UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA

CURSO DE GRADUAÇÃO EM ESTATÍSTICA

MATHEUS MOREIRA TEIXEIRA

INTRODUÇÃO À TEORIA DOS MODELOS POLINOMIAIS FRACIONÁRIOS

FORTALEZA

2022

MATHEUS MOREIRA TEIXEIRA

INTRODUÇÃO À TEORIA DOS MODELOS POLINOMIAIS FRACIONÁRIOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Estatística do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Estatística.

Orientador: Prof. Dr. Juvêncio Santos Nobre

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

T267i Teixeira, Matheus Moreira.
Introdução à teoria dos modelos polinomiais fracionários / Matheus Moreira Teixeira. – 2022.
97 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências, Curso de Estatística, Fortaleza, 2022.
Orientação: Prof. Dr. Juvêncio Santos Nobre.

1. Modelos polinomiais fracionários. 2. Regressão. 3. Software R. I. Título.

CDD 519.5

MATHEUS MOREIRA TEIXEIRA

INTRODUÇÃO À TEORIA DOS MODELOS POLINOMIAIS FRACIONÁRIOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Estatística do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Estatística.

Aprovada em: 20 de dezembro de 2022

BANCA EXAMINADORA

Prof. Dr. Juvêncio Santos Nobre (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Roberto Silva dos Santos
Universidade Federal do Ceará (UFC)

Prof. Dr. Luis Gustavo Bastos Pinho
Universidade Federal do Ceará (UFC)

À minha família, especialmente meus pais, pela sua participação ativa no meu progresso no curso de estatística e na vida.

AGRADECIMENTOS

Primeiramente, agradeço a Jeová Deus pelo privilégio de existir e ter uma relação com Ele. Ele me proporciona por meio do estudo da Bíblia conhecimento, sabedoria e discernimento para tomar boas decisões. Além disso, a adoração a Jeová e Sua bondade imerecida são o que me dá paz e estabilidade na vida. Sem estas coisas, eu certamente não teria chegado até aqui nem seria quem sou hoje.

Agradeço aos meus pais, Antônio Marcos Teixeira e Maria Auxiliadora Moreira Teixeira. A educação, disciplina, orientação, amor, apoio e conselhos deles me ajudaram a trilhar o caminho para chegar a este ponto e a resistir às adversidades que surgiram ao longo trajetória. Agradeço também ao meu finado avô, João Mendes Teixeira, por ter, enquanto pôde, ter me dado apoio para prosseguir com os estudos. Agradeço também ao meu irmão, Marcos Messias Moreira Teixeira, pela apoio dado nesta trajetória (me refiro à carona que ele me deu uma vez na volta da faculdade) e à minha cunhada, Bruna Costa Teixeira, pelo apoio moral.

Agradeço ao meu professor de física do ensino médio, Raphael Freitas. Foi ele quem me apresentou o curso de estatística e me inspirou a fazê-lo.

Agradeço também ao professor Juvêncio Santos Nobre por ter aceitado me orientar nesta monografia. O seu curso ministrado de modelos de regressão I certamente me inspirou interesse pela área. Também agradeço pela paciência do professor para lidar com este orientando particularmente teimoso e precipitado. Aos professores José Roberto Silva dos Santos e Luis Gustavo Bastos Pinho agradeço pela disponibilidade de participar na banca e pelas valiosas contribuições a este trabalho.

Agradeço à coordenação do curso de Estatística e a todos os professores do DEMA. Em especial, aos professores Maria Jacqueline Batista, Gualberto Agamez Segundo Montalvo e José Ailton Alencar Andrade. Os cursos que fiz com estes professores não apenas me ensinaram fatos, mas me fizeram desenvolver ainda mais interesse pelo estudo de áreas variadas da estatística (especialmente estatística bayesiana). Também agradeço ao professor Tibérius de Oliveira Bonates pela oportunidade de uma bolsa de monitoria na disciplina de fundamentos de programação. Esta bolsa aumentou muito minha desenvoltura na área.

Agradeço também a todos os meus colegas do curso de estatística. Em especial, a Marília de Melo Sombra e Eric Oliveira Rocha. Sem os conselhos e as dúvidas tiradas por estes dois, não teria sobrevivido à maioria das disciplinas que fiz neste curso.

Agradeço também ao pessoal do meu estágio no Supermercado Pinheiro. Sem a

bolsa generosa, eu não teria tido acesso à bibliografia necessária para fazer este trabalho.

Agradeço também a todas as outras pessoas que contribuíram para o meu desenvolvimento enquanto pessoa, estudante e prospectivo profissional.

Obrigado a todos vocês, pessoas maravilhosas! E muito sucesso a todos vocês!

“O único homem que está isento de erros é aquele não arrisca acertar.”

(Albert Einstein)

RESUMO

Frequentemente lidamos com situações em que desejamos modelar o comportamento de uma variável de interesse em função de outras. O conjunto de técnicas estatísticas, matemáticas e probabilísticas usadas em tais situações é denominado modelos de regressão. Por muitas décadas, os modelos lineares usuais têm constituído a classe mais frequentemente utilizada de modelos de regressão. Porém, em muitas situações, estes não são suficientes para se modelar o comportamento da variável de interesse. Um dos motivos disto é que, muitas vezes, os dados não atendem às suposições de distribuição e forma funcional dos modelos lineares. Por conta disto, diversas classes de modelos mais flexíveis têm sido propostas. Entre elas, podemos citar os modelos polinomiais, que permitem modelar a função de regressão através de polinômios das variáveis explicativas. Estes também podem servir como uma boa aproximação em casos em que esta função é não-linear. Por outro lado, podem não apresentar ajuste satisfatório a dados com determinadas características. Por exemplo, esta classe de modelos tende a não gerar bons ajustes a funções de regressão que têm assíntotas, além de, em alguns casos, ser particularmente sensível a pontos influentes. Alternativas aos modelos polinomiais incluem os modelos de regressão baseados em funções de suavização, como o *lowess*, que são não-paramétricos, e os FPMs (modelos polinomiais fracionários, do inglês *fractional polynomial models*). Os FPMs, em particular, têm tido desempenho satisfatório em modelar problemas em que os modelos polinomiais usuais apresentam limitações, dadas sua flexibilidade e parcimônia, de forma totalmente paramétrica. Estes têm sido aplicados junto outras classes mais complexas de modelos, como os modelos lineares generalizados (MLGs). Neste trabalho, é feita uma introdução à teoria dos FPMs, com apoio computacional do *software* R. Ademais, realiza-se uma aplicação dos FPMs a dados disponíveis na literatura, com o objetivo de ilustrar a aplicação desta classe de modelos.

Palavras-chave: Modelos polinomiais fracionários. Regressão. *Software* R.

ABSTRACT

We frequently deal with situations where we wish to model one variable's behaviour as a function of other variables of interest. The set of statistical, mathematical and probabilistic techniques used in such situations is called regression models. For several decades, the linear models have been the most commonly used class of regression models. However, in many situations, these are not sufficient to model our variable of interest's behaviour. One of the reasons of that is the fact that very often data do not satisfy the linear model assumptions. Therefore, several more flexible model classes have been proposed. Among them, we can cite polynomial models, which allow us to model the regression function through polynomials of the explanatory variables. These can also work as a good approximation in cases where this function is nonlinear. On the other hand, such models may not fit properly data with certain characteristics; for instance, this model class tends not to fit properly regression functions that have an asymptote and in some cases, is particularly vulnerable to influential points. Alternatives to polynomial models include regression models based on smoothing functions, such as lowess, which are non-parametric, and FPMs (fractional polynomial models). In particular, FPMs commonly outperform the usual polynomial models, mostly due to its flexibility, considering a fully parametric modeling. Such models have been used along with more complex model classes, such as GLMs (generalised linear models). In this work, we introduce the fundamentals of FPMs with computational support of the R software. We also present an application of FPMs in real data, aiming to illustrate the application of this model class.

Keywords: Fractional polynomial models. Regression. R Software.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE SÍMBOLOS

m	Grau de um polinômio fracionário.
φ_m	Polinômio fracionário de grau m.
$\boldsymbol{y}$	Vetor de variáveis resposta.
$\boldsymbol{x}$	Vetor de variáveis explicativas.
$\boldsymbol{\beta}$	Vetor de parâmetros.
$\boldsymbol{p}$	Vetor de potências de um polinômio fracionário.
$\boldsymbol{X}$	Matriz de especificação de um modelo de regressão.
$\boldsymbol{h}(\boldsymbol{x})$	Função de transformações polinomiais fracionárias de $\boldsymbol{x}$.
$\boldsymbol{e}$	Fonte de variação de um modelo de regressão.
n	Número de observações.
σ^2	Parâmetro de variância de uma distribuição normal.
S	Conjunto de potências de transformações polinomiais fracionárias.
ω_δ	Transformação para restringir uma variável ao intervalo (0, 1).
L	Função de verossimilhança.
ℓ	Função de log-verossimilhança.
$\hat{\boldsymbol{y}}$	Valores estimados da variável resposta.
$\hat{\boldsymbol{\beta}}$	Estimador de $\boldsymbol{\beta}$.
$\hat{\sigma}^2$	Estimador de σ^2.
k	Número de covariáveis do modelo.
D	Desvio do modelo.
α	Nível de significância nominal de um teste.
α_1	Nível de significância nominal do procedimento `BE`.
α_2	Nível de significância nominal do procedimento FSP.
gl	Graus de liberdade de um MFP.
$c_{\max}$	Número máximo de iterações do algoritmo MFP.
B	Número de réplicas em um *bootstrap*.

$\hat{f}_b$	Função estimada na b-ésima réplica *bootstrap*.
$\tilde{f}_b$	Função estimada na b-ésima réplica *bootstrap* padronizada.
f_{bag}	Função estimada por *bagging*.
f_{ref}	Função de referência.
R	Conjunto de réplicas *boostrap*.
T	Variação total entre a função de referência e as funções estimadas por *bagging*.
V	Variação entre as funções estimadas por *bagging*.
D^2	Quadrado da distância entre a curva de referência e a curva estimada por *bagging*.
V_{cond}	Variação condicional entre as funções estimadas por *bagging*.
R_{x_j}	Conjunto das réplicas em que a j-ésima covariável é selecionada.
q	Frequência de inclusão.

SUMÁRIO

1 CONSIDERAÇÕES INICIAIS

Frequentemente nos deparamos com situações em que temos interesse em explicar o resultado de certos fenômenos em termos de outros. Por exemplo, pode ser útil a indivíduos e a empresas descrever o consumo de combustível de um veículo em função do espaço percorrido pelo mesmo em uma viagem. Seguradoras modelam o risco de potenciais clientes sofrerem acidentes ou sinistros com base em uma série de variáveis, como estado civil, idade ou ocupação, por exemplo. Na área médica, encontram-se numerosos estudos que relacionam o desenvolvimento de certas enfermidades ao estilo de vida e hábitos dos pacientes. Nas engenharias, procura-se estimar a duração ou resistência de materiais e estruturas com base em sua composição e outros fatores.

Muitas das situações como as listadas acima têm um impacto relevante ao bem estar de indivíduos e da sociedade em geral. Estimar os gastos futuros com combustível pode ser vital à saúde financeira de uma empresa de logística. Avaliar o risco de um prospectivo cliente sofrer algum sinistro é essencial para que uma seguradora tome decisões assertivas sobre aceitar ou não o cliente e quanto cobrar. Estudos médicos sobre hábitos saudáveis e estilo de vida podem ter um impacto profundo na qualidade e expectativa de vida de milhões de pessoas. A estimação da resistência de materiais de uma edificação é essencial para que se garanta a segurança dos usuários da mesma.

Dada a necessidade de se ter modelos eficazes para se explicar os diferentes fenômenos pertinentes à atividade humana, é evidente a importância de se ter métodos confiáveis para a criação e avaliação dos mesmos. De fato, ao longo dos últimos séculos, e especialmente em décadas recentes, foi desenvolvido um amplo arcabouço teórico estatístico e probabilístico para auxiliar em tal tarefa. Tais técnicas de modelagem de fenômenos são denominadas modelos de regressão. Neles, busca-se explicar o resultado de um ou mais fenômenos, chamados variáveis respostas, com base no valor de uma ou mais variáveis explicativas, também chamadas covariáveis.

Reconsidere o caso em que se quer modelar o consumo de combustível de um veículo em função da distância percorrida. Nesse problema, o consumo é o que temos interesse em explicar, sendo a variável resposta. A distância percorrida é o que será usado para se modelar o consumo de combustível, sendo a variável explicativa. Por hora, denotaremos o consumo de combustível, em litros, por y e a distância percorrida na viagem, em quilômetros, por x. Em um primeiro momento, uma ideia razoável seria supor que o consumo é aproximadamente

proporcional à distância. Por exemplo, se na estrada determinado veículo gasta 0,1 L de gasolina quando percorre 1 km, é natural se esperar que ao fazer 2 km, ele consuma 0,2 L de combustível. Tal relação pode ser expressa matematicamente através da expressão

$$y = \beta x$$

em que β é o consumo de combustível por quilômetro percorrido. Em nosso caso hipotético, por exemplo, β seria igual a 0,1 L/km. Assim, quando se percorrese 1 km ($x = 1$), nosso consumo y seria igual a 0,1 L. Ao percorrer 2 km, ele consumiria $y = 0,2$ litros e assim por diante.

A realidade, evidentemente, não é tão simples. Há incontáveis fatores além da distância percorrida que influenciam o consumo de combustível de um veículo. Todavia, muitas vezes não se tem acesso direto a uma mensuração razoável de tais informações. Com efeito, considerando fixa a distância percorrida, sempre haverá uma pequena variação no consumo de combustível do automóvel. Por exemplo, se em determinada viagem ontem um veículo gastou 1 L de combustível, na mesma viajem hoje é possível que ele gaste 0,9 L ou 1,1 L, a depender de outros fatores.

Tal variação não ocorre apenas no caso do consumo de combustível, e deve ser levada em consideração nos modelos de regressão. A porção da variável resposta que não pode ser explicada por suas covariáveis é chamada de erro ou fonte de variação e é denotada por e. Tomando isso em consideração, uma forma mais completa de se abordar o problema do consumo de combustível seria expressá-lo através de

$$y = \beta x + e.$$

Naturalmente, espera-se que o valor do erro seja relativamente baixo. É possível, ainda adicionarmos à equação outro parâmetro, que chamaremos de β_0, caso haja evidências de que o automóvel gaste combustível mesmo sem percorrer nenhuma distância. Tal gasto poderia se dar no ato de se ligar o veículo ou mantê-lo com o motor ligado, mas parado. Nesse caso, renomeamos nosso β para β_1 e temos a seguinte expressão:

$$y = \beta_0 + \beta_1 x + e.$$

Suponha agora que foram realizadas recentemente 10 viagens em diferentes automóveis e que foram anotados a distância percorrida e o consumo em todas elas. Então temos 10 pares de observações (y_i, x_i), com i variando entre 1 e 10. Para a primeira viagem, em que $i = 1$, por exemplo, temos o par de consumo de combustível e distância percorrida (y_1, x_1). Para

a segunda viagem, temos (y_2, x_2) e assim por diante, até a décima viagem, em que temos o par (y_{10}, x_{10}). Podemos expressar funcionalmente o gasto de combustível na i-ésima viagem por

$$y_i = \beta_0 + \beta_1 x_i + e_i, \qquad i = 1, \ldots, 10, \tag{1.1}$$

em que e_i é o valor do erro ou fonte de variação na i-ésima viagem.

Para que sejam aplicadas técnicas inferenciais para se estimar os valores de β_0 e β_1 em (1.1), é necessário que façamos algumas suposições adicionais. A primeira delas, naturalmente, é que a função de regressão é linear. Outra suposição é que as observações são independentes, isto é, o consumo de combustível em uma viagem não influencia no consumo de outra. Pode-se mostrar que isso é equivalente a dizer que os erros são independentes. Também assumimos que os erros seguem todos uma distribuição probabilística normal com média zero e variância desconhecida, que chamaremos de σ^2.

Considere, finalmente, que temos um número n qualquer de viagens para as quais registramos o consumo e a distância percorrida do veículo. Nosso modelo seria então

$$y_i = \beta_0 + \beta_1 x_i + e_i, \qquad i = 1, \ldots, n. \tag{1.2}$$

Somado às suposições listadas acima, temos em (1.2) o caso mais simples de um modelo de regressão, chamado modelo de regressão linear simples (MRLS). Nele, procuramos expressar uma variável resposta em função de apenas uma variável explicativa através de uma função de regressão linear, assumindo fontes de variação independentes, normais e identicamente distribuídas. Informações adicionais sobre aspectos teóricos do MRLS podem ser encontradas em Davidson e MacKinnon (2004), Kutner *et al.* (2005), Hoffmann (2016) e Montgomery *et al.* (2021), por exemplo.

Quando se procura um modelo de regressão para explicar determinado fenômeno, tem-se interesse em dois aspectos: a qualidade do ajuste e a parcimônia. Naturalmente, se quer modelar a variável resposta tão bem quanto possível. Por outro lado, também é importante que o modelo seja simples, de fácil interpretação. Não é razoável, por exemplo, incluírem-se em um modelo variáveis que não influem de forma significativa na variável resposta, o que trará complicações adicionais e mais parâmetros para se interpretar e, consequentemente, para se estimar. Logo, pode-se dizer que um bom modelo de regressão é um que se ajusta bem aos dados, mas que tem o mínimo de parâmetros necessário a um bom ajuste. Em outras palavras, um bom modelo explica bem os dados e é parcimonioso.

Há muitas extensões possíveis ao MRLS. Por exemplo, pode-se estar interessado em descrever a variável resposta em função mais de uma covariável. Tal problema pode ser resolvido através de um modelo de regressão linear múltiplo (MRLM). Frequentemente, no entanto, as suposições do modelo de regressão linear não são satisfeitas. Por exemplo, é possível que a variância do erro, σ^2, não seja igual para todas as observações, ou mesmo que os erros não sigam distribuição normal. Para casos como os descritos, pode-se fazer adaptações no processo de estimação dos parâmetros ou nas variáveis envolvidas no estudo. Davidson e MacKinnon (2004), Kutner *et al.* (2005), Hoffmann (2016) e Montgomery *et al.* (2021) cobrem todas as extensões listadas neste parágrafo.

Em muitos casos, as suposições do modelo apresentado em (1.2) não se adequam bem aos dados. Por exemplo, é comum que a variável resposta não seja normalmente distribuída. Em outros casos, a função de regressão pode não ser linear em função das variáveis explicativas ou dos parâmetros. Por décadas, devido a limitações computacionais, modelos lineares normais foram usados para se explicar variáveis em casos como tais. Para que isso fosse possível, em muitos casos pesquisadores se utilizavam de transformações excessivamente engenhosas nas variáveis para se atingir a normalidade e a linearidade, mas estas nem sempre proporcionavam resultados desejáveis. Para o caso do comportamento não-linear da função de regressão em função das variáveis explicativas, utilizavam-se polinômios das últimas. Estes, no entanto, frequentemente resultavam em ajuste ruim a tipos específicos de dados ou a valores extremos dos mesmos.

Com os aprimoramentos nas ciências estatística e computacional nas décadas finais do Século XX, puderam ser propostos modelos de regressão cada vez mais flexíveis. Por exemplo, na década de 1970 foram propostos os modelos lineares generalizados, que ampliam o leque de opções para a distribuição da variável resposta. Pela primeira vez, foi possível modelar diretamente variáveis não-normais, ampliando os horizontes da regressão. Em décadas seguintes, outros modelos e métodos, como o uso de *splines* e polinômios fracionários, ampliaram ainda mais a capacidade dos modelos estatísticos. Os polinômios fracionários, em particular, têm ganhado espaço em anos recentes devido à sua capacidade de gerar um bom ajuste a dados que não são facilmente modelados através de polinômios convencionais. Estes, quando juntados aos modelos lineares generalizados, se constituem como uma ferramenta poderosa na análise de regressão.

O objetivo deste trabalho é apresentar uma introdução didática à teoria e à prática

dos modelos polinomiais fracionários. Para isso, será explanado brevemente o conceito de polinômios fracionários, e em seguida a classe de modelos propriamente dita. A parte prática será realizada através de uma exposição do uso do *software* R, de R Core Team (2022), para se ajustar modelos polinomiais fracionários. Também será apresentada uma aplicação da classe de modelos a um conjunto de dados real.

Para cumprir o objetivo descrito acima, este trabalho está organizado em quatro capítulos, além da introdução. No Capítulo 2, serão introduzidos os modelos polinomiais fracionários e algumas técnicas de seleção. Os aspectos computacionais da classe de modelos serão discutidos no Capítulo 3. No Capítulo 4, será realizada uma análise de dados reais a partir do paradigma dos modelos polinomiais fracionários. Por fim, no Capítulo 5, serão feitos uma síntese dos resultados obtidos, comentários a respeito da classe de modelos e sugestões para pesquisas futuras.

2 MODELOS POLINOMIAIS FRACIONÁRIOS

Boa parte dos modelos de regressão utilizados consideram a função de regressão linear nos parâmetros e também nas variáveis explicativas. Todavia, na prática, tal linearidade nem sempre é satisfeita. A função de regressão pode, em algumas situações, ter formato curvilíneo em função das variáveis explicativas. Em casos assim, o uso de regressões contendo apenas as covariáveis em suas formas originais pode resultar em ajuste ruim ou insatisfatório.

Por muitos anos, contornou-se esse problema por se considerar modelos polinomiais, não-lineares ou transformações de variáveis. Em muitos casos, a abordagem polinomial tem desempenho satisfatório, permitindo ao modelo descrever um comportamento curvilíneo da função de regressão. Em outros, ela deixa a desejar. Por exemplo, às vezes o grau do polinômio necessário para se obter um bom ajuste é elevado, o que pode causar ajuste ruim para valores extremos das covariáveis, problemas relacionados ao número de parâmetros do modelo, à sua parcimônia e à sua interpretabilidade. Ademais, polinômios com expoentes naturais não costumam se ajustar bem a funções de regressão com assíntotas.

Uma alternativa aos modelos polinomiais usuais é o uso de *splines* cúbicos e funções de suavização. Por exemplo, o método de regressão robusta localmente ponderada *lowess*, proposto por Cleveland (1979), tem tido bastante uso dada sua flexibilidade em se ajustar aos dados de forma não-paramétrica, isto é, sem pressupor uma forma funcional aos mesmos. Em contrapartida, como argumentam Royston e Altman (1994) e Binder *et al.* (2013), tais modelos frequentemente pecam em fornecer equações simples para se predizer novas observações e para se entender o comportamento de variáveis, pondo em dúvida, em alguns casos, sua parcimônia. Um exemplo de uso de funções suavizadas no preditor linear é a classe dos modelos aditivos generalizados, propostos por Hastie e Tibshirani (1990).

