
    ylim=c(minli, maxls), xlim=c(1, r2),
    type='n')
  Axis(side=2, labels=ata[(n+1-a), 9],
    col.axis=ata[(n+1-a), 10], at=0,
    tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1), lty=8)
  polygon(c(r2:1, 1:r2),
    c(as.numeric(rev(limsupata[n+1-a, 1:r2])),
    as.numeric(liminfata[n+1-a, 1:r2])),
    col = ata[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = ata[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)
  lines(c(1:r2), as.numeric(ata[n+1-a, 1:r2]),
    col = ata[n+1-a, r2+3])
}
mtext("EBs┐-┐Modelo┐Dinamico", side = 3,
  line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 16:13){
  plot(as.numeric(ata[,1]), (n:1), main='',
    xlab='', ylab='', axes=F,

```



```

        ylim=c(minli,maxls),xlim=c(1,r2),
        type='n')
Axis(side=2,labels=ata[(n+1-a),9],
      col.axis=ata[(n+1-a),10],at=0,
      tick=F,las=1)
Axis(side=1,line=T,at=seq(1,r2,1),lty=8)
polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupata[n+1-a,1:r2])),
          as.numeric(liminfata[n+1-a,1:r2])),
        col = ata[n+1-a,r2+2], border = NA)
abline(a=0,b=0,col = ata[n+1-a,r2+3])
Axis(side=4,at=0,las=1)
lines(c(1:r2),as.numeric(ata[n+1-a,1:r2]),
      col = ata[n+1-a,r2+3])
}
mtext("EBs□-□Modelo□Dinamico", side = 3,
      line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 12:9){
  plot(as.numeric(ata[,1]),(n:1),main='',
        xlab='',ylab='',axes=F,
        ylim=c(minli,maxls),xlim=c(1,r2),
        type='n')
Axis(side=2,labels=ata[(n+1-a),9],
      col.axis=ata[(n+1-a),10],at=0,
      tick=F,las=1)
Axis(side=1,line=T,at=seq(1,r2,1),
      lty=8)
polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupata[n+1-a,1:r2])),
          as.numeric(liminfata[n+1-a,1:r2])),
        col = ata[n+1-a,r2+2], border = NA)
abline(a=0,b=0,col = ata[n+1-a,r2+3])
Axis(side=4,at=0,las=1)

```

```

    lines(c(1:r2), as.numeric(ata[n+1-a, 1:r2]),
          col = ata[n+1-a, r2+3])
}
mtext("EBs□-□Modelo□Dinamico", side = 3,
      line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 8:5){
  plot(as.numeric(ata[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=ata[(n+1-a), 9],
        col.axis=ata[(n+1-a), 10], at=0,
        tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1),
        lty=8)
  polygon(c(r2:1, 1:r2),
          c(as.numeric(rev(limsupata[n+1-a, 1:r2])),
            as.numeric(liminfata[n+1-a, 1:r2])),
          col = ata[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = ata[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)
  lines(c(1:r2), as.numeric(ata[n+1-a, 1:r2]),
        col = ata[n+1-a, r2+3])
}
mtext("EBs□-□Modelo□Dinamico", side = 3,
      line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 4:1){
  plot(as.numeric(ata[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=ata[(n+1-a), 9],

```

```

        col.axis=ata[(n+1-a),10], at=0,
        tick=F, las=1)
Axis(side=1, line=T, at=seq(1, r2, 1),
     lty=8)
polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupata[n+1-a,1:r2])),
          as.numeric(liminfata[n+1-a,1:r2])),
        col = ata[n+1-a, r2+2], border = NA)
abline(a=0, b=0, col = ata[n+1-a, r2+3])
Axis(side=4, at=0, las=1)
lines(c(1:r2), as.numeric(ata[n+1-a,1:r2]),
      col = ata[n+1-a, r2+3])
}
mtext("EBs□-□Modelo□Dinamico", side = 3,
      line = -2, outer = TRUE)