Dadas as complicações que pode haver em alguns casos para os modelos polinomiais comuns, podem ser considerados modelos polinomiais fracionários (FPMs, do inglês *fractional polynomial models*), que envolvem polinômios com expoentes não inteiros. Estes últimos podem ser vistos como uma generalização dos primeiros. Nos FPMs, a ideia de se utilizar transformações das covariáveis é estendida para potências reais das mesmas. Nos modelos polinomiais tradicionais, as potências utilizadas se restringem aos números naturais. Dessa forma, os modelos polinomiais, bem como os lineares, são um subconjunto da mais geral classe dos modelos polinomiais fracionários. Neste trabalho, usaremos as siglas para polinômios fracionários e modelos polinomiais fracionários em inglês para manter a consistência com

outros trabalhos em língua portuguesa sobre o assunto, como o de Garcia (2019).

Os modelos polinomiais fracionários foram propostos por Royston e Altman (1994), sendo uma tentativa para superar as limitações dos modelos polinomiais tradicionais para alguns tipos de dados. Antes de sua formalização, porém, tais modelos eram esporadicamente usados, mas sem um referencial teórico unificado. Exemplos de uso de modelos de regressão com potências não naturais anteriores ao artigo de Royston e Altman podem ser encontrados em Box e Tidwell (1962) e Ounsted *et al.* (1982), por exemplo.

Desde sua proposição, os modelos polinomiais fracionários têm sido usados principalmente em estudos das áreas médica e biológica (GARCIA, 2019). Para citar alguns, temos o trabalho de Castelnuovo *et al.* (2006) que, através do modelo, sugeriram uma relação positiva entre o consumo baixo de álcool e a redução na mortalidade total, mas ao mesmo tempo dando suporte à relação positiva entre o alto consumo da droga e a mortalidade. Já Aregay *et al.* (2014) modelaram a persistência a longo prazo de anticorpos anti-HPV induzidos por vacinação através de uma abordagem bayesiana com polinômios fracionários. Outro estudo que se utiliza do modelo é o de Zhang *et al.* (2020), que sugeriram uma relação entre os níveis de vitamina D no cordão umbilical e pré-eclâmpsia de mulheres grávidas e a pressão arterial de sua prole na infância e adolescência. Ainda referente ao consumo de álcool, Biddinger *et al.* (2022) propuseram uma associação entre o risco de doença cardiovascular e o uso habitual da substância através de polinômios fracionários.

Todavia, os modelos polinomiais fracionários também têm sido aplicados em áreas além da medicina. Por exemplo, Royston e Altman (1995) foram capazes de modelar a economia de carros em função do tamanho de seu motor utilizando essa metodologia. Gilmour e Trinca (2005) propuseram o modelo para uso na análise de superfícies de respostas de experimentos. Obteve-se bom ajuste para os dados provenientes de um experimento que modelava o rendimento de plantações de nabos em função do espaçamento e densidade de sementes. Uma técnica semelhante à apresentada por Royston e Altman (1994) foi aplicada por Zhou *et al.* (2021) em problemas na área de análise de confiabilidade.

Numerosas extensões ao modelo podem ser encontradas na literatura. Sauerbrei *et al.* (2007) propuseram uma modificação do modelo de Cox, utilizando polinômios fracionários, para casos em que a suposição de riscos proporcionais não é atendida. Royston e Sauerbrei (2008) abordam o modelo a partir do paradigma dos modelos lineares generalizados (MLGs). Silke *et al.* (2009) modelaram o risco de mortalidade de pacientes ao entrar em um hospital através de

regressão logística com polinômios fracionários. Garcia (2019, p. 4) investigou estratégias para a implementação do uso de polinômios fracionários em modelos lineares mistos.

Os FPMs também têm sido utilizados no paradigma bayesiano. Sabané Bové e Held (2011) introduziram o conceito de modelos polinomiais fracionários bayesianos, além de um método de seleção baseado em Monte Carlo via cadeias de Markov (MCMC). Aregay *et al.* (2014), por exemplo, utilizaram a técnica para modelar a persistência de anticorpos anti-HPV induzidos por vacinação. Sabané Bové e Held (2011) também introduziram um pacote para a aplicação de FPMs bayesianos na linguagem R, chamado `bfp`.

Os polinômios fracionários (FPs, do inglês *fractional polynomials*) podem ser bastante úteis na modelagem de alguns tipos de dados. Como já visto, eles frequentemente apresentam ajuste superior aos polinômios convencionais, especialmente a valores discrepantes dos dados, isto é, a valores muito elevados ou reduzidos, e a dados extremais que possuem uma assíntota. Outrossim, têm tido ampla aplicação com classes diversas de modelos de regressão, tais como modelos lineares normais, modelos lineares generalizados e modelos de Cox para dados censurados, por exemplo.

Em termos de flexibilidade, os FPMs são, em alguns casos, comparáveis a modelos semiparamétricos ou não-paramétricos. Entre estes estão, por exemplo, o *lowess* e a classe dos modelos aditivos generalizados, proposta por Hastie e Tibshirani (1990). Entre as vantagens dos modelos polinomiais fracionários está o fato de os FPMs serem totalmente paramétricos e de interpretação mais direta. Ademais, após realizarem uma série de comparações entre *splines* e FPMs, Royston e Sauerbrei (2008) destacam que os FPMs tendem a concordar com modelos baseados em *splines*, mas apresentam resultados mais satisfatórios em contextos com múltiplas variáveis regressoras.

O objetivo deste capítulo é apresentar uma introdução sucinta dos FPMs. Na Seção 2.1, será explanada brevemente a base funcional do modelo, que são os polinômios fracionários. Na Seção 2.2, é descrito o caso univariado dos FPMs. Os métodos de estimação e testes de hipóteses são abordados na Seção 2.3. Na Seção 2.4, são descritas as técnicas de seleção para FPMs no caso univariado contínuo. Na Seção 2.5, apresentamos os FPMs multivariados e técnicas de seleção de variáveis para os mesmos. A investigação de interações entre as variáveis é abordada na Seção 2.6. Finalmente, na Seção 2.7, são discutidas as técnicas de diagnóstico para situações particulares dos FPMs.

2.1 Polinômios Fracionários

Os polinômios fracionários (FPs) constituem a base dos FPMs. Esses constituem um conjunto de funções que representam uma extensão dos polinômios convencionais, permitindo transformações de potências reais e logarítmicas naturais, e não apenas naturais, de suas variáveis. Em um primeiro momento, pode-se dizer que os polinômios fracionários apresentam a seguinte formal funcional:

$$\varphi_m(x;\ \boldsymbol{p}) = \beta_0 + \sum_{j=1}^{m} \beta_j x^{p_j}, \tag{2.1}$$

em que m é um inteiro positivo, $\boldsymbol{p} = (p_1, \ldots, p_m)^\top$ é um vetor de valores reais com $p_1 < \ldots < p_m$ e $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)^\top$ é um vetor de parâmetros reais.

A função em (2.1) permite uma ampla gama de formas para $\varphi_m(x)$. Isso se dá porque o conjunto de possíveis transformações potências para x é o próprio conjunto dos números reais. De tal maneira, termos contendo $x^{1/2}$, x^{-3} e até x^{π} são possíveis. Uma restrição que surge em decorrência de tal flexibilidade, porém, é que $x > 0$. Se x é negativo, a transformação x^p não está definida para todos os valores possíveis de $p \in \mathbb{R}$. Neste texto, será seguida a notação simplificada proposta por Royston e Altman (1994) para a transformação de Box e Tidwell (1962). Nela, considera-se que

$$x^{p_j} = \begin{cases} x^{p_j} & \text{se } p_j \neq 0, \\ \log x & \text{se } p_j = 0. \end{cases}$$

Neste trabalho nos concentraremos no uso de polinômios fracionários de graus 1 e 2. Royston e Sauerbrei (2008) comentam que, na maioria dos casos práticos, os graus 1 e 2 são suficientes para se obter um ajuste satisfatório. Exemplos de uso de polinômios fracionários de graus maiores que 2 podem ser encontrados em Royston e Altman (1997) e no Capítulo 5 de Royston e Sauerbrei (2008), por exemplo. Notacionalmente, os polinômios fracionários de graus 1 e 2 são denotados por FP1 e FP2, respectivamente. Considerando-se (2.1), os polinômios fracionários de grau 1 são da forma

$$\varphi_1(x;\ p_1) = \beta_0 + \beta_1 x^{p_1},$$

enquanto um FP2 possui a seguinte forma funcional:

$$\varphi_2(x;\ \boldsymbol{p}) = \beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_2}.$$

Uma extensão útil da definição em (2.1) pode ser feita para o caso em que há potências repetidas no FP. Considere $p_j = p_l$ para ao menos um par de índices (j, l), $1 \leq j, l \leq m$. Tomando-se o grau 2 como exemplo, pode-se mostrar que se $p_1 = p_2$,

$$\varphi_2(x;\ \boldsymbol{p}) = \beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_1} \log x.$$

De fato, considere o modelo

$$\begin{aligned}\varphi(x;\ \boldsymbol{p}) &= \beta_0^* + \beta_1^* x^{p_1} + \beta_2^* x^{p_1} \\ &= \beta_0^* + (\beta_1^* + \beta_2^*) x^{p_1}.\end{aligned}$$

Fazendo $\beta_0 = \beta_0^*$, $\beta_1 = \beta_1^* + \beta_2^*$ e $\beta_2 = (p_2 - p_1)\beta_2^*$ e tomando-se o limite quando $(p_1 - p_2) \to 0$, tem-se

$$\lim_{p_1 - p_2 \to 0} \varphi(x;\ \boldsymbol{p}) = \beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_1} \frac{x^{p_2 - p_1} - 1}{p_2 - p_1},$$

pois

$$\begin{aligned}\lim_{p_1 - p_2 \to 0} \beta_2 x^{p_1} \frac{x^{p_2 - p_1} - 1}{p_2 - p_1} &= \lim_{p_1 - p_2 \to 0} (p_1 - p_2)\beta_2^* x^{p_1} \frac{x^{p_2 - p_1} - 1}{p_2 - p_1} \\ &= \beta_2^* \lim_{p_1 - p_2 \to 0} (x^{p_2} - x^{p_1}) \\ &= 0.\end{aligned}$$

Por outro lado,

$$\lim_{p_1 - p_2 \to 0} \frac{x^{p_2 - p_1} - 1}{p_2 - p_1} = \log x.$$

Portanto,

$$\lim_{p_1 - p_2 \to 0} \varphi(x;\ \boldsymbol{p}) = \beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_1} \log x.$$

Tal relação se prova verdadeira para qualquer número arbitrário de repetições de uma determinada potência p_j. Para $p_1 = \ldots = p_m$, tem-se

$$\varphi_m(x;\ \boldsymbol{p}) = \beta_1 x^{p_1} + \sum_{j=2}^{m} \beta_j x^{p_1} (\log x)^{j-1}.$$

Uma definição mais geral para polinômios fracionários, que leva em conta a possibilidade de potências repetidas, é apresentada a seguir.

Definição 2.1 *(Royston e Sauerbrei (2008, p. 74)) Um polinômio fracionário de grau m em função de x tem a forma*

$$\varphi_m(x;\ \boldsymbol{p}) = \beta_0 + \sum_{j=1}^{m} \beta_j h_j(x), \tag{2.2}$$

em que

$$h_j(x) = \begin{cases} x^{p_j} & \text{se } p_j \neq p_{j-1}, \\ h_{j-1}(x)\log x & \text{se } p_j = p_{j-1}, \end{cases}$$

para $j = 1, \ldots, m$*, em que* m *é um inteiro positivo,* $\boldsymbol{p} = (p_1, \ldots, p_m)^\top$ *é um vetor de valores reais com* $p_1 \leq \ldots \leq p_m$ *e* $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_m)^\top$ *é um vetor de coeficientes reais.*

Pode-se depreender a forma de um FP a partir do valor de seu vetor de potências $\boldsymbol{p}$. Como observam Royston e Sauerbrei (2008), polinômios fracionários de grau 1 são sempre monótonos, e aqueles com $p_1 < 1$ têm assíntotas quando $x \to \infty$ (veja a Figura 1).

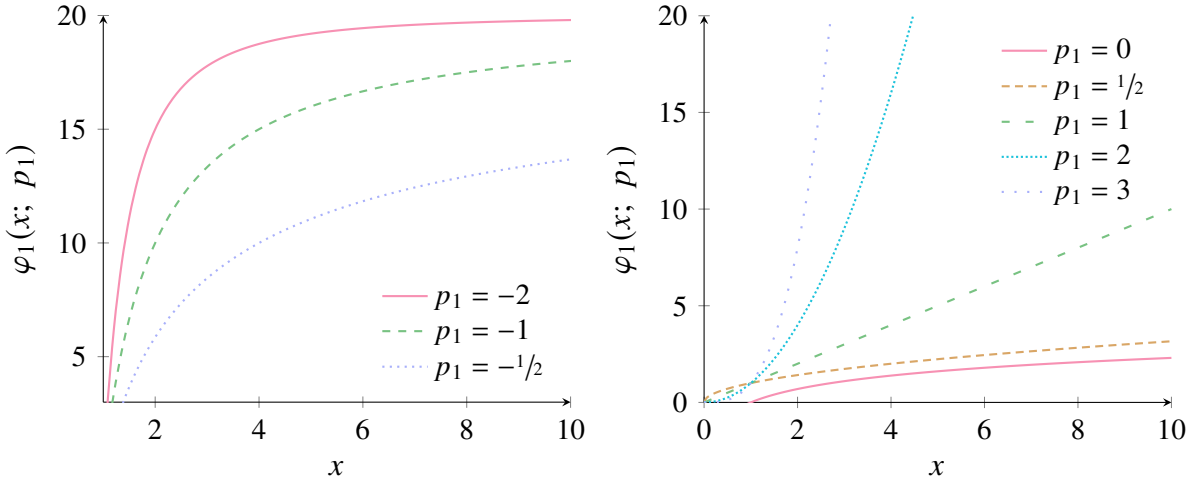
Os polinômios fracionários de grau 2 podem ser monótonos ou unimodais, tendo um ponto de máximo ou de mínimo. Quando ambos p_1 e p_2 são negativos, a curva apresenta uma assíntota. Royston e Sauerbrei (2008) observam que se $p_1 \neq p_2$, o FP2 será monótono se

$$\operatorname{sgn}(\beta_1\beta_2)\operatorname{sgn}(p_2) = \operatorname{sgn}(p_1),$$

considerando-se $\operatorname{sgn}(0) = 1$, e unimodal caso contrário. Se $p_1 = p_2$, a curva será unimodal (veja a Figura 2).

Figura 1 – Exemplos de funções FP1 para $x > 0$.

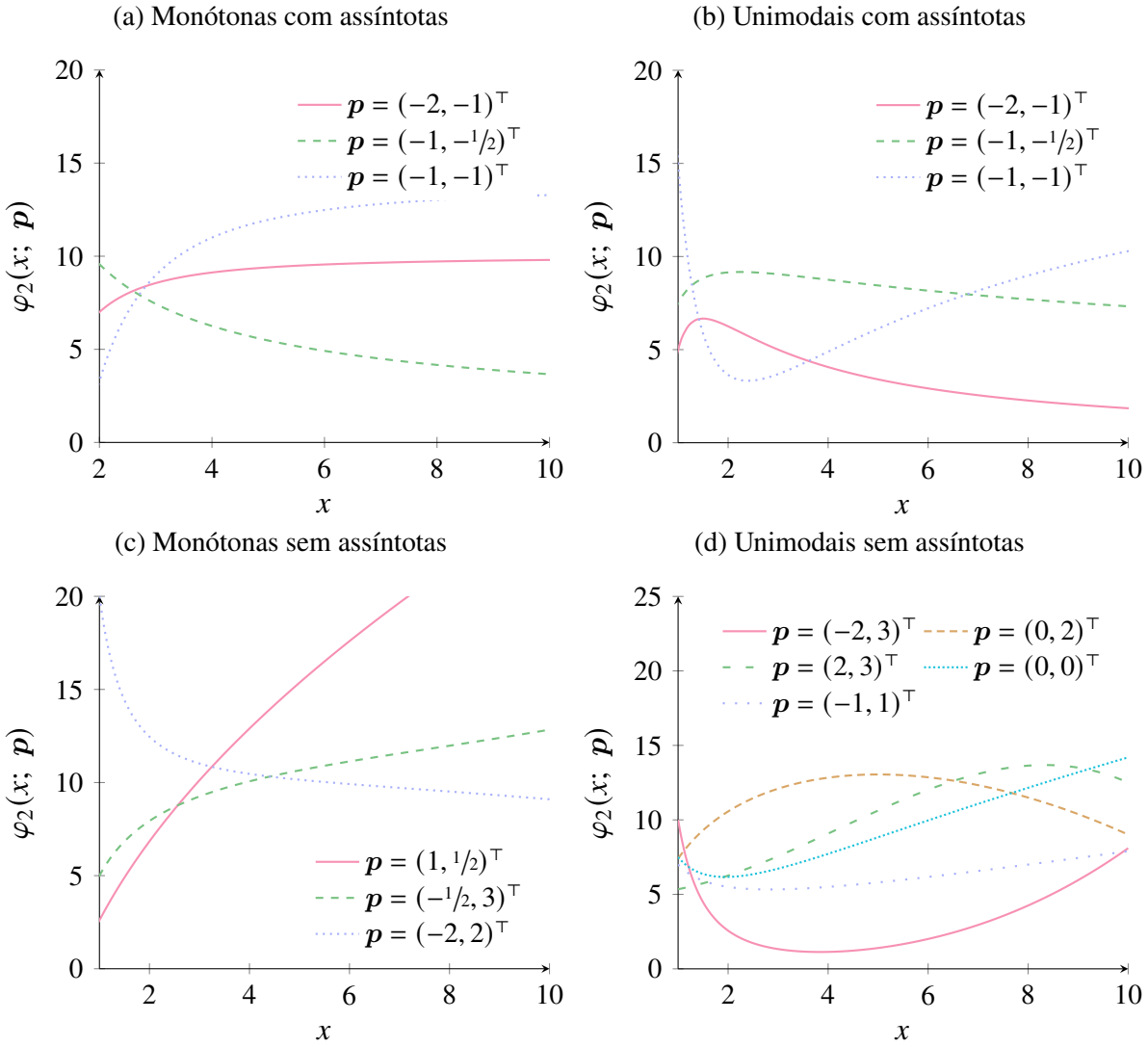
(a) Para $p_1 < 0$, com $\beta_1 < 0$ (b) Para $p_1 \geq 0$, com $\beta_1 > 0$



Fonte: adaptadas de Garcia (2019).

Note que os polinômios fracionários de graus 1 e 2 formam uma família extensa de curvas e formas. Nela, encontram-se funções com assíntotas, monótonas e unimodais. Uma característica das funções FP2 é a possibilidade de se ter variação acentuada em determinadas regiões e em outras se ter comportamento aproximadamente linear. FPs de grau 2 podem ainda se assemelhar a "polinômios" de graus 2 e 3 assimétricos, como se vê nas Figuras 2(b) e 2(d). Além disso, FPs de grau m podem possuir até $(m-1)$ modas, como mostram Royston e Sauerbrei (2008). Logo, FPs de grau 3 podem ter duas modas, FPs de grau 4 podem ter três modas, e assim por diante.

Figura 2 – Exemplos de funções FP2 para $x > 0$.



Fonte: elaboradas pelo autor.

A grande variedade de formas possíveis para polinômios fracionários e sua relativa simplicidade os tornam fortes candidatos para modelar fenômenos com diversas características.

Por exemplo, fenômenos que possuem assíntotas frequentemente não conseguem ser bem modelados por meio de modelos polinomiais convencionais, mas podem ser facilmente expressos através de FPs. Entre tais fenômenos, encontramos o crescimento de árvores e de seres humanos. Curvas assimétricas ou com mudanças bruscas de inclinação podem exigir polinômios naturais de graus elevados, que em geral são extremamente instáveis, ou *splines*, para serem ajustadas. Por outro lado, FPs de graus baixos podem modelar fenômenos como estes fazendo uso poucos parâmetros. Nas seções seguintes, discutiremos o uso de FPs em modelos de regressão.

2.2 Modelos Polinomiais Fracionários para Uma Única Variável Explicativa Quantitativa Contínua

Nesta seção, introduzimos o conceito de modelos polinomiais fracionários. Sua forma mais simples, bem como em outros modelos de regressão, é a com apenas uma variável regressora contínua e com fonte de variação com distribuição normal. Considere uma amostra com n indivíduos $(y_1, x_1), \ldots, (y_n, x_n)$. Um FPM de grau m univariado, em seu caso mais simples, tem forma funcional dada por

$$\begin{aligned} y_i &= \varphi_m(x_i;\, \boldsymbol{p}, \boldsymbol{\beta}) + e_i \\ &= \beta_0 + \sum_{j=1}^{m} \beta_j h_j(x_i) + e_i, \qquad i = 1, \ldots, n \end{aligned} \tag{2.3}$$

em que y_i é o valor da variável resposta para o i-ésimo indivíduo, x_i é o valor da variável regressora para o mesmo e os $e_1, \ldots, e_n$ são independentes e identicamente distribuídos com distribuição $\mathcal{N}(0, \sigma^2)$. Aqui, as suposições são similares às do modelo de regressão linear.

Como já comentado, neste trabalho nos concentraremos nos FPMs baseados em FPs de graus 1 e 2. Royston e Sauerbrei (2008) apontam que, na maioria dos casos práticos, tais FPs são suficientes para fornecer aproximações razoáveis para a função de regressão, com graus maiores que 2 raramente causando uma melhora significativa no ajuste de um FPM. De fato, depreende-se pelas Figuras 1 e 2 que FPs de graus 1 e 2 são capazes de modelar curvas de comportamentos diversos, como lineares, quadráticas e cúbicas, que são as mais comuns na modelagem polinomial convencional. Logo, nossos FPMs de interesse aqui seriam dados por

$$y_i = \beta_0 + \beta_1 x_i^{p_1} + e_i, \quad i = 1, \ldots, n$$

para $m = 1$,

$$y_i = \beta_0 + \beta_1 x_i^{p_1} + \beta_2 x_i^{p_2} + e_i, \quad i = 1, \ldots, n$$

para $m = 2$ com $p_1 \neq p_2$ e

$$y_i = \beta_0 + \beta_1 x_i^{p_1} + \beta_2 x_i^{p_1} \log x_i + e_i, \quad i = 1, \ldots, n$$

para $m = 2$ com $p_1 = p_2$. Não obstante, FPs de graus maiores que 2 são, por vezes, necessários para se ajustar bem dados de comportamento extremamente complexo.

Naturalmente, há uma quantidade infinita de potências que podem, em tese, ser usadas em FPMs. Royston e Sauerbrei (2008) sugerem, como regra de bolso, que se limitem as potências usadas em um FPM ao conjunto $S = \{-2, -1, -1/2, 0, 1/2, 1, 2, 3\}$, sendo este suficiente para aproximar a melhor escolha de potências dentro do intervalo $[-2, 3]$. O uso de um conjunto limitado de transformações de potência ao invés de todos os valores no conjunto dos reais facilita substancialmente a escolha de um modelo, especialmente para o caso multivariado. A estimação de potências depende de otimização não-linear, que não tem convergência garantida. Ademais, transformações com expoentes elevados tendem a ser muito sensíveis a valores extremos, prejudicando a robustez do modelo.

Convém lembrar que há casos em que a sugestão acima é insuficiente para um bom ajuste. Por vezes, pode ser necessário expandir o conjunto S para que o modelo se ajuste melhor ao comportamento dos dados. Por exemplo, Royston e Sauerbrei (2008) discorrem sobre a possibilidade da necessidade da adição da potência $p = 1/3$ ao conjunto S para se ajustar variáveis referentes a volume. Em casos em que se sabe que a função de regressão dos dados tem assíntota, pode ser prático restringir S a potências apenas negativas.

Há ainda certos tipos de fenômenos para os quais os FPMs dificilmente geram um bom ajuste. Royston e Sauerbrei (2003) asseveram que polinômios fracionários são ineficazes para se modelar dados com comportamento sigmoide, por exemplo. Para dados oriundos de séries temporais e de natureza periódica ou cíclica, o ajuste por polinômios fracionários, de forma geral, se mostra inadequado. Por outro lado, dado que os modelos polinomiais convencionais são um caso particular dos FPMs, estes últimos, na pior das hipóteses, gerarão um ajuste tão bom quanto os primeiros.

Os FPMs herdam dos polinômios fracionários a restrição de que x deve ser positivo. Não obstante, tal restrição frequentemente deixa de ser atendida em casos práticos. Uma solução inicial proposta por Royston e Altman (1994) é usar a transformação

$$x^* = x - x_{\min} + \gamma,$$

em que $x_{\min}$ é o valor mínimo observado para x e γ é o intervalo de arredondamento, isto é, o mínimo incremento possível entre os valores de x. A título de exemplo, se x foi medida com a precisão de duas casas decimais, γ seria 0,01. Se a variável foi medida com uma casa decimal, então $\gamma = 0,1$. Outra correção proposta por Sauerbrei *et al.* (2007) é uma combinação linear convexa entre 1 e o valor de x transformado, de forma a restringi-lo ao intervalo $(0, 1)$. Esta é dada por

$$\omega_\delta(x) = \delta + (1 - \delta)\frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

em que $x_{\max}$ é o valor máximo observado para x e $\delta = 0,2$.

Valores muito baixos ou muito altos das covariáveis podem afetar o ajuste em modelos com transformações negativas ou com graus elevados. Para mais transformações possíveis para a correção da origem de x, melhora da robustez e da qualidade de ajustamento em FPMs, veja Sauerbrei *et al.* (2007) e Royston e Sauerbrei (2008), por exemplo. Note que as potências de um FPM podem estar fortemente relacionadas à origem da variável. Portanto, é possível que diferentes transformações levem a modelos com diferentes potências.

Também é possível se utilizar FPMs com MLGs. Neste caso, basta usar o polinômio fracionário no preditor linear. Tal abordagem é realizada em Royston e Sauerbrei (2008), por exemplo.

2.3 Estimação e testes de hipóteses

Observe que uma função $\varphi_m(x;\, \boldsymbol{p},\, \boldsymbol{\beta})$ como definida em (2.2) pode ser vista como um preditor linear em função de $h(x)$ e $\boldsymbol{\beta}$. Logo, FPMs podem ser estimados através de métodos convencionais, como máxima verossimilhança, sem maiores complicações.

Considere uma matriz $\boldsymbol{X}$ com dimensão $n \times (m + 1)$ cuja primeira coluna é dada por um vetor $\mathbf{1}$ e cujas demais m colunas são dadas pelas n observações de cada transformação da variável regressora tal que

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & h_1(\boldsymbol{x}) & h_2(\boldsymbol{x}) & \dots & h_m(\boldsymbol{x}) \end{pmatrix},$$

em que $\boldsymbol{x}$ é o vetor de observações da variável explicativa e $\boldsymbol{h}$ é o vetor de transformações escolhido para esta. Sob as suposições apresentadas na Seção 2.2, a função de verossimilhança é

$$L(\boldsymbol{e}, \boldsymbol{\beta}, \sigma^2) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right\},$$

em que $\boldsymbol{y}$ é o vetor de observações e $\boldsymbol{\beta}$ é o vetor de parâmetros do modelo. Note que a função de verossimilhança para um FPM é igual à de um modelo linear usual. A log-verossimilhança é

$$\ell(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}). \tag{2.4}$$

Maximizando (2.4) com relação a $\boldsymbol{\beta}$ e σ^2, chegamos aos estimadores

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

e

$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^\top(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}).$$

Note que $\hat{\sigma}^2$ é um estimador viesado para a variância, mas para $n \to \infty$, o viés tende a zero. Pormenores do método de estimação e as propriedades dos estimadores de máxima verossimilhança são amplamente documentados na literatura. Para mais detalhes, veja Mood *et al.* (1974), Draper e Smith (1998), Davidson e MacKinnon (2004), Kutner *et al.* (2005), Hoffmann (2016) e Montgomery *et al.* (2021), por exemplo.

Dados os estimadores de máxima verossimilhança $\hat{\boldsymbol{\beta}}$ e $\hat{\sigma}^2$, pode-se testar as hipóteses

$$\begin{cases} \mathcal{H}_0 : \beta_j = 0, \\ \mathcal{H}_1 : \beta_j \neq 0, \end{cases} \qquad j = 0, 2, \ldots k$$

e a respectiva significância dos parâmetros, isto é, a importância de sua presença, através de testes t usuais.

2.4 Seleção de Modelos para o Caso Contínuo Univariado

Os testes de hipóteses para a escolha do modelo são baseados em testes F. Tais testes visam determinar se os parâmetros de determinado FPM (modelo completo) são significativos, isto é, se determinada potência da variável causa efeito significativo na variável resposta através da comparação com um modelo menor (encaixado).

Suponha que, em um FPM de grau m, as componentes do vetor $\boldsymbol{p}$ possam assumir qualquer valor real, sem estar restritas a S. Tal FPM seria um modelo não-linear com $2m + 1$ parâmetros. Royston e Altman (1994), Royston e Sauerbrei (2008) e Garcia (2019) mostram que, sob a hipótese nula, a estatística usada para testar $\boldsymbol{\beta} = \mathbf{0}$ (tanto no caso linear como no caso linear generalizado) conserva suas propriedades quando se limita $\boldsymbol{p}$ ao conjunto S.

Royston e Sauerbrei (2008) mostram que os modelos FP1 são encaixados nos modelos FP2. Estes explicam como isso ocorre fazendo uso da componente de desvio dos modelos. Considere o desvio de um modelo como sendo

$$D = -2\ell,$$

em que ℓ é a log-verossimilhança deste. Tem-se que para cada modelo FP1 com potência p_1^* há oito modelos FP2 com potências $\boldsymbol{p} = (p_1^*, p_2)^\top$. Naturalmente, o modelo FP1 seria um caso particular deste FP2 quando $\beta_2 = 0$. Desta forma, o FP1 com $p_1 = p_1^*$ está encaixado no modelo FP2 com $\boldsymbol{p} = (p_1^*, p_2)^\top$. Logo, o desvio D_2 de qualquer modelo FP2 com $\boldsymbol{p} = (p_1^*, p_2)$ é menor ou igual ao desvio D_1^* do melhor FP1 com $p_1 = p_1^*$. Portanto, tem-se que $D_2 \leq D_1^*$. Por outro lado, considerando D_2^* o desvio do melhor FP2, tem-se que $D_2^* \leq D_2$, estabelecendo-se a seguinte relação:

$$D_2^* \leq D_2 \leq D_1^*.$$

Portanto, a diferença de desvios $D_1^* - D_2^*$ é sempre não-negativa. Com isso, qualquer modelo FP1 está encaixado em um modelo FP2.

A partir de tal raciocínio, pode-se utilizar o procedimento FSP (*function selection procedure*), proposto por Ambler e Royston (2001) para se escolher um modelo adequado aos dados. O procedimento consiste em se partir de um modelo FP2 para os mesmos e, se possível, reduzi-lo para um mais simples por meio de testes F. Neles, são testadas as hipóteses

$$\begin{cases} \mathcal{H}_0 : \text{O modelo menor é tão eficaz em explicar os dados quanto o modelo maior,} \\ \mathcal{H}_1 : \text{O modelo menor é menos eficaz em explicar os dados que o modelo maior.} \end{cases}$$

Garcia (2019) resume os passos do FSP da seguinte maneira:

1. Escolha o nível de significância nominal α.
2. Teste o melhor modelo FP2 contra o modelo nulo β_0. Se o valor-p do teste é maior que α, não se rejeita a hipótese nula; logo, deve-se parar, pois os dados não podem ser explicados de forma satisfatória por um polinômio fracionário de grau 2 da variável x. Caso contrário, continue.
3. Teste o melhor modelo FP2 contra o modelo linear, com $p = 1$. Se o valor-p do teste é maior que α, tem-se que um modelo linear já é suficiente para se explicar os dados, então deve-se parar. Caso contrário, continue.
4. Teste o melhor modelo FP2 contra o melhor modelo FP1. Se o valor-p do teste é maior que α, selecione o FP1. Caso contrário, o FP2 é o modelo final.

Os valores mais comuns para α são 0,05 e 0,10. No segundo passo do FSP, é testado se a variável regressora é, de fato, capaz de explicar a variável resposta por meio de um FPM. Caso não o seja, não há porque prosseguir com a modelagem polinomial fracionária. O passo 3 testa se um modelo linear seria adequado aos dados. De tal forma, se dá primazia a modelos lineares, que são a escolha mais simples, obedecendo ao critério da parcimônia. Apenas se o modelo linear não se ajustar bem se considerarão FPs propriamente ditos no passo 4, sendo selecionado o FPM que melhor se adequar aos critérios de ajuste e de parcimônia.

Convém notar que há numerosas variações possíveis ao FSP apresentado. Por exemplo, a função do passo 3 pode ser modificada a depender do caso. Em modelos de Cox, não é incomum que se considere a transformação $p = 0$ ($\log x$) como o "caso mais simples". Caso se utilize um MLG ao invés de um modelo linear, utiliza-se o teste da razão de verossimilhanças generalizada ao invés do teste F. Garcia (2019) nota que este é um teste assintótico e conservador. Portanto, deve-se ter cautela ao aplicá-lo a amostras pequenas. Ademais, o procedimento pode ser estendido para se testar FPs de graus maiores que 2. Para mais detalhes, consulte Royston e Sauerbrei (2008).

Note que, no FSP e nos procedimentos de seleção que serão abordados adiante, pressupõe-se que o modelo "correto" está entre os que serão considerados. Destarte, é imprescindível o uso de técnicas de diagnóstico, que serão abordadas na Seção 2.7, para se avaliar a real eficácia do modelo selecionado.

2.5 Seleção de FPMs para Mais de Uma Variável Explicativa Quantitativa Contínua

Na maioria das aplicações práticas, há interesse em se modelar a variável resposta em função de mais de uma regressora. Isto gera modelos de regressão múltipla, trazendo mais complexidade para selecionar o modelo a ser utilizado. No caso específico de polinômios fracionários, tal processo envolve a escolha da transformação apropriada a cada covariável e a seleção daquelas que causam efeito significativo na variável resposta.

Sauerbrei e Royston (1999) propuseram um procedimento para realizar simultaneamente a seleção das covariáveis e de suas formas funcionais para modelos polinomiais fracionários, chamado MFP (*multiple fracional polynomials*). O mesmo consiste em uma combinação entre o método FSP descrito na Seção 2.4 para a verificação de não-linearidade e o procedimento `BE` (*backward elimination*) para a escolha de variáveis, que para determinada

variável testa

$$\begin{cases} \mathcal{H}_0 : \text{a variável não é significativa no modelo,} \\ \mathcal{H}_1 : \text{a variável é significativa.} \end{cases}$$

O algoritmo permite que se tenha certo controle sobre complexidade do modelo: o grau máximo do polinômio fracionário pode ser determinado por meio de sua componente FSP e o número de variáveis ajustado pela componente `BE`.

Há dois parâmetros que devem ser definidos ao se utilizar o procedimento MFP. O primeiro é α_1, que é o nível de significância nominal do método `BE` de seleção de variáveis. O segundo é o nível de significância nominal para o procedimento FSP, α_2. Royston e Sauerbrei (2008) propõem a notação MFP(α_1, α_2) para denotar um modelo selecionado pelo procedimento em questão utilizando os níveis de significância α_1 e α_2. Uma ocorrência comum é considerar $\alpha_1 = \alpha$. Nesse caso, pode-se utilizar a notação mais breve MPF(α). Note que α_1 está relacionado à propensão do modelo para manter variáveis, com $\alpha_1 = 1$ implicando na seleção de todas. Já α_2 está relacionado ao nível de complexidade do modelo final. Fazer $\alpha_2 = 1$ implica na escolha dos FPs mais complexos para todas as variáveis. É ainda possível que se determinem valores diferentes dos níveis de significância nominais para diferentes variáveis.

Outro parâmetro a ser definido antes de se realizar o procedimento MFP propriamente dito é o número máximo de graus de liberdade que serão usados (gl). Esse determina o grau máximo dos FPs que se deseja considerar para as variáveis regressoras. Por exemplo, se gl = 4, então o modelo mais complexo considerado é um FP2. Se gl = 2, o modelo mais complexo é um FP1; já se gl = 1, considera-se apenas modelos lineares. Royston e Sauerbrei (2008) sugerem que se utilize gl = 4, embora em alguns casos FPs de grau 2 possam ser insuficientes.

O ajuste de um MFP é realizado por intermédio de um procedimento iterativo, sendo descrito no Algoritmo 1. A cada ciclo, são avaliadas a significância e a forma funcional apropriadas para as variáveis. Deve-se determinar com antecedência um número máximo de iterações $c_{\max}$ a serem realizadas pelo procedimento. Royston e Sauerbrei (2008) recomendam que $c_{\max} = 5$, asseverando que o algoritmo geralmente alcança a convergência após 3 ou 4 ciclos.

O procedimento MFP inicialmente ajusta um modelo linear usual com todas as variáveis regressoras (linha 1 do Algoritmo 1). Em seguida, é realizada a ordenação das mesmas por seus níveis de significância de forma decrescente (linha 2). Dessa maneira, as regressoras menos significativas serão verificadas primeiro nos ciclos de seleção de variáveis e,

se for o caso, eliminadas.

Algoritmo 1: MFP.

Input: $0 < \alpha_1, \alpha_2 \leq 1$, gl, $c_{\max} \in \mathbb{N}$

Output: Um FPM (possivelmente) múltiplo

1 Ajuste um modelo linear usual com todas as variáveis;
2 Ordene as variáveis de acordo com os valores-p do ajuste obtidos no modelo linear de forma crescente;
3 $c \leftarrow 0$;
4 **while** $c < c_{\max}$ **do**
5 $j \leftarrow 1$;
6 **while** $j \leq k$ **do**
7 **if** x_j *é contínua* **then**
8 Aplique o passo 1 do procedimento FSP à variável x_j ao nível α_1;
9 **if** x_j *não é significativa* **then** Descarte x_j;
10 **else**
11 Aplique o procedimento FSP à variável x_j ao nível de significância α_2;
12 $x_j \leftarrow x_j^{\boldsymbol{p}}$, em que $x_j^{\boldsymbol{p}}$ é a transformação escolhida para x_j pelo FSP;
13 **end**
14 **end**
15 **else**
16 Avalie a significância de x_j no modelo ao nível α_1;
17 **if** x_j *não é significativa* **then** Descarte x_j ;
18 **end**
19 $j \leftarrow j + 1$
20 **end**
21 **if** *o melhor modelo obtido no ciclo c é igual ao modelo obtido no ciclo* $c - 1$ **then**
22 **pare**;
23 **end**
24 $c \leftarrow c + 1$
25 **end**

O MFP discrimina variáveis contínuas de categóricas (linha 7). Para as contínuas,

aplica-se a seleção de funções FP apenas se as mesmas forem significativas; do contrário, elas são eliminadas (linhas 8 a 12). A transformação escolhida da variável é mantida em todos os ciclos subsequentes do procedimento. Por outro lado, não podem ser feitas transformações FP em regressoras categóricas; logo, o FSP não é aplicado a estas. Tais covariáveis são mantidas no modelo apenas se suas variáveis *dummy* forem conjuntamente significativas (linhas 16 e 17).

Cada iteração nova do MFP começará com as transformações selecionadas na iteração anterior e todas as covariáveis. Alcança-se a convergência, isto é, encontra-se o modelo mais adequado pelos critérios do algoritmo, quando o modelo final não muda de um ciclo para o outro (linhas 21 e 22). É possível ainda que não se alcance a convergência, como quando o MFP oscila entre dois modelos "ótimos" (ROYSTON; SAUERBREI, 2008).

2.6 Modelos Polinomiais Fracionários com Interações

Em modelos com várias covariáveis, frequentemente surgem situações em que há interações entre as mesmas. O MFP como descrito na Seção 2.5 não é capaz de modelar tais ocorrências. Royston e Sauerbrei (2004) propuseram o procedimento `MFPI` (*multiple fractional polynomials with interactions*) para tornar possível a consideração de interações dentro do paradigma dos modelos polinomiais fracionários.

Inicialmente, considerar-se-á o procedimento para a verificação de interações entre variáveis categóricas e contínuas. Considere um vetor de covariáveis $\boldsymbol{x}$, uma variável categórica z_1 e uma variável contínua z_2, estas últimas podendo ou não integrar o vetor $\boldsymbol{x}$. Aqui, consideraremos z_1 como binária, mas o algoritmo pode ser estendido para mais categorias, como mostram Royston e Sauerbrei (2004). Para investigar a existência de interação entre z_1 e z_2, o `MFPI` realiza um teste F entre o modelo com interação e o sem interação. As variáveis cuja interação será verificada devem ser selecionadas antes da aplicação do algoritmo. Seguem os passos do procedimento `MFPI`:

1. Aplique o procedimento MFP ao vetor de covariáveis $\boldsymbol{x}$. Seja $\boldsymbol{x}^*$ o vetor de covariáveis e transformações escolhidas pelo MFP.
2. Encontre o estimador de máxima verossimilhança para o modelo FP2 com z_2, incluindo z_1 e $\boldsymbol{x}^*$ no ajuste.
3. Para $j = 0, 1$ e potências $p_i,\ i = 1, 2$ defina novas variáveis regressoras z_{ji} dadas por

$$z_{ji} = \begin{cases} z_2^{p_i} & \text{se } z_1 = j \\ 0 & \text{caso contrário} \end{cases}$$

4. Realize o teste F entre os modelos com $z_1, z_{01}, z_{02},\ z_{11}, z_{12}, \boldsymbol{x}^*$ (com interação) e com $z_1, z_2^{p_1}, z_2^{p_2}, \boldsymbol{x}^*$ (sem interação).

Royston e Sauerbrei (2008) argumentam que o motivo de se usar uma transformação FP2 para a variável contínua está no fato de a família de transformações FP2 ser flexível. Evita-se o uso de modelos FPs para os diferentes níveis da variável categórica para se evitar sobreajuste. Note que o procedimento pode ser generalizado para variáveis categóricas não-binárias. De forma geral, para uma variável categórica com k categorias, serão criadas $2k$ variáveis no passo 3.

Também é possível investigar a interação entre duas variáveis contínuas quaisquer no contexto de modelos polinomiais fracionários. Tal investigação é útil quando há possíveis interações entre covariáveis que têm efeito possivelmente não linear na variável resposta. Royston e Sauerbrei (2008) propõem o algoritmo `MFPIgen` para realizar tal tarefa. Considere que se tem um vetor de covariáveis $\boldsymbol{x}$ e duas variáveis z_1 e z_2 cuja interação se quer testar. O `MFPIgen` consiste em:

1. Aplique um MFP(α^*) a $\boldsymbol{x}$ e, simultaneamente, um MFP($1, \alpha_2$) a z_1 e z_2.
2. Reajuste o MFP com os termos de interação entre todas as transformações selecionadas de z_1 e z_2.
3. Realize um teste F entre os modelos ajustados no passo 1 e no passo 2.
4. Considere todos os pares de preditores para interação, independentemente de sua significância no modelo sem interações.
5. Se mais de uma interação for detectada, utilize o método *stepwise forward* para adicionar mais interações ao modelo.

Há algumas diferenças entre o `MFPIgen` e o `MFPI` convencional. Primeiramente, no `MFPIgen`, $\boldsymbol{x}^*$ não é ajustado de forma independente das covariáveis cuja interação se quer testar. z_1 e z_2 são forçadas ao modelo no passo 1, independentemente de seu nível de significância. Ademais, não há necessariamente apenas um termo de interação entre z_1 e z_2. Por exemplo, suponha que se tenha escolhido transformações FP2 para ambas as regressoras. Nesse caso, há quatro termos de interação entre as variáveis, sendo um para cada par multiplicativo de potências de z_1 e z_2.

2.7 Métodos de Diagnóstico

As técnicas de diagnóstico para FPMs são semelhantes àquelas usadas em modelos de regressão lineares usuais, podendo ser estendidas a esses primeiros sem complicações. Estas incluem, além da verificação das suposições do modelo, a análise de resíduos, discutida por Belsley *et al.* (1980), Atkinson (1981) e Cook e Weisberg (1982), por exemplo. As técnicas de diagnóstico usuais para modelos lineares também envolvem a detecção de pontos alavanca, introduzida por Hoaglin e Welsch (1978). Há ainda as técnicas de eliminação de pontos (de influência global) e de avaliação da influência local, propostas por Cook (1977, 1986), respectivamente. Abordagens práticas de tais conceitos para modelos lineares são fornecidas por Paula (2012), Kutner *et al.* (2005), Hoffmann (2016), e Montgomery *et al.* (2021), por exemplo.

Apesar da facilidade com a qual as técnicas de diagnóstico usuais podem ser estendidas para FPMs, pode haver complicadores adicionais nestes últimos. Por exemplo, a presença de multicolinearidade entre as transformações selecionadas para uma variável pode afetar o uso das técnicas de diagnóstico supracitadas. Há ainda situações que exigem o uso de procedimentos específicos para a avaliação da qualidade de um FPM. Nesta seção, serão discutidas as situações em que os métodos de diagnóstico usuais podem não ser suficientes para FPMs e como se pode proceder em tais casos. Na Subseção 2.7.1, serão considerados os casos em que há valores extremos ou muito baixos nas covariáveis. Já a Subseção 2.7.2 abordará a avaliação da qualidade da seleção das variáveis e suas transformações nos FPMs.