#Grafico dos EMVs do campo
par(mfrow=c(2,2))
for(a in 20:17){
  plot(as.numeric(cam[,1]), (n:1), main='',
       xlab='', ylab='', axes=F,
       ylim=c(minli, maxls), xlim=c(1, r2),
       type='n')
  Axis(side=2, labels=cam[(n+1-a),9],
       col.axis=cam[(n+1-a),10], at=0,
       tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1),
       lty=8)
  polygon(c(r2:1, 1:r2),
          c(as.numeric(rev(limsupcam[n+1-a,1:r2])),
            as.numeric(liminfcam[n+1-a,1:r2])),
          col = cam[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = cam[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)

```

```

    lines(c(1:r2), as.numeric(cam[n+1-a, 1:r2]),
          col = cam[n+1-a, r2+3])
}
mtext("EBs┐-┐Modelo┐Dinamico", side = 3,
      line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 16:13){
  plot(as.numeric(cam[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=cam[(n+1-a), 9],
        col.axis=cam[(n+1-a), 10], at=0,
        tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1),
        lty=8)
  polygon(c(r2:1, 1:r2),
          c(as.numeric(rev(limsupcam[n+1-a, 1:r2])),
            as.numeric(liminfcam[n+1-a, 1:r2])),
          col = cam[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = cam[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)
  lines(c(1:r2), as.numeric(cam[n+1-a, 1:r2]),
        col = cam[n+1-a, r2+3])
}
mtext("EBs┐-┐Modelo┐Dinamico", side = 3,
      line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 12:9){
  plot(as.numeric(cam[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=cam[(n+1-a), 9],

```

```

    col.axis=cam[(n+1-a),10], at=0,
    tick=F, las=1)
Axis(side=1, line=T, at=seq(1, r2, 1),
    lty=8)
polygon(c(r2:1, 1:r2),
    c(as.numeric(rev(limsupcam[n+1-a,1:r2])),
      as.numeric(liminfcam[n+1-a,1:r2])),
    col = cam[n+1-a, r2+2], border = NA)
abline(a=0, b=0, col = cam[n+1-a, r2+3])
Axis(side=4, at=0, las=1)
lines(c(1:r2), as.numeric(cam[n+1-a,1:r2]),
    col = cam[n+1-a, r2+3])
}
mtext("EBs□-□Modelo□Dinamico", side = 3,
    line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 8:5){
    plot(as.numeric(cam[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
    Axis(side=2, labels=cam[(n+1-a),9],
        col.axis=cam[(n+1-a),10], at=0,
        tick=F, las=1)
    Axis(side=1, line=T, at=seq(1, r2, 1),
        lty=8)
    polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupcam[n+1-a,1:r2])),
          as.numeric(liminfcam[n+1-a,1:r2])),
        col = cam[n+1-a, r2+2], border = NA)
    abline(a=0, b=0, col = cam[n+1-a, r2+3])
    Axis(side=4, at=0, las=1)
    lines(c(1:r2), as.numeric(cam[n+1-a,1:r2]),
        col = cam[n+1-a, r2+3])
}

```

```

}
mtext ("EBs┐-┐Modelo┐Dinamico", side = 3,
        line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 4:1){
  plot(as.numeric(cam[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=cam[(n+1-a), 9],
        col.axis=cam[(n+1-a), 10], at=0,
        tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1),
        lty=8)
  polygon(c(r2:1, 1:r2),
           c(as.numeric(rev(limsupcam[n+1-a, 1:r2])),
             as.numeric(liminfcam[n+1-a, 1:r2])),
           col = cam[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = cam[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)
  lines(c(1:r2), as.numeric(cam[n+1-a, 1:r2]),
         col = cam[n+1-a, r2+3])
}
mtext ("EBs┐-┐Modelo┐Dinamico", side = 3,
        line = -2, outer = TRUE)

#Grafico dos EMVs da defesa
par(mfrow=c(2,2))
for(a in 20:17){
  plot(as.numeric(def[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=def[(n+1-a), 9],

```