2.7.1 Covariáveis com Valores Extremos ou Próximos de Zero

Um dos casos nos quais os FPMs exigem atenção especial é quando há covariáveis com valores extremos. Na regressão polinomial usual, tem-se que transformações de graus 2 ou 3 de regressores com valores elevados podem levar a valores preditos aberrantes. Já ao se lidar com polinômios fracionários, também é importante que haja cuidado com valores muito próximos de zero nas covariáveis. Por exemplo, potências negativas podem levar a transformações com valores muito elevados. Observações com tais características podem ser pontos influentes ou alavancas. Ademais, uma observação com uma covariável com valor extremo pode levar à escolha de um modelo (por exemplo, linear, FP1 ou FP2) diferente do que seria selecionado se a mesma não estivesse presente no conjunto de dados.

Pode-se avaliar a existência de pontos influentes através da distância de Cook, por exemplo. Uma solução para o problema de pontos cujos valores dos regressores afetam fortemente a escolha do modelo é realizar uma mudança na origem, isto é, uma translação. Este método é especialmente útil quando o problema é causado por valores próximos de zero para alguma covariável. Sendo análogas às correções de valores negativos das covariáveis, possíveis transformações para mudança de origem são discutidas na Seção 2.2. Note que, embora a mudança de origem possa melhorar o ajuste do modelo em alguns casos, deve-se ter critério ao usá-la, pois ela pode afetar negativamente a interpretação deste.

Há situações em que o modelo pode não se ajustar bem a dados cujas variáveis regressoras apresentam observações com valores muito elevados. Nestes casos, Royston e Sauerbrei (2008) propõem que se use uma transformação exponencial negativa, dada por

$$x^* = \exp\left\{-\frac{x}{\mathrm{SD}(x)}\right\},$$

em que $\mathrm{SD}(x)$ é o desvio-padrão amostral de x. Esta visa trazer as observações com valores mais elevados para a vizinhança das demais, reduzindo assim a sua influência no processo de estimação. Ademais, ela é recomendada sobre a transformação logarítmica, pois esta última pode produzir valores negativos, que não são compatíveis com os FPMs. Naturalmente, deve-se utilizar a exponencial negativa apenas quando se constatar que ela é necessária para o bom ajuste do modelo.

2.7.2 Estabilidade do Modelo

Outra área que exige atenção especial no contexto de FPMs é a análise de estabilidade. Diz-se que um modelo é estável, ou robusto, se pequenas alterações nos dados não afetarem, ou afetarem pouco, a seleção deste. Quando se obtém um ajuste, o que se espera idealmente é que pequenas variações nos valores das variáveis não causem grandes alterações na escolha do modelo. Perceba que aqui estamos nos referindo à estabilidade do modelo escolhido, isto é, se ele é linear, FP1, FP2, suas potências e as variáveis escolhidas, e não à influência das observações na estimação dos parâmetros do mesmo. A verificação da sensibilidade na estimação dos parâmetros pode ser feita através de métodos bem conhecidos para modelos lineares, como aqueles propostos por Hoaglin e Welsch (1978), Cook (1977) e Cook (1986).

Royston e Sauerbrei (2008) propõem um método empírico baseado no proposto por Sauerbrei e Schumacher (1992) para se avaliar a estabilidade de modelos, denominado critério

`BIF` (do inglês *bootstrap inclusion frequency*, frequência de inclusão bootstrap). Este consiste em se fazer B amostras de bootstrap (com reposição) das n observações presentes no conjunto de dados. Daí, para cada réplica bootstrap, se aplica um procedimento de seleção de variáveis, como o `BE`, e se avalia a frequência de inclusão de cada um dos regressores. Esta quantidade é dada por

$$\text{BIF}(x_j) = \frac{1}{B}\sum_{b=1}^{B} I_b(x_j), \quad j = 1, 2, \ldots, k,$$

em que k é o número de covariáveis em consideração, x_j é a j-ésima covariável e $I_b(x_j)$ é uma função indicadora definida por

$$I_b(x_j) = \begin{cases} 1, \text{ se a } j\text{-ésima covariável foi inclusa no modelo baseado na } b\text{-ésima réplica} \\ \quad \text{bootstrap,} \\ 0, \text{ caso contrário.} \end{cases}$$

Em outras palavras, o `BIF` de uma variável é a proporção de vezes em que ela foi inclusa no modelo final. Para um nível de significância fixo α, o `BIF` de uma variável estar próximo de 50% é uma evidência de que ela é apenas marginalmente significativa. `BIF`s próximos de 100% são indicadores que de uma variável regressora é fortemente significativa.

Uma limitação do critério `BIF` é sua dificuldade ao lidar com covariáveis correlacionadas, isto é, na presença de multicolinearidade. Royston e Sauerbrei (2008) pontuam que, frequentemente, quando há dois ou mais regressores correlacionados no modelo, apenas um deles tende a ser selecionado. Isto pode levar a valores irrealisticamente baixos do `BIF` para todas as variáveis correlacionadas envolvidas e à eliminação inapropriada delas do modelo final. O uso das frequências de inclusão bidimensionais pode amenizar o problema quando há conjuntos de apenas duas variáveis correlacionadas. Para uma investigação mais detalhada do uso de reamostragem bootstrap para checar a estabilidade de modelos que apresentam covariáveis com relações multidimensionais, consulte Royston e Sauerbrei (2003), por exemplo.

Outra limitação do critério `BIF` é sua incapacidade de tratar as escolhas de diferentes transformações de uma mesma covariável, como se dá nos FPMs. Royston e Sauerbrei (2003) propuseram um método para a verificação da estabilidade de modelos em casos em que há escolha de função, também baseado em bootstrap. Este é baseado na função estimada da variável x_j na b-ésima amostra bootstrap $\hat{f}_b(x_j)$ e sua padronização, dada por

$$\tilde{f}_b(x_j) = \hat{f}_b(x_j) - \frac{1}{n}\sum_{i=1}^{n} \hat{f}_b(x_{ij}), \quad j = 1, 2, \ldots, k,$$

em que x_{ij} é a i-ésima observação da j-ésima variável regressora do conjunto original de dados. Note que $\tilde{f}_b(x_j)$ tem média zero quando avaliada em todos os pontos dos dados. Considera-se que $\hat{f}_b(x_j) = 0$ se x_j não for inclusa no b-ésimo modelo.

A partir das B funções $\tilde{f}_b(x_j)$, pode-se avaliar a instabilidade da seleção das transformações escolhidas em um modelo de referência para x_j através de *bagging* (agregação bootstrap). Este método foi proposto por Breiman (1996) e consiste em se avaliar a média das funções escolhidas nas amostras bootstrap. O estimador obtido é

$$f_{\text{bag}}(x_j) = \frac{1}{|R|} \sum_{b \in R} \tilde{f}_b(x_j),$$

em que R é um conjunto de réplicas bootstrap e $|R|$ é a quantidade de réplicas em R. A variação entre as réplicas bootstrap e a função selecionada no modelo de referência pode ser expressa através de

$$\frac{1}{|R|} \sum_{b \in R} \left(\tilde{f}_b(x_j) - f_{\text{ref}}(x_j)\right)^2 = \frac{1}{|R|} \sum_{b \in R} \left(\tilde{f}_b(x_j) - f_{\text{bag}}(x_j)\right)^2 + \left(f_{\text{bag}}(x_j) - f_{\text{ref}}(x_j)\right)^2, \qquad (2.5)$$

em que $f_{\text{ref}}(x_j)$ é a transformação escolhida para x_j no modelo de referência, com $f_{\text{ref}}(x_j) = 0$ se x_j não tiver sido inclusa no mesmo. A Equação (2.5) pode ser reescrita como

$$T(x_j) = V(x_j) + D^2(x_j),$$

em que $T(x_j)$ é a variação total entre a função de referência e as funções selecionadas por bootstrap para x_j, $V(x_j)$ é a variância entre as funções das réplicas e $D^2(x_j)$ é o quadrado da distância entre a curva de referência e a curva estimada por *bagging*. A quantidade $V(x_j)$ pode ser interpretada como a contribuição da variação aleatória para a variabilidade das curvas. O que se espera de modelos estáveis é que $D^2(x_j)$ tenha contribuição relativamente pequena para $T(x_j)$.

Uma desvantagem do uso de $V(x_j)$ é o fato de a medida não levar em conta a frequência de inclusão da variável. Uma solução é usar

$$V_{\text{cond}}(x_j) = \frac{1}{|R_{x_j}|} \sum_{b \in R_{x_j}} \left(\tilde{f}_b(x_j) - \frac{1}{q} f_{\text{bag}}(x_j)\right)^2,$$

em que R_{x_j} é o conjunto das réplicas em que x_j é selecionada e

$$q = \frac{|R_{x_j}|}{|R|}$$

é a frequência de inclusão de x_j.

As medidas de estabilidade podem ser todas expressas através das observações originais x_{ij}, $i = 1, \ldots, n$. A variação é dada por

$$V_j = \frac{1}{n} \sum_{i=1}^{n} V(x_{ij}), \quad j = 1, \ldots, k,$$

o quadrado do desvio pode ser escrito como

$$D_j^2 = \frac{1}{n} \sum_{i=1}^{n} D^2(x_{ij}), \quad j = 1, \ldots, k$$

e a variação condicional é

$$V_{\text{cond}_j} = \frac{1}{n} \sum_{i=1}^{n} V_{\text{cond}}(x_{ij}), \quad j = 1, \ldots, k.$$

Uma discussão sobre os aspectos práticos das técnicas de análise de estabilidade apresentadas nesta subseção pode ser encontrada em Royston e Sauerbrei (2008).

3 ASPECTOS COMPUTACIONAIS

Modelos polinomiais fracionários demandam uma quantidade considerável de procedimentos iterativos. Muitas vezes, estes são utilizados com MLGs, que demandam métodos de otimização e cálculos complexos. A realização de tais processos de forma manual é demasiadamente onerosa, tornando-a virtualmente impraticável. Por causa disto, tem-se feito uso extensivo de métodos computacionais na prática dos modelos supracitados.

Uma diversidade de *softwares* tem sido utilizada para se ajustar e verificar FPMs. Entre os mais utilizados, encontram-se o SAS®, o STATA®, e o R, desenvolvido pelo R Core Team (2022). Destes, apenas o STATA® e o R têm no momento suporte implementado para polinômios fracionários, sendo este último um *software* de código aberto. No SAS®, não há procedimentos nativamente implementados para o ajuste de FPMs, mas o usuário pode criar *macros* para isto. Este trabalho se concentrará no uso do R para se ajustar FPMs. Informações sobre o uso do STATA para este fim podem ser encontradas em STATA (2022).

No *software* R, pode-se encontrar suporte para o uso de FPMs no pacote `mfp`, criado por Ambler e Benner (2022). Modelos lineares e MLGs podem ser ajustados diretamente pelo `mfp`. O pacote `mfp` também carrega a biblioteca `survival`, desenvolvida por Therneau e Grambsch (2000), que é direcionada ao ajuste de modelos para análise de sobrevivência, como o modelo de riscos proporcionais de Cox para dados censurados, por exemplo. Os aspectos do `mfp` relativos à análise de sobrevivência não serão abordados neste trabalho.

O objetivo deste capítulo é discutir, de forma introdutória, o ajuste de modelos com polinômios fracionários através do pacote `mfp`. Para isto, na Seção 3.1, será introduzido de maneira breve o pacote e suas principais características. Para ilustrar o uso das funcionalidades da biblioteca, será utilizado um conjunto de dados simples como exemplo. Este será apresentado na Seção 3.2. Note que ele será usado apenas para fins didáticos; para uma análise de dados mais realista utilizando o `mfp`, incluindo análise exploratória detalhada e diagnóstico, consulte o Capítulo 4. Na Seção 3.3, serão abordadas as funcionalidades do pacote para o ajuste dos modelos. Finalmente, na Seção 3.4, serão discutidos os aspectos computacionais do diagnóstico que são intrínsecos aos FPMs.

3.1 O Pacote `mfp`

O pacote `mfp` traz o ferramental básico para se ajustar modelos polinomiais fracionários no R. Entre as funções e classes de objetos mais relevantes da biblioteca, estão:

- **`fp`**: define um objeto polinomial fracionário. Objetos da classe são então usados para se ajustar um modelo adequado.
- **`mfp`**: seleciona o melhor modelo para os dados, usando no procedimento MFP descrito na Subseção 2.5. Leva em consideração as transformações polinomiais fracionárias definidas pelo usuário através da função `fp`.
- **`mfp.object`**: classe de objetos de modelos polinomiais fracionários ajustados. Carregam toda informação de interesse sobre um FPM.

Além das funções e classes supracitados, a biblioteca também inclui funções usadas para o ajuste de modelos de Cox e alguns conjuntos de dados usados como exemplos em Royston e Sauerbrei (2008). Para uma descrição completa de todos os elementos do pacote, consulte Ambler e Benner (2022). Caso não tenha o pacote instalado, pode-se utilizar os seguintes comandos no R para instalá-lo e carregá-lo:

```
> install.packages('mfp')
> require(mfp)
```

3.2 Dados

Este capítulo utilizará dados referentes à concentração de imunoglobulina-G no soro do sangue de crianças em função de sua idade. Eles pertencem ao *dataset* `ImmunoG` do pacote `Brq`, de Alhamzawi (2018), da linguagem R. O objetivo do uso deste conjunto de dados é apresentar um caso simples do uso de polinômios fracionários. De tal forma, pode-se fazer uma introdução adequada dos principais aspectos referentes à pratica computacional do ajuste de FPMs.

O conjunto de dados tem duas variáveis. A variável resposta é a concentração de imunoglobulina-G (em g/L) no soro do sangue de 298 crianças saudáveis de 6 meses a 6 anos. A variável explicativa é a idade (em anos) das crianças. Este *dataset* foi originalmente analisado por Isaacs *et al.* (1983), que a partir dos dados, propuseram intervalos de confiança para os níveis de imunoglobulina-A, imunoglobulina-G e imunoglobulina-M de acordo com a idade das crianças. Neste capítulo, o conjunto de dados será utilizado para ilustrar o uso das funções

do pacote `mfp` para se fazer o ajuste de um modelo polinomial fracionário. Cabe ressaltar que a metodologia usada por Isaacs *et al.* (1983) não foi a de polinômios fracionários, mas sim a regressão polinomial usual. Uma abordagem utilizando polinômios fracionários foi proposta por Royston (2017). Pode-se encontrar uma amostra das observações do *dataset* na Tabela 1.

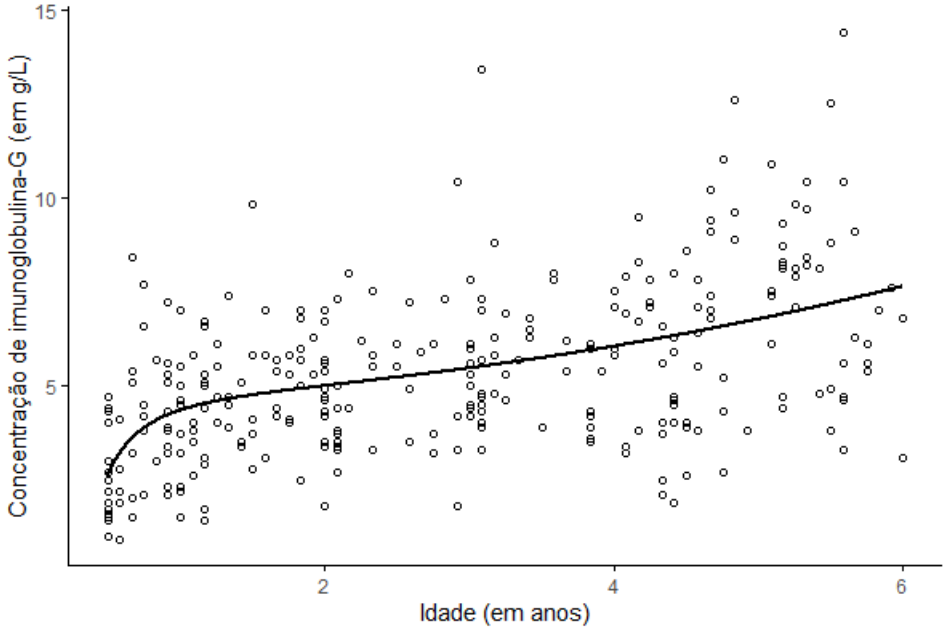
Tabela 1 – Amostra do *dataset* da imunoglobulina-G.

Nº da observação	**Imunoglobulina-G (em g/L)**	**Idade (em anos)**
101	5,7	1,6667
111	2,5	1,3333
133	3,4	2,0833
204	10,4	5,5833

Fonte: elaborada pelo autor.

Para nos ajudar a entender melhor o comportamento dos dados, a Figura 3 mostra o gráfico da concentração de imunoglobulina-G no soro do sangue *versus* a idade das crianças. O coeficiente de correlação linear de Pearson entre as variáveis é 0,5082. Não é possível identificar com clareza se há ou não um comportamento não-linear no gráfico. O método FPM poderá ser útil na identificação de tal comportamento, caso esteja presente.

Figura 3 – Gráfico de dispersão da imunoglobulina-G (em g/L) *versus* idade (em anos).



Fonte: elaborada pelo autor.

Nos trechos de código que serão apresentados neste capítulo, o *dataset* da imunoglobulina será referido como `ImmunogG`. A variável de imunoglobulina será referenciada por `IgG`,

e a idade por `Age`.

3.3 Ajuste do Modelo

Esta seção tem como objetivo demonstrar o uso do pacote `mfp` para se realizar o ajuste de um modelo polinomial fracionário. Logo, não serão feitas comparações entre classes diversas de modelos para o conjunto de dados usado ou discussões extensas sobre sua validade, distribuição da variável resposta ou análise de resíduos. Serão apresentados apenas os aspectos práticos referentes às opções de ajuste de FPMs e de seu diagnóstico. Para isto, a Subseção 3.3.1 explanará a função `mfp`, que é a principal função do pacote em questão, responsável pelo ajuste em si dos FPMs. Já a Subseção 3.3.2 discutirá o uso da função `fp`, que cuida da especificação dos polinômios fracionários do modelo. A Subseção 3.3.3 abordará alguns detalhes sobre a classe `mfp.object`.

3.3.1 Função `mfp`

A função `mfp` é responsável pela execução do procedimento MFP, descrito no Algoritmo 1. Esta é a principal função do pacote homônimo, e realiza tanto a seleção de variáveis quanto a seleção de suas transformações polinomiais fracionárias, visando a parcimônia do modelo.

Como visto nas Seções 2.4 e 2.5, o procedimento de seleção de transformações polinomiais fracionárias tem três parâmetros principais. O primeiro deles é α_1, que é o nível de significância nominal do método `BE` de seleção de variáveis. O segundo é α_2, o nível de significância nominal dos testes entre os modelos completos e encaixados. Lembre-se que α_1 está relacionado à propensão do modelo de selecionar uma variável, enquanto α_2 está relacionado à propensão do modelo de escolher FPs mais complexos. O terceiro parâmetro é gl, referente aos graus de liberdade que serão considerados para a variável. Este parâmetro determina o grau máximo da transformação polinomial fracionária que será considerada pelo modelo.

Os parâmetros supracitados estão entre os mais relevantes da função `mfp`. Entre estes, podemos citar:

- **`formula`**: um objeto de fórmula. Determina a forma do modelo considerado. Neste argumento, pode ser usada a função `fp`, que será descrita mais adiante.
- **`select`**: o mesmo que α_1, isto é, o nível de significância nominal do método BE para

todas as variáveis. Naturalmente, deve estar entre 0 e 1. O valor *default* é 1. Em outras palavras, por padrão a função mantém todas as variáveis no modelo.

- **alpha**: o mesmo que α_2, isto é, o nível de significância nominal dos testes usados para comparar os diferentes graus de FP para todas as variáveis. Deve estar entre 0 e 1, com valor *default* de 0,05.
- **maxits**: número máximo de iterações a serem executadas pelo algoritmo MFP. O valor *default* é 20.
- **rescale**: determina se os coeficientes dos regressores deve ser retornados em termos de sua escala original. Este parâmetro é útil, pois o MFP pode alterar a escala das variáveis para evitar problemas numéricos. O valor deve ser booleano. Seu *default* é `TRUE`.

Os outros parâmetros da função não serão abordados neste trabalho. Estes estão relacionados ao uso do MFP para modelos de Cox ou são parâmetros que já são comuns em funções usadas para se ajustar modelos no R, como `lm` e `glm`. Uma descrição completa dos parâmetros da função `mfp` pode ser encontrada em Ambler e Benner (2022).

A partir do uso da função, obtém-se um ajuste. O uso da função e a saída desta são ilustrados a seguir.

```
> fit <- mfp(data = ImmunogG, formula = IgG ~ fp(Age))
> fit
Call:
mfp(formula = IgG ~ fp(Age), data = ImmunogG)


Deviance table:
           Resid. Dev
Null model     1539.692
Linear model    1142.046
Final model     1106.351

Fractional polynomials:
    df.initial select alpha df.final power1 power2
Age          4      1  0.05        4     -2      2
```

```
Transformations of covariates:
               formula
Age I(Age^-2)+I(Age^2)

Rescaled coefficients:
Intercept      Age.1      Age.2
  4.81361   -0.55094    0.07883

Degrees of Freedom: 297 Total (i.e. Null);  295 Residual
Null Deviance:        1540
Residual Deviance: 1106     AIC: 1245
```

Na saída, encontra-se uma série de informações relevantes a respeito do ajuste do modelo e suas propriedades. Sob `Call`, apresenta-se a fórmula usada no ajuste deste. Em `Deviance table`, são dados os desvios dos modelos nulo, linear e selecionado. Na seção `Fractional polynomials`, pode-se encontrar as potências escolhidas para as variáveis e seus valores de gl, α_1 e α_2. O modelo selecionado foi um FP1, com vetor de potências $\boldsymbol{p} = (-2, 2)$. Sob `Transformations of covariates`, são dadas as transformações usadas em cada variável do modelo. Em `Rescaled coefficients`, são apresentados os valores dos parâmetros para cada termo do modelo. Neste caso, os parâmetros são um intercepto de $4,8136$, o efeito quadrático inverso da idade, de $-0,5509$, e o efeito quadrático da idade, de $0,0788$. Portanto, o modelo selecionado foi

$$\hat{y}_i = 4,8136 - 0,5509\texttt{Age}^{-2} + 0,0788\texttt{Age}^2, \quad i = 1, 2, \ldots, 298,$$

em que $\hat{y}_i$ é a concentração estimada de imunoglobulina-G no soro do sangue do indivíduo i (em g/L). Por fim, nas três últimas linhas da saída encontram-se os graus de liberdade do modelo, seus desvios nulo e residual e seu AIC, medida comumente usada na comparação de modelos.

Assim como no caso de objetos criados a partir das funções `lm`, é possível obter os resultados do ajuste através da função `summary`. Esta apresenta um resumo descritivo dos resíduos, além de informações detalhadas a respeito das estimativas dos parâmetros do modelo, bem como a significância destes. Segue um exemplo do uso da função `summary` no FPM ajustado.

```
> summary(fit)

Call:
glm(formula = IgG ~ I(Age^-2) + I(Age^2), data = ImmunogG)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5360  -1.3168  -0.0418   1.1287   7.8950

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.81361    0.22076  21.804  < 2e-16 ***
I(Age^-2)   -0.55094    0.12773  -4.313 2.20e-05 ***
I(Age^2)     0.07883    0.01294   6.093 3.45e-09 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for gaussian family taken to be 3.750344)

    Null deviance: 1539.7  on 297  degrees of freedom
Residual deviance: 1106.4  on 295  degrees of freedom
AIC: 1244.6

Number of Fisher Scoring iterations: 2
```

A função `mfp` tem algumas limitações. Primeiramente, por meio dela, não é possível executar de forma direta os algoritmos `MFPI` ou `MFPIgen`, descritos na Seção 2.6, para investigar interações. Também não há no pacote funções específicas para esta finalidade. Ademais, não é possível limitar ou expandir o conjunto S de transformações potência, conforme discutido na Seção 2.2, o que é essencial em algumas aplicações.

A partir da função `mfp`, pode-se entender a função `fp`, essencial para o uso adequado desta primeira, que será discutida a seguir.

3.3.2 Função `fp`

A função `fp` é responsável pela definição das transformações polinomiais fracionárias que se deseja usar para determinada variável. Note que, para diferentes variáveis, pode-se escolher diferentes valores para α_1, α_2 e gl. Em algumas situações, por exemplo, pode ser de interesse do pesquisador testar configurações específicas para a transformação de determinada variável, usando diferentes valores de α_2 e de gl para diferentes regressores. Em outras situações, necessita-se que determinadas variáveis sejam inclusas no estudo, independentemente de sua significância ser menor que a das demais, como explicitam Royston e Sauerbrei (2008). Neste último caso, faz-se $\alpha_1 = 1$ apenas para os regressores que se deseja "fixar" no modelo. Logo, um método satisfatório de seleção de modelos polinomiais fracionários deve tomar em consideração tais possibilidades.

A função `fp` foi pensada para atender às necessidades supracitadas. Esta tem cinco parâmetros, a saber:

- **`x`**: a variável de entrada.
- **`df`**: o parâmetro de graus de liberdade (gl) que serão usados para a variável `x`. Para `df = 1`, tem-se que o grau máximo da transformação considerada para a variável será a identidade (linear). Quando `df = 2` o grau máximo considerado será um FP1. Para `df = 4`, o grau máximo considerado será um FP2. Valores além destes não são permitidos. O valor *default* é `df = 4`.
- **`select`**: o mesmo que α_1, isto é, o nível de significância nominal do método `BE` para a variável. Naturalmente, deve estar entre 0 e 1. O valor *default* é 1.
- **`alpha`**: o mesmo que α_2, isto é, o nível de significância nominal dos testes usados para comparar os diferentes graus de FP para a variável. Deve estar entre 0 e 1, com valor *default* de 0,05.
- **`scale`**: determina se deve ser usada ou não uma pré-transformação para evitar problemas numéricos. Deve ser um valor booleano, com o *default* sendo `TRUE`.

Os parâmetros da função `fp` permitem significativa flexibilidade no modo como são tratadas as covariáveis na seleção pelo algoritmo MFP. Note que, se ao chamarmos a função `mfp` não fizermos uso da função `fp`, será ajustado um modelo linear. Por exemplo, ao se executar o seguinte código, tem-se um modelo linear, sem nenhuma transformação ser realizada:

```
> fit2 <- mfp(data = ImmunogG, formula = IgG ~ Age)
```

```
> fit2
Call:
mfp(formula = IgG ~ Age, data = ImmunogG)


Deviance table:
                  Resid. Dev
Null model         1539.692
Linear model         1142.046
Final model         1142.046

Fractional polynomials:
    df.initial select alpha df.final power1 power2
Age          1      1  0.05        1      1      .


Transformations of covariates:
    formula
Age     Age

Rescaled coefficients:
Intercept      Age.1
   3.3640     0.6951

Degrees of Freedom: 297 Total (i.e. Null);  296 Residual
Null Deviance:            1540
Residual Deviance: 1142         AIC: 1252
```

Ao fazermos α_2 igual a 1, forçamos a seleção da transformação polinomial mais complexa, dentro dos limites impostos pelo parâmetro de graus de liberdade. Por exemplo, na saída a seguir, ao se fazer `alpha = 1`, tem-se que o modelo escolhido é um FP2 com $\boldsymbol{p} = (-2, 2)$, que neste caso, é igual ao modelo escolhido ao se fazer `alpha = 0.05`:

```
> fit3 <- mfp(data = ImmunogG, formula = IgG ~ fp(Age, alpha = 1))
```

```
> fit3
Call:
mfp(formula = IgG ~ fp(Age, alpha = 1), data = ImmunogG)


Deviance table:
                  Resid. Dev
Null model         1539.692
Linear model         1142.046
Final model         1106.351

Fractional polynomials:
    df.initial select alpha df.final power1 power2
Age          4      1     1        4     -2      2


Transformations of covariates:
               formula
Age I(Age^-2)+I(Age^2)

Rescaled coefficients:
Intercept      Age.1      Age.2
  4.81361   -0.55094    0.07883

Degrees of Freedom: 297 Total (i.e. Null);  295 Residual
Null Deviance:            1540
Residual Deviance: 1106         AIC: 1245
```

Similarmente, em um contexto multivariado, fazer `select = 1` forçará a seleção de uma variável no modelo. Variando o valor de `df`, pode-se controlar o grau máximo da tansformação FP utilizada para o regressor. Considere, por exemplo, o ajuste anterior, que força a escolha do grau mais elevado para o FP. Limitando os graus de liberdade com `gl = 2` e forçando a transformação mais complexa, o modelo escolhido é um FP1:

```
> fit4 <- mfp(data = ImmunogG, formula = IgG ~ fp(Age, alpha = 1, df = 2))
> fit4
Call:
mfp(formula = IgG ~ fp(Age, alpha = 1, df = 2), data = ImmunogG)


Deviance table:
                  Resid. Dev
Null model         1539.692
Linear model         1142.046
Final model         1132.003

Fractional polynomials:
    df.initial select alpha df.final power1 power2
Age          2      1     1        2    0.5      .


Transformations of covariates:
       formula
Age I(Age^0.5)

Rescaled coefficients:
Intercept      Age.1
    1.790      2.217

Degrees of Freedom: 297 Total (i.e. Null);  296 Residual
Null Deviance:            1540
Residual Deviance: 1132         AIC: 1249
```

Perceba que a função `fp` tem algumas limitações. Primeiramente, o grau máximo do FP que pode ser considerado nela é 2. Royston e Sauerbrei (2008) mostram, no entanto, que em algumas ocasiões faz-se necessário o uso de polinômios fracionários de graus mais elevados. Logo, dados que apresentam comportamento muito complexo não podem ser ajustados

adequadamente com o uso desta função. Também não é possível limitar ou expandir o conjunto S na função, assim como se dá com a função `mfp`.

3.3.3 Classe `mfp.object`

A função `mfp` retorna objetos da classe `mfp.object`. Tal classe contém toda a informação relevante a respeito da chamada da função `mfp`, do modelo ajustado e do processo de ajuste. No contexto dos MLGs, também herda os atributos de objetos gerados pela função `glm`, chamada internamente para se ajustar o modelo. O objeto também pode conter informações referentes a um modelo de Cox, como os objetos gerados pela função `coxph` da biblioteca `surv`, caso a função `mfp` seja utilizada para ajustar um modelo desta última classe. Atributos referentes ao MLGs e ao modelo de Cox não serão abordados neste trabalho. Aqui serão apresentados apenas atributos que são intrínsecos aos FPMs.

A classe `mfp.object` pode ser útil na análise do modelo ajustado, especialmente no que se refere ao diagnóstico deste. Atualmente, não há funções no R para se realizar o diagnóstico dos aspectos que são intrínsecos aos FPMs. Logo, o pesquisador deve fazer uso dos atributos do objeto do modelo ajustado para fazer o diagnóstico por conta própria. Entre os atributos mais importantes da classe `mfp.object` para este contexto, encontram-se:

- **`powers`**: matriz contendo as potências selecionadas para cada variável no modelo final. As linhas da matriz são dadas pelos nomes das variáveis. O atributo `powers` tem duas colunas: `power1` e `power2`, referentes às potências 1 e 2 escolhidas para cada variável, respectivamente. Caso o modelo final para a variável envolva um FP1, o valor de sua coluna `power2` é apresentado como `NA`. Caso determinada variável não tenha sido selecionada para o modelo final, todas as entradas de sua linha na matriz são `NA`.
- **`dev`**: o desvio do modelo final selecionado.
- **`fptable`**: tabela que contém as transformações do modelo final para cada variável.

Os atributos `powers` e `fptable` são úteis na análise de estabilidade do modelo. Sua relação de potências escolhidas pode ser utilizada para se comparar a diferença entre a seleção de transformações em diferentes modelos. O exemplo a seguir mostra como são suas saídas. Note que, em `fptable`, também são apresentados os parâmetros `select` e `alpha` para a variável. Das duas saídas abaixo, torna-se claro que o modelo escolhido foi um FP2 com $\boldsymbol{p} = (-2, 2)$. A matriz `powers`, em especial, será bastante útil no uso de *bagging* para se avaliar a variabilidade das funções escolhidas para as variáveis, como será visto na Seção 3.4.

```
> fit$powers
    power1 power2
Age     -2      2

> fit$fptable
    df.initial select alpha df.final power1 power2
Age          4      1  0.05        4     -2      2
```

O atributo `dev` é útil na detecção de observações que causam variações discrepantes no desvio do modelo. A impressão do atributo é bastante simples e de fácil interpretação, como mostra a saída a seguir. Nela, o desvio do modelo selecionado é 1106,351. Os desvios de modelos ajustados com observações excluídas podem ser usados para se verificar o efeito destas sobre o processo de escolha de transformações.

```
> fit$dev
[1] 1106.351
```

Uma descrição completa dos atributos intrínsecos à classe `mfp.object` pode ser encontrada em Ambler e Benner (2022).

3.4 Análise de Diagnóstico

Esta seção tem como objetivo apresentar uma introdução às técnicas de diagnóstico para FPMs no R. Não serão abordados aqui assuntos referentes ao diagnóstico de modelos normais ou modelos lineares generalizados. Métodos para estes são amplamente documentados na literatura. Para isso, veja, por exemplo, Paula (2012). Serão discutidos nesta seção os tópicos intrínsecos à detecção de inadequabilidade nos FPMs. Estes abrangem a identificação de valores com efeitos discrepantes a e verificação da estabilidade do modelo.

Como já comentado, não há funções no R para a realização do diagnóstico dos aspectos intrínsecos aos modelos polinomiais fracionários. Logo, os usuários precisam implementar tais métodos por conta própria. A elaboração de *scripts* de diagnóstico para tarefas simples, até mesmo para a confecção de um gráfico, podem ser laboriosas, especialmente se o modelo incluir muitas variáveis e transformações de graus elevados. *Scripts* para variados métodos de diagnóstico usados neste trabalho podem ser encontrados nos Apêndices A, B e C. Os pacotes

utilizados nas funções foram `ggplot2`, de Wickham (2016), `stringr`, de Wickham (2022) e `ggExtra`, de Attali e Baker (2022).

Para verificar a existência de pontos que exercem peso desproporcional sobre a escolha do modelo, pode-se utilizar um gráfico da função escolhida por cada variável. No caso dos FPMs, faz-se necessária a construção de funções para cada variável com base na matriz `powers` da classe `mfp.object`. Uma função que automatiza este processo, chamada `plot_mfp`, é documentada no Apêndice A. Seus argumentos são:

- **`mfp.object`**: objeto com um FPM ajustado.
- **`data`**: uma *dataframe* com os dados utilizados para se ajustar o modelo contido em `mfp.object`.
- **`x`**: nome da variável do eixo x, isto é, a variável explicativa para a qual será feito o gráfico.
- **`y`**: nome da variável do eixo y. No contexto de diagnóstico, deve ser a variável resposta.
- **`xlab` e `ylab`**: Nomes para os eixos x e y, respectivamente. Caso não sejam determinados, receberão os nomes das variáveis `x` e `y`.

Um exemplo de uso da função é apresentado abaixo, e a saída é mostrada na Figura 4. Note que, visualmente, não se pode identificar pontos que exercem efeito desproporcional sobre o formato do curva. Caso houvesse, notaríamos a curva ajustada de forma a passar próximo a determinado ponto influente.

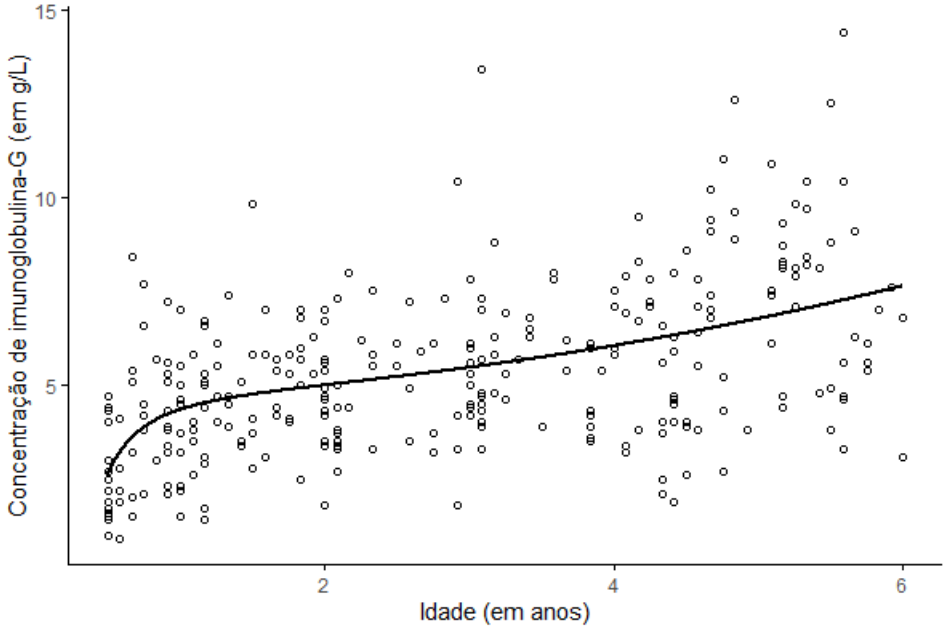
```
plot_fp(
    fit, data = ImmunogG, x = "Age", y = "IgG",
    xlab = "Idade (em anos)",
    ylab = "Concentração de imunoglobulina-G (em g/L)"
)
```

Outro elemento importante na análise de diagnóstico dos FPMs é a análise da estabilidade da escolha do modelo. Como discutido na Subseção 2.7.2, esta pode ser feita através de *bagging*. Uma função para se determinar os valores de $T(x_j)$, $V_{\text{cond}}(x_j)$ e $D^2(x_j)$ pode ser encontrada no Apêndice B, chamada `bagging_diagnostics`. Esta retorna as frações de D^2 da variação total para cada variável do modelo. Os parâmetros da função são:

- **`mfp.object`**: o objeto com o modelo de referência ajustado.
- **`formula`**: objeto com a fórmula do modelo de referência.
- **`data`**: *dataframe* com os dados usados no ajuste do modelo de referência.
- **`bootstrap_size`**: tamanho das réplicas *bootstrap* que serão utilizadas (o mesmo que B).

- **`seed`**: a semente aleatória inicial que será usada para gerar as réplicas. É útil para se garantir a reprodutibilidade dos resultados.
- **`B`**: o número de réplicas *bootstrap* a serem usadas.
- **`alpha`**: nível de significância nominal dos testes de razão de verossimilhança. Deve ser, de preferência, igual ao `alpha` do modelo de referência.
- **`select`**: nível de signifcância nominal do procedimento `BE`. Deve ser, de preferência, igual ao `select` do modelo de referência.

Figura 4 – Gráfico de dispersão da imunoglobulina-G (em g/L) *versus* idade (em anos) com a curva ajustada pelo modelo `fit`.



Fonte: elaborada pelo autor.

Ao se executar a função, obtém-se uma saída semelhante à abaixo. Nela, vê-se que a fração de D^2 para `Age` (a idade) é de aproximadamente 0,02%. Este é um forte indício de que o processo de seleção modelo é estável, não estando o modelo escolhido distante daquele estimado por *bagging*.

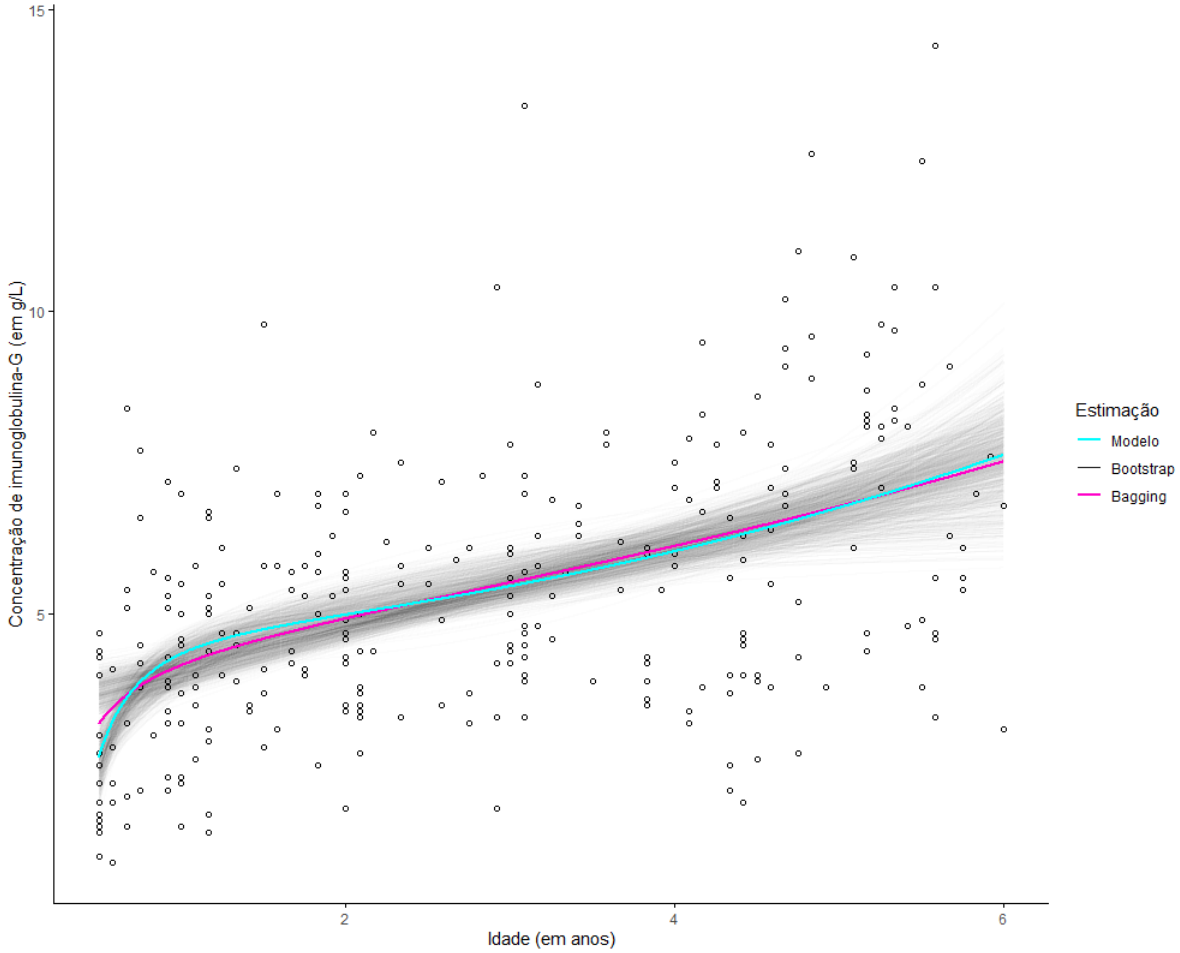
```
> bagging_diagnostics(fit, formula = IgG ~ fp(Age),
    data = ImmunogG, bootstrap_size = 200,
    seed = 474276, alpha = 0.05, select = 0.05, B = 1000
)
        Age
1.75167e-05
```

Um outro modo de se avaliar a estabilidade do modelo é fazê-lo de forma gráfica. Para isto, Royston e Sauerbrei (2008) recomendam que se faça um gráfico contendo os diversos modelos estimados por *bootstrap* para cada variável. Depois, pode-se comparar a variação destes com as funções de referência e estimada por *bagging*. Uma função para se realizar tal tarefa pode ser encontrada no Apêndice C. Seus argumentos são iguais aos da `bagging_diagnostics`, além dos parâmetros `x`, `y`, `xlab` e `ylab` da função `plot_fp`. O exemplo a seguir demonstra o uso da desta. A saída é apresentada na Figura 5.

```
> plot_bagging(fit, formula = IgG ~ fp(Age), data = ImmunogG,
                bootstrap_size = 200, x = "Age", y = "IgG",
                xlab = "Idade (em anos)",
                ylab = "Concentração de imunoglobulina-G (em g/L)",
                seed = 474276, alpha = 0.05, select = 0.05,    B = 1000
)
```

Pela Figura 5, depreende-se que a escolha do modelo é, de fato, estável. Nela, as regiões mais escuras são as que apresentam a maior concentração de funções *bootstrap*. Note que as funções de referência e de *bagging* são, de forma geral, próximas uma da outra. Por outro lado, as funções estimadas por *bootstrap* tendem a estar mais distantes entre si. Estas últimas são todas semelhantes nas regiões mais densas dos dados. Nas caudas, porém, elas apresentam maior variação, como pontuado por Royston e Sauerbrei (2008).

Figura 5 – Gráfico do modelo `fit` e suas curvas estimadas por *bootstrap* e *bagging*.



Fonte: elaborada pelo autor.

4 APLICAÇÃO

Esta seção tem como objetivo demonstrar a flexibilidade dos modelos polinomiais fracionários junto aos MLGs de forma prática. Para isso, será avaliado um conjunto de dados extraído de Luke *et al.* (1997). Royston e Sauerbrei (2008) usam o conjunto de dados para ilustrar aspectos diversos da prática dos FPMs. Aqui, faremos uma análise do *dataset*, comparando seu ajuste por FPMs e seu ajuste por modelos polinomiais tradicionais.