```

        col.axis=def[(n+1-a),10], at=0,
        tick=F, las=1)
Axis(side=1, line=T, at=seq(1, r2, 1),
    lty=8)
polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupdef[n+1-a,1:r2])),
          as.numeric(liminfdef[n+1-a,1:r2])),
        col = def[n+1-a, r2+2], border = NA)
abline(a=0, b=0, col = def[n+1-a, r2+3])
Axis(side=4, at=0, las=1)
lines(c(1:r2), as.numeric(def[n+1-a,1:r2]),
      col = def[n+1-a, r2+3])
}
mtext("EBs□-□Modelo□Dinamico", side = 3,
      line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 16:13){
    plot(as.numeric(def[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
Axis(side=2, labels=def[(n+1-a),9],
    col.axis=def[(n+1-a),10], at=0,
    tick=F, las=1)
Axis(side=1, line=T, at=seq(1, r2, 1),
    lty=8)
polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupdef[n+1-a,1:r2])),
          as.numeric(liminfdef[n+1-a,1:r2])),
        col = def[n+1-a, r2+2], border = NA)
abline(a=0, b=0, col = def[n+1-a, r2+3])
Axis(side=4, at=0, las=1)
lines(c(1:r2), as.numeric(def[n+1-a,1:r2]),
      col = def[n+1-a, r2+3])

```

```

}
mtext ("EBs┐-┐Modelo┐Dinamico", side = 3,
        line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 12:9){
  plot(as.numeric(def[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=def[(n+1-a), 9],
        col.axis=def[(n+1-a), 10], at=0,
        tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1),
        lty=8)
  polygon(c(r2:1, 1:r2),
           c(as.numeric(rev(limsupdef[n+1-a, 1:r2])),
             as.numeric(liminfdef[n+1-a, 1:r2])),
           col = def[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = def[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)
  lines(c(1:r2), as.numeric(def[n+1-a, 1:r2]),
         col = def[n+1-a, r2+3])
}
mtext ("EBs┐-┐Modelo┐Dinamico", side = 3,
        line = -2, outer = TRUE)
par(mfrow=c(2,2))
for(a in 8:5){
  plot(as.numeric(def[,1]), (n:1), main='',
        xlab='', ylab='', axes=F,
        ylim=c(minli, maxls), xlim=c(1, r2),
        type='n')
  Axis(side=2, labels=def[(n+1-a), 9],
        col.axis=def[(n+1-a), 10], at=0,
        tick=F, las=1)

```



```

Axis(side=1, line=T, at=seq(1, r2, 1),
     lty=8)
polygon(c(r2:1, 1:r2),
        c(as.numeric(rev(limsupdef[n+1-a, 1:r2])),
          as.numeric(liminfdef[n+1-a, 1:r2])),
        col = def[n+1-a, r2+2], border = NA)
abline(a=0, b=0, col = def[n+1-a, r2+3])
Axis(side=4, at=0, las=1)
lines(c(1:r2), as.numeric(def[n+1-a, 1:r2]),
      col = def[n+1-a, r2+3])
}
mtext("EBs-ModeloDinamico", side = 3,
     line = -2, outer = TRUE)
par(mfrow=c(2, 2))
for(a in 4:1){
  plot(as.numeric(def[, 1]), (n:1), main='',
       xlab='', ylab='', axes=F,
       ylim=c(minli, maxls), xlim=c(1, r2),
       type='n')
  Axis(side=2, labels=def[(n+1-a), 9],
       col.axis=def[(n+1-a), 10], at=0,
       tick=F, las=1)
  Axis(side=1, line=T, at=seq(1, r2, 1),
       lty=8)
  polygon(c(r2:1, 1:r2),
          c(as.numeric(rev(limsupdef[n+1-a, 1:r2])),
            as.numeric(liminfdef[n+1-a, 1:r2])),
          col = def[n+1-a, r2+2], border = NA)
  abline(a=0, b=0, col = def[n+1-a, r2+3])
  Axis(side=4, at=0, las=1)
  lines(c(1:r2), as.numeric(def[n+1-a, 1:r2]),
        col = def[n+1-a, r2+3])
}
mtext("EBs-ModeloDinamico", side = 3,

```