Para cumprir este objetivo, este capítulo está organizado em quatro seções. Na Seção 4.1, é feita uma apresentação dos dados e sua análise exploratória, com o intuito de explorar as opções de modelagem para estes. Na Seção 4.2, é ajustado um modelo linear para os dados, sendo também realizada uma análise de diagnóstico para este. Na Seção 4.3, é ajustado um modelo quadrático para os dados, junto com uma análise de diagnóstico. Um modelo cúbico é ajustado na Seção 4.4, também nos moldes da Seção 4.2. Um modelo polinomial fracionário é avaliado na Seção 4.5. Nesta seção também são examinados os aspectos da análise de diagnóstico intrínsecos aos FPMs para o modelo obtido. Por fim, na Seção 4.6 são sintetizadas as propriedades dos modelos propostos, com o intuito de auxiliar na decisão de qual modelo é o mais adequado.

4.1 Dados

O conjunto de dados tem duas variáveis relevantes. A variável resposta é a porcentagem de gordura corporal em uma amostra de 327 mulheres negras dos Estados Unidos. A variável explicativa é seu IMC (índice de massa corporal) em kg/m^2. No estudo de Luke *et al.* (1997), também são inclusas as amostras de pessoas da Jamaica e da Nigéria. O objetivo do estudo é determinar o quão bem a porcentagem de gordura corporal de um indivíduo pode ser predita pelo seu índice de massa corporal, que é mais simples de se obter. Royston e Sauerbrei (2008) analisam o conjunto de dados e propõem um FPM para se modelar a gordura corporal em função do IMC. Uma amostra da observações é apresentada na Tabela 2.

Tabela 2 – Amostra do *dataset* da gordura corporal.

Nº da observação	IMC	Porcentagem de gordura corporal
112	26,40	38,11
198	39,34	48,08
222	43,20	55,54
284	26,86	40,01

Fonte: elaborada pelo autor.

4.1.1 Análise descritiva

Esta subseção objetiva analisar o comportamento das variáveis do *dataset* da gordura corporal. Desta forma, pode-se tomar decisões acertadas sobre o tipo de modelagem mais eficaz para para os dados.

A Tabela 3 apresenta algumas medidas descritivas dos dados. A partir dela, depreende-se que a porcentagem de gordura apresenta uma leve assimetria à esquerda, enquanto o IMC tem uma leve assimetria à direita. As duas apresentam coeficientes de variação semelhantes, com o IMC tendo desvio-padrão maior. Note que ambas variáveis são estritamente positivas, com a porcentagem de gordura estando limitada ao intervalo $(0, 100)$. Dada sua assimetria modesta, pode-se cogitar um modelo normal para esta última variável. Outra possibilidade para ela seria um modelo beta, dado que esta é referente a uma proporção. Neste trabalho, nos limitaremos ao ajuste de modelos normais para os dados. Sugerimos como trabalho futuro o ajuste de um modelo beta junto com polinômios fracionários a estes.

Tabela 3 – Medidas descritivas das variáveis do conjunto de dados da gordura corporal.

Variável	Mínimo	1ºquartil	Mediana	Média	3ºquartil	Máximo	DP	CV
% de Gordura	11,2343	37,8318	42,4254	42,200	47,4893	58,5648	7,9231	0,1878
IMC	14,6	25,2842	30,1	30,9358	35,8688	56,8	7,9202	0,2560

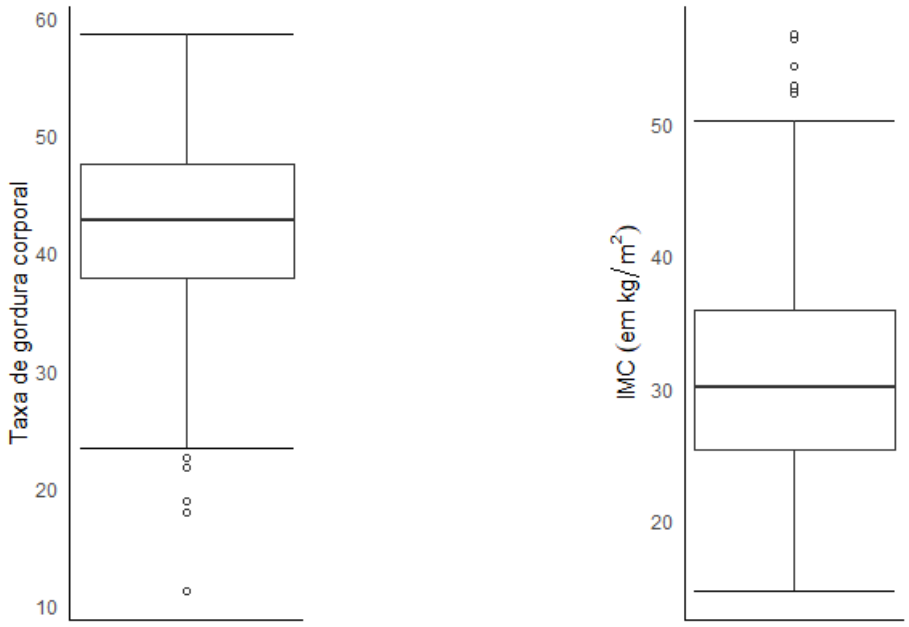
Fonte: elaborada pelo autor.

A Figura 6 traz os *boxplots* das variáveis em consideração. Como visto através da tabela, ambas as variáveis apresentam assimetria. A porcentagem de gordura corporal tem assimetria à esquerda com alguns valores discrepantes, enquanto o IMC tem assimetria e valores discrepantes à direita. Percebe-se também que o IMC tem uma assimetria levemente maior.

A Figura 7 mostra o gráfico de dispersão da porcentagem de gordura corporal *versus* o IMC em kg/m^2. O coeficiente de correlação linear de Pearson é de 0,8845. Não obstante, note que a relação entre as duas variáveis é possivelmente não-linear. Para valores mais baixos do

IMC, a variável resposta cresce mais rapidamente. À medida que o índice de massa corporal aumenta, a inclinação do gráfico diminui. Para os valores mais elevados o IMC, a inclinação se aproxima de zero, assemelhando-se a uma assíntota horizontal. Dado que a porcentagem de gordura corporal em um ser humano é um valor limitado, a hipótese de haver uma assíntota na relação deve ser considerada. Se esse for o caso, as potências da transformação do IMC no FPM ajustado serão todas negativas. Isto será avaliado na Seção 4.5.

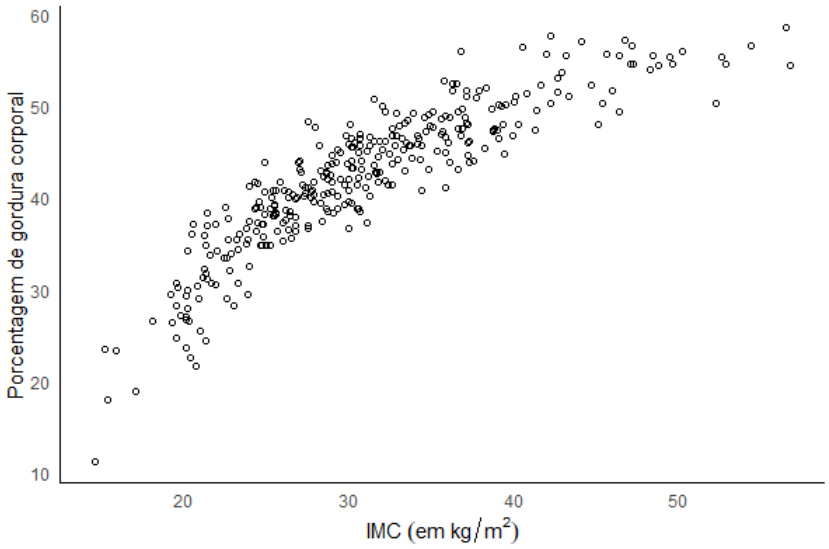
Figura 6 – *Boxplots* das variáveis do conjunto de dados da gordura corporal.
(a) Porcentagem de gordura corporal (b) IMC (em kg/m^2)



Fonte: elaboradas pelo autor

Nas seções seguintes, a variável de percentual de gordura corporal será referenciada por `pbfm`, e o IMC por `bmi`. A partir de agora, começaremos a explorar meios de modelar a relação entre as duas variáveis.

Figura 7 – Gráfico de dispersão da porcentagem de gordura corporal *versus* IMC (em kg/m^2).



Fonte: elaborada pelo autor.

4.2 Modelo Linear

Começaremos a análise com o caso mais simples, um modelo linear. Sua forma funcional é

$$\texttt{pbfm} = \beta_0 + \beta_1 \texttt{bmi}^* + e_i, \quad i = 1, \ldots, 327,$$

em que $e_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\forall i$ e $\texttt{bmi}^*$ é o IMC centralizado na média. A centralização foi realizada para tornar o intercepto interpretável.

A Tabela 4 traz uma análise inferencial dos parâmetros do modelo. Tem-se que o valor estimado para o intercepto é $42,2$. Em outras palavras, o percentual de gordura corporal esperado para um indivíduo com IMC de 30,9358 kg/m^2 (valor médio do IMC) é $42,2\%$. O valor estimado para β_1 é de $0,8848$. Isto significa que, ao se aumentar o IMC de um indivíduo em 1 kg/m^2, tem-se uma variação esperada de $0,8848\%$ em sua taxa de gordura corporal. Para verificar a qualidade do ajuste, precisa-se realizar uma análise de diagnóstico do modelo.

Tabela 4 – Análise inferencial do modelo linear.

Parâmetro	**Valor estimado**	**Erro-padrão**	**Valor t**	**Valor-p**
β_0	$42,2000$	$0,2048$	$206,11$	$< 0,0001$
β_1	$0,8848$	$0,0258$	$34,17$	$< 0,0001$

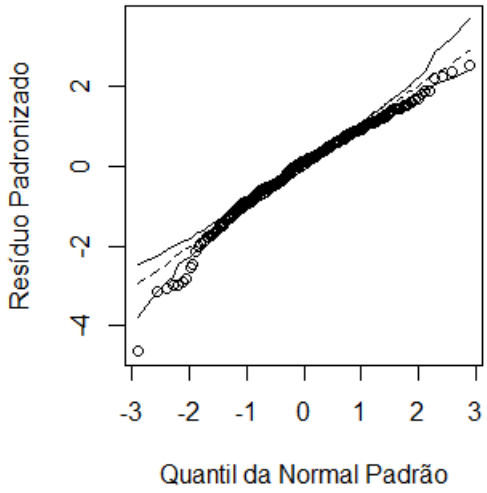
Fonte: elaborada pelo autor.

4.2.1 Análise de Diagnóstico

Os pontos importantes a se verificar no ajuste de um modelo são sua adequação às suposições e sua sensibilidade. Nesta subseção, serão avaliados estes dois pontos do modelo ajustado anteriormente. Algumas das funções de diagnóstico utilizadas neste trabalho são baseadas naquelas disponibilizadas por Paula (2012), podendo ser encontradas em `http://www.ime.usp.br/giapaula`.

A Figura 8 mostra o gráfico de quantis-quantis com envelopes simulados para os resíduos padronizados do modelo linear. Note que há uma evidente violação da suposição de normalidade por parte destes. Várias observações ficam fora dos envelopes simulados. Isto se dá, em especial, para aquelas com os valores mais baixos dos resíduos. Este é um indício de ajuste ruim por parte do modelo proposto.

Figura 8 – Gráfico de quantis-quantis com envelopes simulados a 95% de confiança para o modelo linear.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

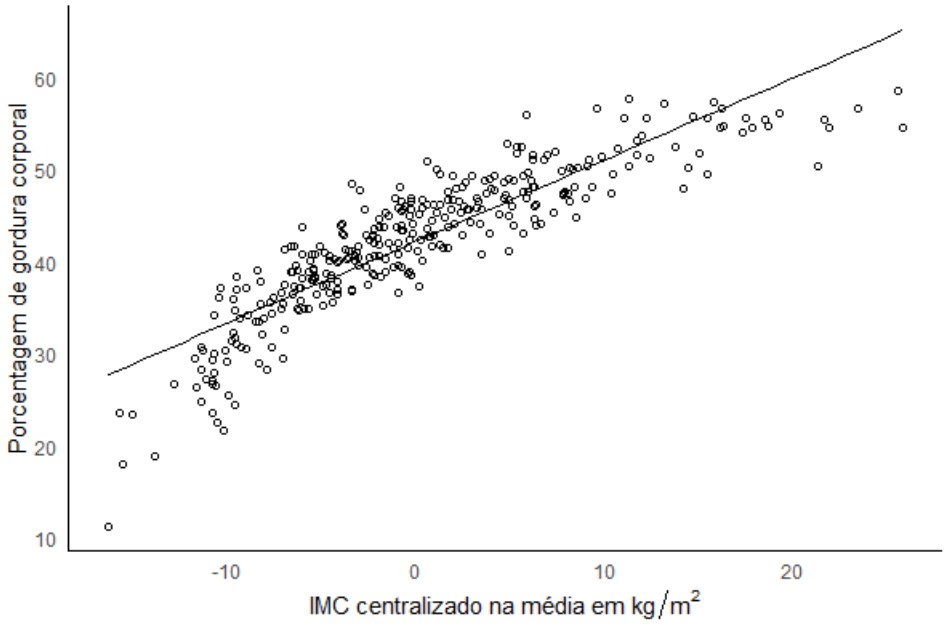
Um gráfico da função de regressão estimada para os dados, junto às observações do *dataset*, pode ser visto na Figura 9. Percebe-se que a função não se ajusta bem para valores nas caudas. Nestas, o valor médio estimado é claramente maior que o real. Tal problema no ajuste, somado à não normalidade dos resíduos, é uma evidência de que o ajuste obtido pelo modelo linear não é confiável.

Por fim, os gráficos de influência, alavancagem e pontos aberrantes do modelo

linear são apresentados na Figura 10. Nestes, observa-se um reflexo do ajuste insatisfatório às observações nas caudas. Três pontos se destacam entre os gráficos: #97, #304 e #315. Suas características são:

- **#97**: apresenta IMC próximo ao valor máximo (52,3 kg/m^2). A taxa de gordura corporal também está próxima ao valor máximo (50,35%).
- **#304**: apresenta os valores mínimos tanto para o IMC (14,6 kg/m^2) quanto para a gordura corporal (11,23%).
- **#315**: apresenta o valor máximo do IMC (56,8 kg/m^2) e um valor próximo ao máximo da gordura corporal (58,56%).

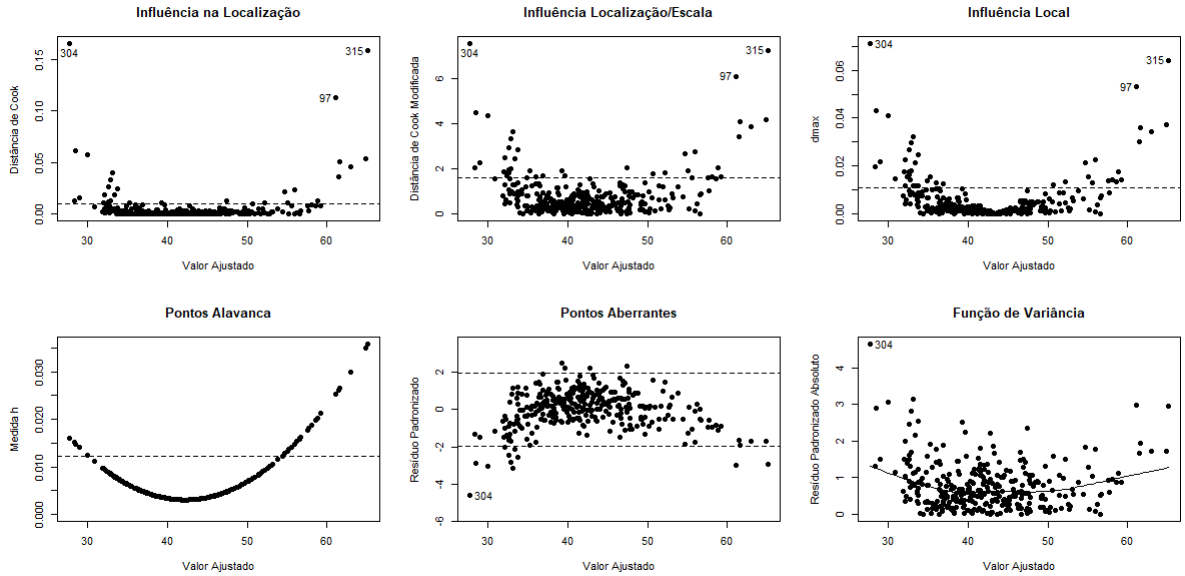
Figura 9 – Gráfico da taxa de gordura corporal *versus* IMC centralizado (em kg/m^2) com a curva de regressão do modelo linear.



Fonte: elaborada pelo autor.

A variação nas estimativas dos parâmetros acarretadas pela eliminação dos pontos que se destacaram é mostrada na Tabela 5. Nota-se que tais pontos, individual e conjuntamente, causam mudanças modestas nos estimadores. A maior variação é causada pela remoção dos pontos #97 e #304, no estimador $\hat{\beta}_1$, sendo de $2,97\%$. Conclui-se que o modelo não apresenta grandes problemas relacionados à sua sensibilidade. Por outro lado, deixa de satisfazer as suposições de normalidade por parte dos resíduos e não se ajusta bem aos dados nas caudas. Ademais, existe ainda a possibilidade de o modelo estimar valores fora do intervalo $(0, 100)$ para a taxa de gordura corporal. Portanto, outros modelos devem ser considerados.

Figura 10 – Gráficos de influência, alavancagem e pontos aberrantes para o modelo linear.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

Tabela 5 – Estimativas dos parâmetros do modelo linear após se retirar os pontos influentes.

Modelo	$\hat{\beta}_0$	$\hat{\beta}_1$
Completo	$42,2000 \pm 0,2048$	$0,8848 \pm 0,0259$
Sem #97	$42,2337 \pm 0,2026$ $0,07\%$	$0,8963 \pm 0,0259$ $1,30\%$
Sem #304	$42,2513 \pm 0,1989$ $0,12\%$	$0,8714 \pm 0,0253$ $-1,52\%$
Sem #315	$-42,2337 \pm 0,2027$ $0,08\%$	$0,8988 \pm 0,0260$ $-1,58\%$
Sem #97 e #304	$42,2838 \pm 0,1968$ $0,20\%$	$0,8828 \pm 0,0253$ $-0,23\%$
Sem #97 e #315	$-42,2696 \pm 0,2003$ $0,17\%$	$0,9111 \pm 0,0260$ $2,97\%$
Sem #304 e #315	$42,2835 \pm 0,1968$ $0,20\%$	$0,8851 \pm 0,0254$ $0,03\%$
Sem #97, #304 e #315	$42,3181 \pm 0,1945$ $0,28\%$	$0,8973 \pm 0,0254$ $1,42\%$

Fonte: elaborada pelo autor.

4.3 Modelo Quadrático

Dada a inadequação do modelo linear, prosseguimos nossa análise com um modelo quadrático. Sua forma funcional é

$$\texttt{pbfm} = \beta_0 + \beta_1 \texttt{bmi}^* + \beta_2 (\texttt{bmi}^*)^2 + e_i, \quad i = 1, \ldots, 327,$$

em que $e_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\forall i$. O IMC foi novamente centralizado na média para permitir uma interpretação para o intercepto.

A Tabela 6 mostra a análise inferencial dos parâmetros do modelo. Note que eles são todos fortemente significativos. Tem-se que β_0, estimado em $43,7309$, é o intercepto do modelo. Portanto, a taxa de gordura corporal esperada para um indivíduo com IMC de $30,9358$ kg/m^2 é de 43,7309%. β_1 (estimado em 1,0176) e β_2 (estimado em $-0,0245$) são, respectivamente, o efeito linear e o efeito quadrático do IMC sobre a taxa de gordura corporal nas mulheres negras. Note que o erro-padrão do estimador do intercepto é substancialmente maior que os dos estimadores dos outros parâmetros. Agora, precisa-se avaliar se o modelo se adequa bem aos dados, por meio de seu diagnóstico.

Tabela 6 – Análise inferencial do modelo quadrático.

Parâmetro	**Valor estimado**	**Erro-padrão**	**Valor t**	**Valor-p**
β_0	43,7309	0,2072	211,09	$< 0,0001$
β_1	1,0176	0,0237	42,94	$< 0,0001$
β_2	$-0,0245$	0,0019	$-12,62$	$< 0,0001$

Fonte: elaborada pelo autor.

4.3.1 Análise de Diagnóstico

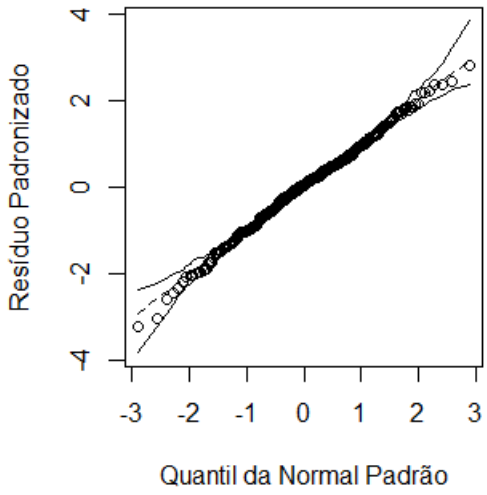
A Figura 11 apresenta o gráfico de quantis-quantis com envelopes simulados para os resíduos padronizados do modelo quadrático. Perceba que, de forma geral, os pontos ficam dentro dos envelopes simulados, mas em algumas regiões eles se afastam do comportamento esperado e até saem destes, como na região entre os valores -2 e -1 dos quantis da normal padrão. Este é um indício de que, na forma como está, o modelo quadrático viola a suposição de normalidade.

Pode-se visualizar o desempenho geral do modelo através do gráfico da variável regressora *versus* a explicativa com a curva de regressão. Tal gráfico pode ser visto na Figura 12. Note a curva não se ajusta bem às observações com os valores mais baixos da variável IMC.

O modelo estima a média em um valor maior que o real, o que provoca o surgimento de resíduos com valores elevados. Também se nota uma tendência de decrescimento no valor estimado para valores altos do IMC, o que não parece ser apoiado pelos dados.

A Figura 13 mostra os gráficos de influência, alavancagem e pontos aberrantes para o modelo quadrático. Note que neles se percebem algumas tendências. Por exemplo, valores mais baixos do IMC tendem a ser mais influentes. Os pontos que mais se destacam são o #123 e o #304. Este último também se destacou no modelo linear. Já o ponto #123 tem valor do IMC próximo ao máximo (56,5 kg/m^2) e é responsável pelo valor máximo da gordura corporal (58,56%). Há algumas observações com alavancagem elevada, mas não a ponto de se destacarem fortemente das outras. Não se podem detectar pontos aberrantes com clareza, e pode-se identificar pelo gráfico da função de variância que esta não é constante, violando mais uma suposição do modelo.

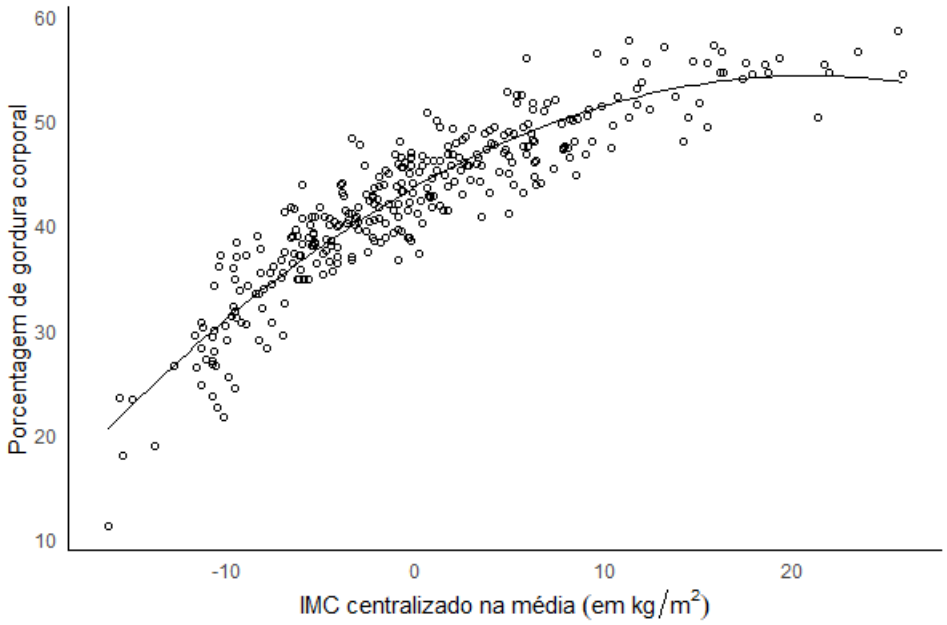
Figura 11 – Gráfico de quantis-quantis com envelopes simulados a 95% de confiança para o modelo quadrático.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

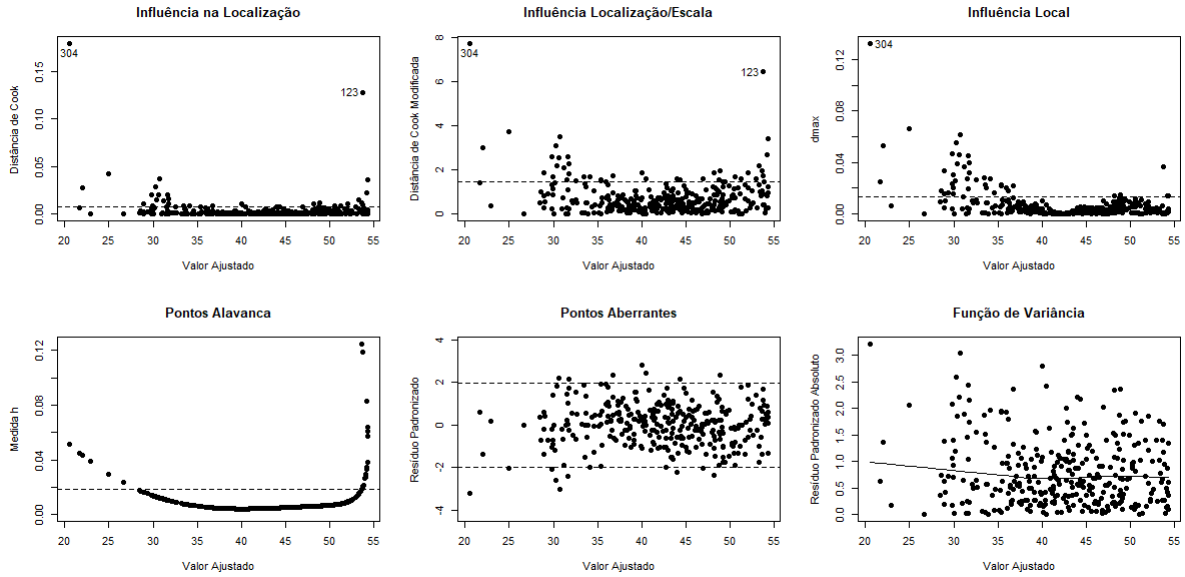
Estas observações tornam claro que o modelo pode apresentar problemas para se ajustar a observações com valores nas caudas das duas variáveis em consideração, o que fornece evidências contra a confiabilidade deste. Precisa-se realizar uma análise confirmatória para se verificar o efeito real de tais observações sobre o processo de estimação.

Figura 12 – Gráfico da gordura corporal *versus* IMC (em kg/m^2) centralizado com a curva de regressão do modelo quadrático.



Fonte: elaborada pelo autor.

Figura 13 – Gráficos de influência, alavancagem e pontos aberrantes para o modelo quadrático.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

A Tabela 7 mostra as estimativas dos parâmetros do modelo quadrático sem os pontos influentes e suas respectivas variações. Note que o intercepto é pouco sensível aos pontos influentes, apresentando variação de $0,1\%$ e $-0,1\%$ com a retirada das observações #123 e #324, respectivamente. $\hat{\beta}_1$, sofre alterações maiores, mas ainda modestas. $\hat{\beta}_2$, por outro lado, sofre mudanças consideráveis. As influências relativamente maiores apontadas aqui, além

da aparente falta de ajuste a valores nas caudas das variáveis, podem comprometer a eficácia do modelo. Logo, faremos bem em considerar alternativas ao modelo quadrático. O próximo que examinaremos é o modelo cúbico.

Tabela 7 – Estimativas dos parâmetros do modelo quadrático após se retirar os pontos influentes.

Modelo	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
Completo	$43,7309 \pm 0,2072$	$1,0176 \pm 0,0237$	$-0,0245 \pm 0,0019$
Sem #123	$43,7772 \pm 0,2084$ $0,10\%$	$1,0163 \pm 0,0236$ $-0,13\%$	$-0,0255 \pm 0,0020$ $-4,12\%$
Sem #304	$43,6873 \pm 0,2047$ $-0,10\%$	$1,0034 \pm 0,0238$ $-1,40\%$	$-0,0233 \pm 0,0019$ $4,81\%$
Sem #123 e #304	$43,7312 \pm 0,2061$ $0,008\%$	$1,0024 \pm 0,0237$ $-1,50\%$	$-0,0243 \pm 0,0020$ $0,89\%$

Fonte: elaborada pelo autor.

4.4 Modelo Cúbico

Dados os indícios de falta de ajuste no modelo quadrático, propomos agora um modelo cúbico para modelar a relação entre o IMC e a taxa de gordura corporal nas mulheres negras americanas. A forma funcional deste é

$$\texttt{pbfm} = \beta_0 + \beta_1 \texttt{bmi}^* + \beta_2 (\texttt{bmi}^*)^2 + \beta_3 (\texttt{bmi}^*)^3 + e_i, \quad i = 1, \ldots, 327,$$

em que $e_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \ \forall i$. O IMC foi centralizado na média para permitir a interpretação do intercepto.

A Tabela 8 mostra a análise inferencial do modelo cúbico. Assim como no modelo quadrático, todos os parâmetros são fortemente significativos. O intercepto estimado é $44,0055$. Logo, a taxa de gordura corporal esperada para as mulheres negras com IMC de $30,9358$ kg/m^2 é de 44,0055%. Tem-se que os efeitos linear, quadrático e cúbico do IMC estimados são, respectivamente, $0,9134$, $-0,0327$ e $0,0007$. Não obstante, para se examinar o modelo de maneira mais ampla, precisa-se verificar sua adequação aos dados.

Tabela 8 – Análise inferencial do modelo cúbico.

Parâmetro	Valor estimado	Erro-padrão	Valor t	Valor-p
β_0	44,0055	0,2130	206,572	$< 0,0001$
β_1	0,9134	0,0343	26,601	$< 0,0001$
β_2	$-0,0327$	0,0028	$-11,876$	$< 0,0001$
β_3	0,0007	0,0002	4,11	0,0001

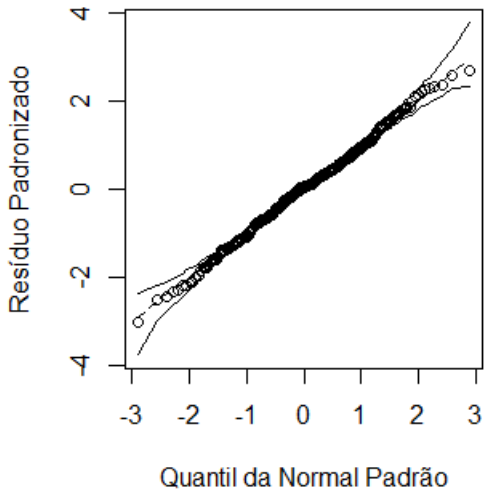
Fonte: elaborada pelo autor.

4.4.1 Análise de Diagnóstico

Assim como na Subseção 4.3.1, nesta subseção avaliaremos a adequação do modelo cúbico através de uma série de técnicas e medidas.

Primeiramente, avaliaremos a validade da suposição de normalidade por parte dos erros. Isto pode ser visto através da Figura 14. Neste gráfico, não há evidências diretas da quebra da suposição de normalidade no modelo cúbico. Isto é um indício de que o ajuste deste modelo é superior ao do modelo quadrático. A veracidade desta hipótese será avaliada através das outras técnicas de diagnóstico para o modelo cúbico.

Figura 14 – Gráfico de quantis-quantis com envelopes simulados a 95% de confiança para o modelo cúbico.



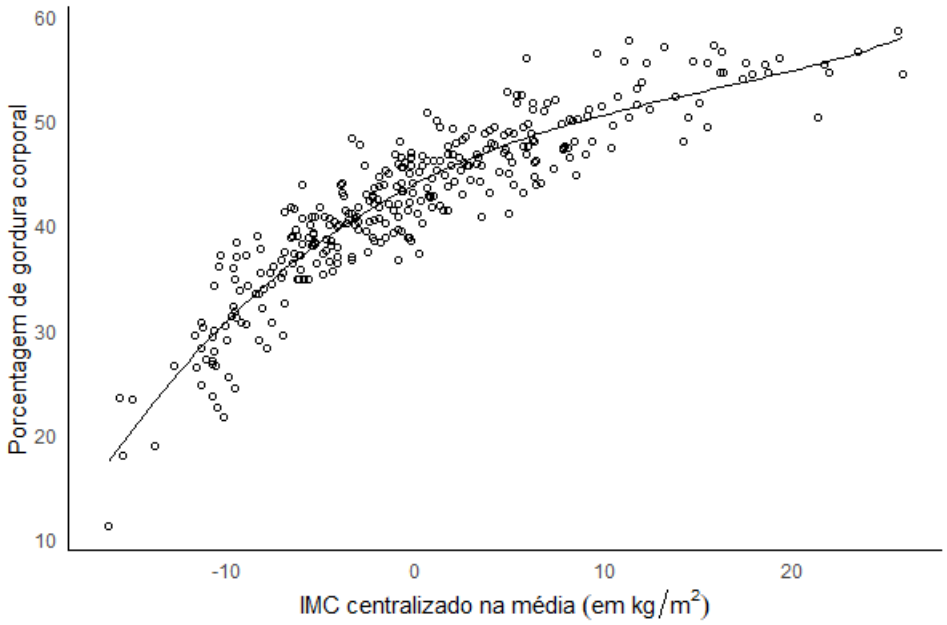
Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

A curva de regressão ajustada pelo modelo polinomial cúbico é mostrada na Figura 15. De forma geral, a curva obedece ao padrão estabelecido pelos pontos melhor que a curva

do modelo quadrático. Note que há uma mudança de inclinação para valores altos do IMC, o que pode ser um indício de leve sobreajuste.

Para avaliar os fatores observados anteriormente, será importante realizar uma análise de sensibilidade no modelo. Os gráficos que nos auxiliarão nesta análise estão na Figura 16. Nos gráficos de influência, há dois pontos que se destacam: o #304 e o #315. Ambos também se destacaram no modelo linear. Estes último ponto pode ser um dos responsáveis pelo aparente sobreajuste notado na Figura 15. Percebe-se também que os pontos do modelo cúbico apresentam alavancagem maior que os do modelo quadrático. Não há pontos aberrantes ou tendências evidentes que sugiram não-normalidade, mas ainda há indícios de variância não constante, embora mais fracos que os do modelo quadrático. Far-se-á agora uma análise confirmatória para investigar o efeito dos pontos influentes aqui identificados.

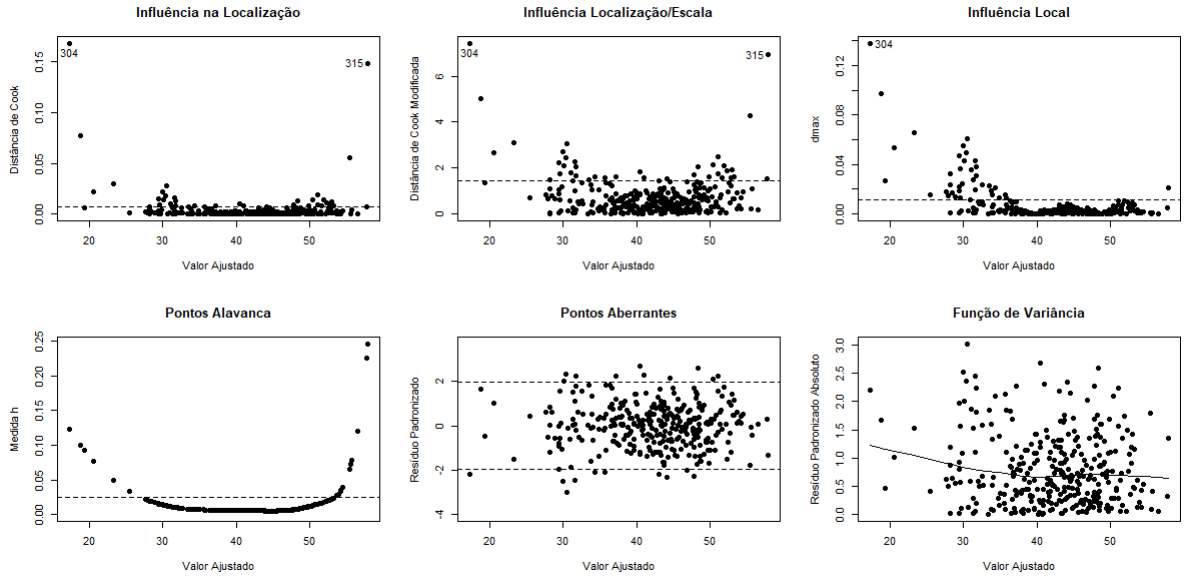
Figura 15 – Gráfico da porcentagem gordura corporal *versus* IMC (em kg/m^2) centralizado com a curva de regressão do modelo cúbico.



Fonte: elaborada pelo autor.

Pode-se encontrar um resumo do efeito da retirada dos pontos influentes do modelo cúbico na Tabela 9. Note que o parâmetro cujo estimador sofre a maior variação com a retirada dos pontos é β_3, seguido de β_2, β_1 e β_0. De forma geral, a magnitude das variações ocorridas nos estimadores com a retirada dos pontos influentes é maior no modelo cúbico do que no modelo quadrático, chegando a quase 15% para $\hat{\beta}_3$. Desta forma, embora pareça atender melhor às suposições, este modelo é mais sensível a variações nas observações, além de ter menor interpretabilidade e ser menos parcimonioso.

Figura 16 – Gráficos de influência, alavancagem e pontos aberrantes para o modelo cúbico.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

Tabela 9 – Estimativas dos parâmetros do modelo cúbico após se retirar os pontos influentes.

Modelo	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$
Completo	$44,0055 \pm 0,2130$	$0,9134 \pm 0,0343$	$-0,0327 \pm 0,0028$	$0,0007 \pm 0,0002$
Sem #304	43.9330 ± 0.2143 $-0,16\%$	0.9193 ± 0.0342 $0,65\%$	-0.0306 ± 0.0029 $-6,36\%$	$0,0006 \pm 0,0002$ $14,29\%$
Sem #315	$44,0010 \pm 0,2128$ $-0,01\%$	$0,9007 \pm 0,0356$ $-1,39\%$	$-0,0329 \pm 0,0028$ $0,63\%$	$0,0008 \pm 0,0002$ $-14,29\%$
Sem #304 e #315	$43,9317 \pm 0,2142$ $-0,17\%$	$0,9077 \pm 0,0355$ $-0,62\%$	$-0,0309 \pm 0,0029$ $-5,57\%$	$0,00068 \pm 0,0002$ $-2,89\%$

Fonte: elaborada pelo autor.

Os dois modelos polinomiais avaliados até agora deixam dúvidas sobre sua capacidade de modelar de forma confiável a relação entre as variáveis envolvidas. Já o modelo linear apresenta ajuste claramente insatisfatório. Partir para um polinômio usual de grau maior pode tornar o modelo ainda mais sensível a pontos influentes e ao sobreajuste. Agora, ajustaremos um modelo polinomial fracionário aos dados para avaliar sua eficácia.

4.5 Modelo Polinomial Fracionário

Para selecionar um modelo polinomial fracionário, deve-se utilizar o algoritmo MFP. Para modelar a taxa de gordura corporal em função do IMC, usaremos um grau máximo igual a

2. Desta forma, procuraremos determinar o melhor modelo dentre os que atendem à forma

$$\texttt{pbfm} = \beta_0 + \beta_1 \texttt{bmi}^{p_1} + \beta_2 \texttt{bmi}^{p_2} + e_i, \quad i = 1, \ldots, 327,$$

em que $e_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\forall i$ e $(p_1, p_2) \in S^2$, com $S = \{-2, -1, -1/2, 0, 1/2, 1, 2, 3\}$, por meio de um MFP$(0, 05)$. Após a execução do algoritmo, que seleciona o melhor modelo por meio de testes F, obtemos o seguinte modelo final:

$$\texttt{pbfm} = \beta_0 + \beta_1 \texttt{bmi}^{-1} + e_i, \quad i = 1, \ldots, 327.$$

Note que o modelo selecionado obedece ao nosso apontamento inicial de que a relação entre a taxa de gordura corporal e o IMC é não-linear e tem assíntota. Precisa-se agora avaliar a interpretação e a adequação deste.

A interpretação dos parâmetros pode ser realizada através da tabela de análise inferencial, apresentada na Tabela 10. O efeito inverso do IMC sobre a taxa de gordura corporal das mulheres negras americanas é $-833,72$. O intercepto é 70,91, sem uma interpretação direta, pois não é possível centralizar o IMC na média. Note que, embora ainda de interpretação indireta, este modelo é mais simples e mais parcimonioso que os dois últimos – com apenas dois parâmetros, contra os três do modelo quadrático e os quatro do modelo cúbico. Precisa-se agora examinar se o modelo satisfaz as suposições e sua sensibilidade.

Tabela 10 – Análise inferencial do modelo polinomial fracionário.

Parâmetro	**Valor estimado**	**Erro-padrão**	**Valor t**	**Valor-p**
β_0	$70,9149$	$0,6579$	$107,80$	$< 0,0001$
β_1	$-833,5154$	$18,5003$	$-45,05$	$< 0,0001$

Fonte: elaborada pelo autor.

4.5.1 Análise de Diagnóstico

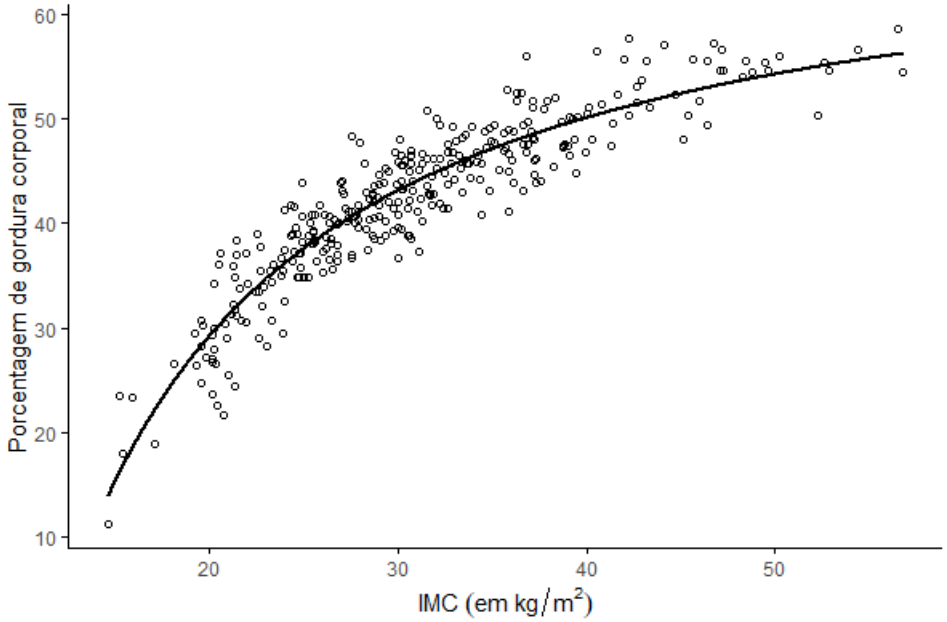
Podem-se obter informações sobre a qualidade geral do ajuste do modelo polinomial fracionário por meio da Figura 17. Note que, neste modelo, a curva não apresenta sinais claros de sobreajuste ou de falta de ajuste. Conclui-se que o modelo polinomial fracionário consegue, até certo ponto, capturar a relação entre a gordura corporal e o IMC em mulheres negras.

Agora, avaliaremos se a suposição de normalidade por parte dos erros é atendida. O gráfico de quantis-quantis dos resíduos do modelo polinomial fracionário é apresentado na Figura 18. Percebe-se que não há sinais claros de desvio da normalidade, o que se configura

como evidência de que não se pode rejeitar a suposição de que a normalidade é atendida pelos erros.

Para se obter mais detalhes sobre a qualidade do ajustem do FPM, deve-se considerar os outros fatores pertinentes ao diagnóstico. Alguns destes são mostrados na Figura 19. Note que, nos três gráficos de influência, a observação #188 se destaca. Esta tem IMC de 15,2 – um dos valores mais baixos registrados – e taxa de gordura corporal de 23,5, também próximo ao mínimo. Comparando o gráfico de pontos alavanca do modelo polinomial fracionário com os dos dois modelo anteriores, constata-se que, de forma geral, as observações no modelo polinomial fracionário apresentam alavancagem menor. Nestes gráficos não se encontram pontos aberrantes ou evidências de não-normalidade ou de variância não constante.