```

line = -2, outer = TRUE)

file=file.choose()
a=read.csv(file=file,header=T)#dados brutos
b=grep(";",a[[1]],value=TRUE)#vetor com os
#dados sem as datas dos jogos
c=strsplit(b,";")
d=unlist(c)
e=matrix(d,length(b),length(c[[1]]),byrow=T)
f=paste(e[,1],e[,2],sep='x')
e=e[,c(-1,-2)]
g=as.numeric(e)
b=matrix(g,length(b),length(c[[1]])-2,byrow=F)
c=file.choose()
colnames(b)=c
rownames(b)=f
c=b[,1:6]#matriz com as VPs por jogo
d=(b[,1]-b[,2]>0)#Vitorias do mandante
e=(b[,1]-b[,2]==0)#Empates
f=(b[,1]-b[,2]<0)#Vitorias do visitante
for(g in 1:6){
  if(g<5){
    c[,g]=d*b[, (g*3+1)]+e*b[, (g*3+2)]+f*b[, (g*3+3)]
  }
  if(g==5){
    c[,g]=1/3
  }
  if(g==6){
    c[,g]=d*b[, (g*3-2)]+e*b[, (g*3-1)]+f*b[, (g*3)]
  }
}
colnames(c)=file.choose()

```

```

d=c*3#matriz com as VPs padronizadas por jogo
e=apply(c[1:10,],FUN=prod,2)#matriz com as VPs por rodada
for(r in 2:37){
  f=apply(c[(10*r-9):(10*r)],FUN=prod,2)
  e=rbind(e,f)
}
f=apply(c[371:379,],FUN=prod,2)
e=rbind(e,f)
rownames(e)=1:38
f=e#matriz com as VPs padronizadas por rodada
for(g in 1:6){
  f[,g]=e[,g]/e[,5]
}
mod=c('Simples','CdG','UFMG','Estatico','Dinamico')
cor=c('green','blue','red','gold','#DC143C')
#Grafico das VPPs de todos os modelos, por jogo
plot(0,0,main='VPPs por Jogo',xlab='',ylab='',
      axes=F,ylim=c(0,15),xlim=c(1,379),type='l')
Axis(side=2,labels=mod[1],col.axis=cor[1],at=1,
      tick=F,las=1)
lines(1:379,d[,6],col=cor[1])
abline(a=1,b=0)
Axis(side=2,labels=mod[2],col.axis=cor[2],at=4,
      tick=F,las=1)
lines(1:379,d[,1]+3,col=cor[2])
abline(a=4,b=0)
Axis(side=2,labels=mod[3],col.axis=cor[3],at=7,
      tick=F,las=1)
lines(1:379,d[,4]+6,col=cor[3])
abline(a=7,b=0)
Axis(side=2,labels=mod[4],col.axis=cor[4],at=10,
      tick=F,las=1)
lines(1:379,d[,2]+9,col=cor[4])
abline(a=10,b=0)

```

```

Axis(side=2, labels=mod[5], col.axis=cor[5], at=13,
      tick=F, las=1, cex.axis=0.8)
lines(1:379, d[,3]+12, col=cor[5])
abline(a=13, b=0)
Axis(side=1, line=T, pos=-1, at=seq(0, 380, 20),
      lty=8, cex.axis=0.5)
# Grafico das VPPs do modelo dinamico, por jogo
plot(c(221:379), d[221:379,3], type='l',
      xlim=c(221, 379), ylim=c(0, 3), col=cor[5],
      ylab='VPP', xlab='Jogo',
      main='VPP do Modelo dinamico por Jogo')
abline(a=1, b=0)
# Grafico das VPPs de todos os modelos, por rodada
plot(0, 0, main='VPPs por Rodada', xlab='',
      ylab='', axes=F, ylim=c(0, 210),
      xlim=c(1, 38), type='l')
Axis(side=2, labels=mod[1], col.axis=cor[1],
      at=4, tick=F, las=1)
lines(1:38, f[,6]+3, col=cor[1])
abline(a=4, b=0)
Axis(side=2, labels=mod[2], col.axis=cor[2],
      at=46, tick=F, las=1)
lines(1:38, f[,1]+45, col=cor[2])
abline(a=46, b=0)
Axis(side=2, labels=mod[3], col.axis=cor[3],
      at=88, tick=F, las=1)
lines(1:38, f[,4]+87, col=cor[3])
abline(a=88, b=0)
Axis(side=2, labels=mod[4], col.axis=cor[4],
      at=130, tick=F, las=1)
lines(1:38, f[,2]+129, col=cor[4])
abline(a=130, b=0)
Axis(side=2, labels=mod[5], col.axis=cor[5],
      at=172, tick=F, las=1, cex.axis=0.8)

```

```

lines (1:38, f[,3]+171, col=cor[5])
abline (a=172, b=0)
Axis (side=1, line=T, pos=-1, at=seq(1, 38, 2),
      lty=8, cex.axis=0.5)
#Grafico das VPPs do modelo estatico, por rodada
plot (c(23:38), f[23:38,4], type='l',
      xlim=c(23, 38), ylim=c(0, 42), col=cor[5],
      ylab='VPP', xlab='Rodada',
      main='VPP do Modelo dinamico por Rodada')
abline (a=1, b=0)

```

```

file=file.choose()
a=read.csv(file=file, header=T) #dados brutos
b=grep(";", a[[1]], value=TRUE) #vetor com os dados
#sem as datas dos jogos
c=strsplit(b, ";")
d=unlist(c)
e=matrix(d, length(b), length(c[[1]]), byrow=T)
f=paste(e[,1], e[,2], sep='x')
e=e[, c(-1, -2)]
g=as.numeric(e)
b=matrix(g, length(b), length(c[[1]])-2, byrow=F)
c=file.choose()
colnames(b)=c
rownames(b)=f
c=b[, 1:6] #matriz com as VPs por jogo
d=(b[,1]-b[,2]>0) #Vitorias do mandante
e=(b[,1]-b[,2]==0) #Empates
f=(b[,1]-b[,2]<0) #Vitorias do visitante
for(g in 1:6){
  if(g<=5){
    c[,g]=d*b[, (g*3+1)]+e*b[, (g*3+2)]+
      f*b[, (g*3+3)]
  }
}

```

```

    }
    if (g==6) {
        c[,g]=1/3
    }
}

colnames(c)=file.choose()
d=c*3#matriz com as VPs padronizadas por jogo
e=apply(c[1:10,],FUN=prod,2)#matriz com as VPs por rodada
for(r in 2:37){
    f=apply(c[(10*r-9):(10*r)],FUN=prod,2)
    e=rbind(e,f)
}
f=apply(c[371:379,],FUN=prod,2)
e=rbind(e,f)
rownames(e)=1:38
f=e#matriz com as VPs padronizadas por rodada
for(g in 1:5){
    f[,g]=e[,g]/e[,6]
}
mod=file.choose()
cor=c(1:5)
#Grafico das VPPs de todos os modelos, por rodada
plot(0,0,main='VPPs por Rodada',xlab='',
     ylab='',axes=F,ylim=c(0,115),
     xlim=c(31,38),type='l')
for(i in 5:1){
    Axis(side=2,labels=mod[i],col.axis=i,
         at=(4+23*(i-1)),tick=F,las=1,
         cex.axis=0.7)
    lines(31:38,f[31:38,i]+23*(i-1)+3,
          col=i)
    abline(a=(4+23*(i-1)),b=0)
}
Axis(side=1,line=T,pos=-1,

```

```

at=seq(31,38,1),lty=8,cex.axis=0.5)

##### GRaFICOS PRECISAO #####

Prec=list()
times=file.choose()
siglas=file.choose()
n=length(times)
atu=c(5,8,10,13,19)
precn=1
Prec[[precn]]=list()
for(r in 22:37){
  r1=atu[precn]#numero de rodadas para a
#atualizacao dos parametros
  r2=ceiling((r/r1))
  n1=1000#tamanho das amostras das posteriors
  n2=3#numero de cadeias
  f=list()#lista com as matrizes das cadeias
  g=list()
  for(h in 1:n2){
    file=file.choose()
    g[[h]]=read.table(file)[[2]]
  }
  h=t(matrix(unlist(g),n2,length(g[[1]]),byrow=TRUE))
  i=list()
  j=(n-1)*r2*n1
  i[[5]]=h[((3*j+n1*r2+1):(length(h[,1]))),]
  l=list()

```

```

l[[5]]=i[[5]][(1:n1),]
f=1
g=as.vector(f[[5]])
prec=g
Prec[[precn]][[r]]=prec
}
a=unlist(lapply(Prec[[precn]],FUN=mean))[22:37]
b=1/a
c=function(x){
  return(sort(x)[75])
}
d=unlist(lapply(Prec[[precn]],FUN=c)[22:37])
c=function(x){
  return(sort(x)[2925])
}
e=unlist(lapply(Prec[[precn]],FUN=c)[22:37])
a=b
b=1/e
c=e/d
plot(c(22:37),a,type='l',col='darkred',
      xlab='Rodada',ylab='Variância',
      main='EBs_da_variância_ao_longo_do_tempo')
plot(c(22:37),c,type='n',col='darkred',
      xlab='Rodada',ylab='Variância',
      ylim=c(0,max(c)),
      main='EBs_da_variância_ao_longo_do_tempo')
polygon(c(37:22, 22:37), c(rev(c),b),
        col = 'darkred', border = NA)
hist(1/Prec[[precn]][[37]],prob=TRUE,
      col='skyblue',xlab='Variância',
      main='Histograma_da_amostra_da_posteriori_da_variância')
bch=mean(1/Prec[[precn]][[37]])/var(1/Prec[[precn]][[37]])
ach=bch*mean(1/Prec[[precn]][[37]])
curve(dgamma(x,shape=ach,rate=bch),add=TRUE)

```


Referências

- Arruda, M. L. de, COMO TUDO FUNCIONA - Entenda os Cálculos do Chance de Gol, Disponível em: <<http://www.chancedegol.com.br/comofunciona.htm>>, Acesso em: 05 de Agosto de 2018.
- Boyce, J., How Our Club Soccer Projections Work, Disponível em: <<https://fivethirtyeight.com/features/our-2017-18-club-soccer-predictions/>>, Acesso em: 05 de Agosto de 2018.
- Bradley, R.A. e Terry, M.E. (1952), Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons, *Biometrika*, vol 39, No. 3-4, pp. 324-345.
- Cattelan, M., Varin, C. e Firth, D., Dynamic Bradley–Terry modelling of sports tournaments, *Journal of the Royal Statistical Society. Series C, Applied Statistics*, Vol. 62, No. 1, pp. 135-150.
- DeGroot, M. H. e Schervish M.J. (2002), *Probability and Statistics*, Addison-Wesley, Quarta Edição.
- Departamento de Matemática da UFMG, TABELA DA PRÓXIMA RODADA, Disponível em: <http://www.mat.ufmg.br/futebol/tabela-da-proxima-rodada_seriea/>, Acesso em: 05 de Agosto de 2018.
- Gamerman, D. e Lopes, H. F. (2006), *Markov Chain Monte Carlo stochastic simulation for bayesian inference*, Chapman & Hall/CRC, Segunda edição.

- Gamerman, D. e Souza Junior, O. G. (2004), Previsão de partidas de futebol usando modelos dinâmicos, Trabalho apresentado na SBPO XXXVI. São João del-Rei.
- Glickman, M.E. (1999), Parameter estimation in large dynamic paired comparison experiments, Journal of the Royal Statistical Society. Series C, Applied Statistics, Vol. 48, No. 3, pp. 377-394.
- Glickman, M. E. e Stern H. S. (1998), A state-space model for National Football League scores, Journal of the American Statistical Association, Volume 93, Issue 441, 1998.
- Knorr-Held, L. (2000), Dynamic rating of sports teams, Journal of the Royal Statistical Society. Series D, The Statistician, Vol. 49, No. 2, pp. 261-276.
- Rue, H. e Salvesen, Ø. (1997), Prediction and retrospective analysis of soccer matches in a league, Journal of the Royal Statistical Society. Series D, The Statistician, Vol. 49, No. 3, pp. 399-418.
- Vizzoni, A. R. (2015), Estatística no futebol europeu, Trabalho Apresentado na JIC-UFRJ de 2015, Rio de Janeiro.
- Vizzoni, A. R. (2016), Estatística no futebol europeu, Trabalho Apresentado na JIC-UFRJ de 2016, Rio de Janeiro.