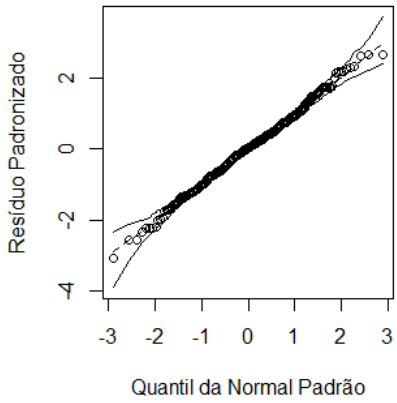
Figura 17 – Gráfico da porcentagem gordura corporal *versus* IMC (em kg/m^2) com a curva de regressão do modelo polinomial fracionário.



Fonte: elaborada pelo autor.

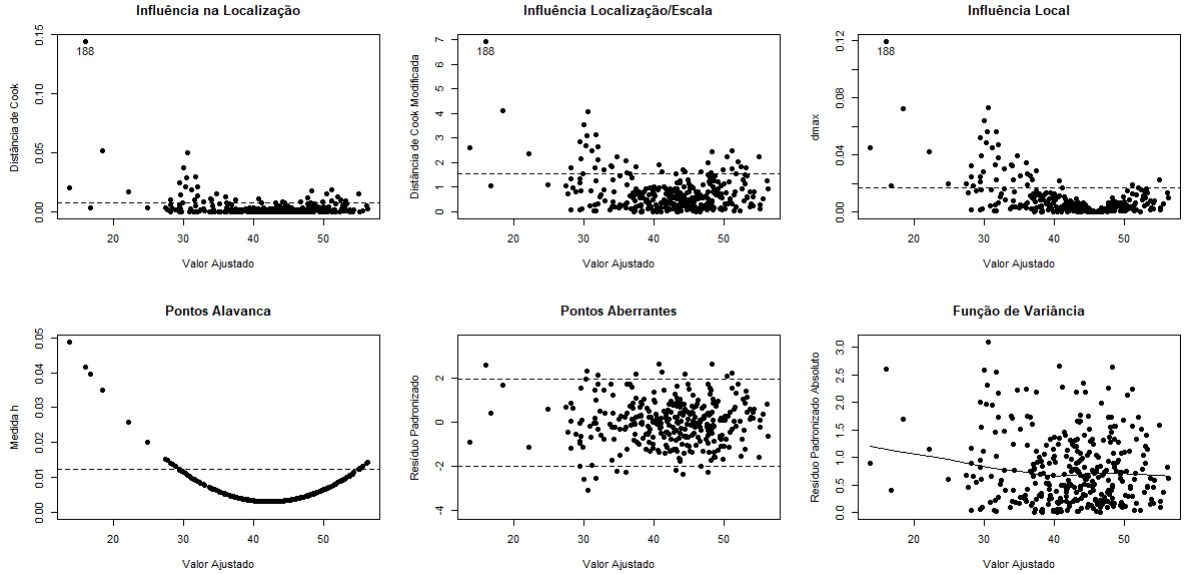
Para examinar o efeito da observação #188 no processo de estimação, pode-se consultar a Tabela 11. A retirada desta causa uma variação de $0,43\%$ no estimador do intercepto e de $-1,15\%$ no estimador de β_1. Embora seja claramente a observação mais influente do modelo polinomial fracionário, seu grau de influência é modesto, especialmente se comparado aos graus de influência dos pontos influentes dos modelos quadrático e cúbico.

Figura 18 – Gráfico de quantis-quantis com envelopes simulados a 95% de confiança para o modelo polinomial fracionário.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

Figura 19 – Gráficos de influência, alavancagem e pontos aberrantes para o modelo polinomial fracionário.



Fonte: elaborada pelo autor, com base nas funções propostas por Paula (2012).

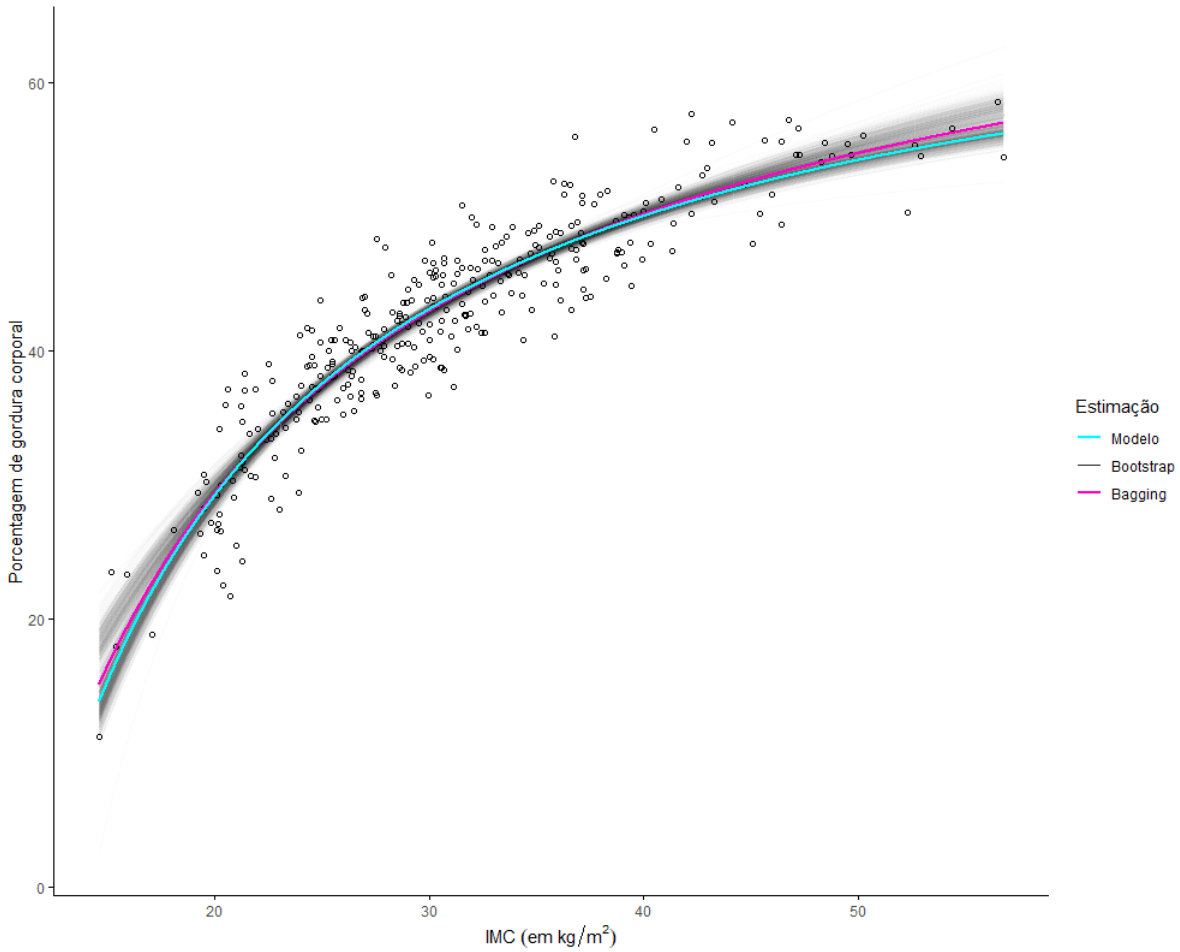
Tabela 11 – Estimativas dos parâmetros do modelo polinomial fracionário após se retirar o ponto influente.

Modelo	$\hat{\beta}_0$	$\hat{\beta}_1$
Completo	$70,9149 \pm 0,6579$	$-833,5154 \pm 18,5003$
Sem #188	$71,2207 \pm 0,6627$ $0,43\%$	$-843,0807 \pm 18,7055$ $-1,15\%$

Fonte: elaborada pelo autor.

Há ainda outros fatores que se devem considerar em um FPM, que são a estabilidade de sua seleção e uma quantidade menor de parâmetros. Não se pode tomar conclusões muito confiáveis se o algoritmo MFP de seleção de modelos tiver comportamento instável. Após se executarem 1000 réplicas de *bootstrap*, com 150 observações cada, e se estimarem por MFP as funções para cada réplica, além do estimador por *bagging* global, obteve-se uma fração do valor de D^2 sobre a variação total das funções de *bootstrap* de $0,2\%$. Este é um forte indício de que o procedimento é estável. A Figura 20 mostra o gráfico da função estimada, junto com as funções *bootstrap* e a curva estimada por *bagging*. Note que a função selecionada pelo MFP para o IMC está próxima à estimada por *bagging*, sendo uma evidência de estabilidade.

Figura 20 – Gráfico do modelo polinomial fracionário e suas curvas estimadas por *bootstrap* e *bagging*.



Fonte: elaborada pelo autor.

4.6 Comparação dos Modelos

Em posse dos ajustes dos modelos, sua interpretação e de uma análise de sua adequação, pode-se tomar uma decisão bem acertada sobre qual melhor atende às nossas necessidades. Além dos fatores já considerados, pode-se também tomar outros, como o AIC (critério de informação Akaike), BIC (critério de informação bayesiano), R^2, e R^2 ajustado, que são mostrados na Tabela 12. O FPM apresenta os menores valores do AIC e BIC e os maiores valores do R^2. Note que, para os modelos linear e FPM, foi usado o R^2, enquanto para os modelos quadrático e cúbico foi usado o R^2 ajustado. Logo, todos os critérios indicam que o modelo polinomial fracionário é o modelo dentre os considerados que melhor explica a variação da taxa de gordura corporal em função do IMC nas mulheres negras americanas.

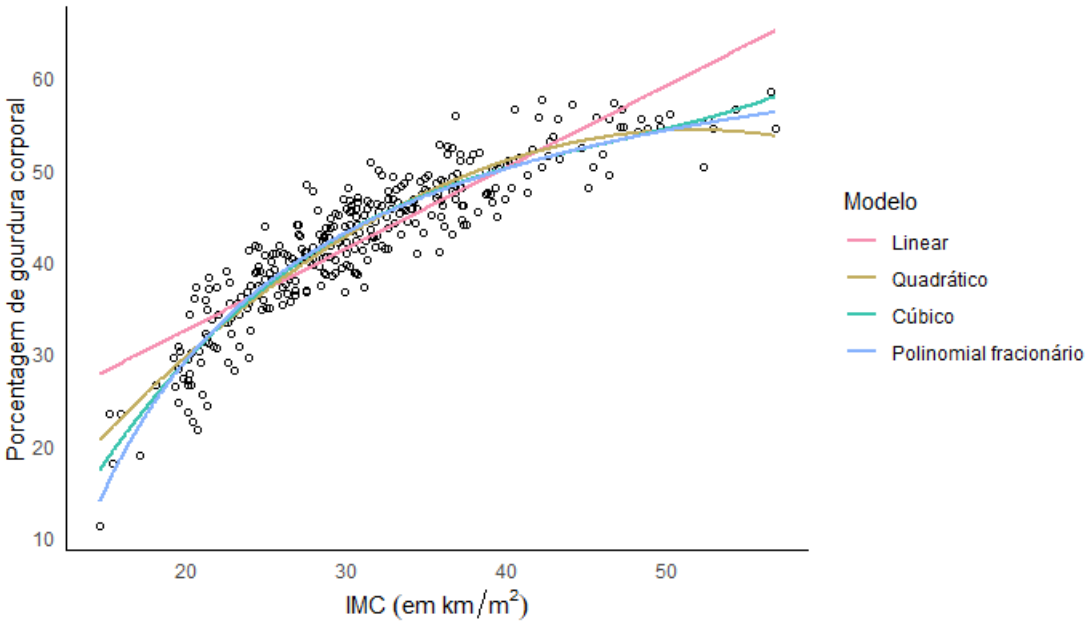
Tabela 12 – Medidas dos ajustes dos modelos propostos.

Medida	**Modelo linear**	**Modelo quadrático**	**Modelo cúbico**	**Modelo FP**
AIC	1788,062	1659,368	1644,708	1639,023
BIC	1799,432	1674,528	1663,658	1650,393
R^2	0,7823	0,8531	0,8600	0,862

Fonte: elaborada pelo autor.

Ao longo deste capítulo, foram comparados um modelo linear, modelos polinomiais de graus diferentes e um modelo polinomial fracionário. A Figura 21 mostra um comparativo de todos os modelos ajustados neste capítulo. Vê-se que o modelo linear é insatisfatório em seu ajuste às caudas. O modelo quadrático deixa a desejar na qualidade do ajuste e peca na robustez, sendo mais sensível a pontos influentes. Ademais, os resíduos deste modelo parecem não se adequar muito bem às suposições de normalidade e de variância constante. O modelo cúbico apresenta ajuste melhor aos dados e satisfaz melhor as suposições, mas tem indícios de sobreajuste. Este, porém, é ainda mais suscetível a pontos influentes, podendo assim ter sua qualidade questionada. O FPM apresenta ajuste mais satisfatório que os demais, evitando tanto a falta de ajuste quanto o sobreajuste. O modelo satisfaz as suposições de normalidade e de variância constante e é claramente mais robusto que os modelos polinomiais usuais propostos.

Figura 21 – Gráfico da porcentagem da gordura corporal *versus* IMC (em kg/m^2) com comparativo dos modelos considerados.



Fonte: elaborada pelo autor.

O último critério a se considerar ao se decidir entre diferentes modelos é a parcimônia. Esta é referente à capacidade de um modelo explicar os dados de maneira simples. Os modelos considerados aqui, especialmente os polinomiais usuais, apresentam algumas barreiras à interpretabilidade. O FPM foi capaz de modelar a variável resposta de maneira melhor e mais parcimoniosa – com apenas dois parâmetros, contra os três parâmetros do modelo quadrático e os quatro do modelo cúbico. Os valores de AIC e BIC também evidenciam o fato de que o FPM considerado é mais parcimonioso.

Portanto, vê-se que os modelos polinomiais fracionários podem, em algumas situações, oferecer um ganho substancial na modelagem e na parcimônia com relação os polinômios tradicionais. Em termos de interpretabilidade, os FPMs são ligeiramente superiores a estes últimos, dada sua tendência a ter uma maior parcimônia. Por vezes, como no caso examinado neste capítulo, os FPMs proporcionam um ajuste melhor com uma abordagem mais simples, proporcionando uma boa relação de custo-benefício.

5 CONCLUSÃO

O objetivo deste trabalho foi introduzir, de forma sucinta, os modelos polinomiais fracionários. Para isto, foi inicialmente realizada uma breve introdução ao conceito de regressão linear. Após algumas discussões sobre os modelos lineares, polinomiais, *splines* e suas limitações, apresentou-se o conceito de polinômios fracionários e, em seguida, de FPMs propriamente ditos. Também foram abordadas técnicas de seleção de modelos dentro desta classe e algumas ferramentas de diagnóstico. Em seguida, fez-se uma introdução aos aspectos computacionais pertinentes à classe de modelos.

Para se avaliar de forma prática a usabilidade dos modelos polinomiais fracionários, foi analisado um conjunto de dados referente à gordura corporal de mulheres negras dos Estados Unidos em função de seu IMC. Tal modelagem é especialmente relevante dado que os meios tradicionais de se medir a taxa de gordura corporal é complexa e onerosa. De início, havia fortes evidências de que um modelo linear não seria suficiente para se modelar a relação entre as variáveis. Os modelos polinomiais propostos sofriam, em variados graus, com falta de adequação às suposições e com a presença de pontos influentes. O modelo quadrático deixou a desejar na qualidade do ajuste, enquanto o modelo cúbico apresentou leves indícios de sobreajuste. Por outro lado, no conjunto de dados analisado, o modelo polinomial fracionário apresentou ajuste mais satisfatório, mais robusto e mais parcimonioso, além de ter tido um procedimento de seleção que pode ser considerado estável. Portanto, dentro do problema analisado, a classe de modelos polinomiais fracionário foi a que obteve o desempenho mais adequado dentre os modelos propostos. Como pesquisa futura, pode-se avaliar o desempenho de um modelo beta aos dados, já que eles estão restritos ao intervalo $(0, 100)$.

Os FPMs constituem uma classe recente e crescentemente usada de modelos, oferecendo consideráveis ganhos ao pesquisador. Ainda assim, há uma quantidade reduzida de referências na área de modelos polinomiais fracionários. Estes foram propostos inicialmente por Royston e Altman (1994), e a maioria das referências que há são dos pioneiros do modelo. Este trabalho busca motivar a pesquisa na área dos polinômios fracionários, que são uma abordagem simples que se destaca por sua parcimônia. Pode-se dizer que, em termos de flexibilidade, estes ficam entre os MLGs tradicionais e os MAGs, sendo totalmente paramétricos.

Destaca-se que o métodos de diagnóstico são imprescindíveis para se determinar a qualidade do ajuste dos FPMs. Deve-se dar especial atenção a covariáveis com valores muito baixos ou muito elevados. Ademais, é importante que se avalie a estabilidade do processo de

seleção de modelos, bem como a presença de multicolinearidade na matriz de especificação deste.

Há diversas áreas em que se pode pesquisar e fortalecer a prática dos modelos polinomiais fracionários. Primeiramente, nota-se uma carência de métodos flexíveis para a aplicação de FPMs em *softwares* de código aberto. Pode-se explorar a aplicação de polinômios fracionários junto ao modelos de análise de sobrevivência, aos MAGs, ou ainda junto aos GAMLLS, o que poderia resultar em uma classe de modelos destacada pela flexibilidade e pela parcimônia. Como já citado, uma proposta interessante seria juntar polinômios fracionários ao modelo beta para ajustar os dados apresentados no Capítulo 4. Outra possibilidade seria usar o algoritmo PIPE para selecionar as potências, considerando um conjunto S mais amplo. Ademais, a técnica de *bagging*, usada nos métodos empíricos de diagnóstico para os FPMs, representa uma abordagem interessante no contexto da modelagem robusta.

REFERÊNCIAS

ALHAMZAWI, R. Brq: An r package for bayesian quantile regression. **Working Paper**, 2018.

AMBLER, G.; BENNER, A. **mfp: Multivariable Fractional Polynomials**. [*S. l.*], 2022. R package version 1.5.2.2. Disponível em: https://CRAN.R-project.org/package=mfp.

AMBLER, G.; ROYSTON, P. Fractional polynomial model selection procedures: investigation of type i error rate. **Journal of Statistical Computation and Simulation**, Taylor & Francis, v. 69, n. 1, p. 89–108, 2001. Disponível em: https://doi.org/10.1080/00949650108812083. Acesso em: 26 de julho de 2022.

AREGAY, M.; SHKEDY, Z.; MOLENBERGHS, G.; DAVID, M.-P.; TIBALDI, F. Nonlinear fractional polynomials for estimating long-term persistence of induced anti-hpv antibodies: A hierarchical bayesian approach. **Statistics in Biopharmaceutical Research**, Taylor & Francis, v. 6, n. 3, p. 199–212, 2014. Disponível em: https://doi.org/10.1080/19466315.2014.911201. Acesso em: 22 de maio de 2022.

ATKINSON, A. C. Two graphical displays for outlying and influential observations in regression. **Biometrika**, [Oxford University Press, Biometrika Trust], v. 68, n. 1, p. 13–20, 1981. Disponível em: http://www.jstor.org/stable/2335801.

ATTALI, D.; BAKER, C. **ggExtra: Add Marginal Histograms to 'ggplot2', and More 'ggplot2' Enhancements**. [*S. l.*], 2022. R package version 0.10.0. Disponível em: https://CRAN.R-project.org/package=ggExtra.

BELSLEY, D. A.; KUH, E.; WELSCH, R. E. **Regression Diagnostics: Identifying Influential Data and Sources of Collinearity**. 1. ed. Hoboken: Wiley, 1980.

BIDDINGER, K. J.; EMDIN, C. A.; HAAS, M. E.; WANG, M.; HINDY, G.; ELLINOR, P. T.; KATHIRESAN, S.; KHERA, A. V.; ARAGAM, K. G. Association of Habitual Alcohol Intake With Risk of Cardiovascular Disease. **JAMA Network Open**, v. 5, n. 3, p. e223849–e223849, 03 2022. ISSN 2574-3805. Disponível em: https://doi.org/10.1001/jamanetworkopen.2022.3849.

BINDER, H.; SAUERBREI, W.; ROYSTON, P. Comparison between splines and fractional polynomials for multivariable model building with continuous covariates: a simulation study with continuous response. **Statistics in Medicine**, v. 32, n. 13, p. 2262–2277, 2013. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.5639. Acesso em: 31 de julho de 2022.

BOX, G. E.; TIDWELL, P. W. Transformation of the independent variables. **Technometrics**, Taylor & Francis, v. 4, n. 4, p. 531–550, 1962. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/00401706.1962.10490038.

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, p. 123 – 140, 1996. Disponível em: https://doi.org/10.1007/BF00058655. Acesso em: 15 de outubro de 2022.

CASTELNUOVO, A. D.; COSTANZO, S.; BAGNARDI, V.; DONATI, M. B.; IACOVIELLO, L.; GAETANO, G. de. Alcohol Dosing and Total Mortality in Men and Women: An Updated Meta-analysis of 34 Prospective Studies. **Archives of Internal Medicine**, v. 166, n. 22, p. 2437–2445, 12 2006. ISSN 0003-9926. Disponível em: https://doi.org/10.1001/archinte.166.22.2437.

CLEVELAND, W. S. Robust locally weighted regression and smoothing scatterplots. **Journal of the American Statistical Association**, Taylor & Francis, v. 74, n. 368, p. 829–836, 1979. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/01621459.1979.10481038. Acesso em: 18 de julho de 2022.

COOK, R. D. Detection of influential observation in linear regression. **Technometrics**, [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], v. 19, n. 1, p. 15–18, 1977. Disponível em: http://www.jstor.org/stable/1268249. Acesso em: 20 de agosto de 2022.

COOK, R. D. Influence assessment. **Journal of Applied Statistics**, Taylor & Francis, v. 14, n. 2, p. 117–131, 1986. Disponível em: https://doi.org/10.1080/02664768700000016.

COOK, R. D.; WEISBERG, S. **Residuals and Influence in Regression**. 1. ed. Nova Iorque: Chapman & Hall, 1982.

DAVIDSON, R.; MacKinnon, J. G. **Econometric Theory and Methods**. 1. ed. Nova Iorque: Oxford University Press, 2004.

DRAPER, N. R.; SMITH, H. **Applied Regression Analysis**. 3. ed. Nova Iorque: John Wiley & Sons, 1998.

GARCIA, E. P. **Uso de Polinômios Fracionários nos Modelos Mistos**. 2019. Tese (Doutorado em Biometria) – Universidade Estadual Paulista "Júlio Mesquita Filho", Botucatu, São Paulo. Disponível em: https://repositorio.unesp.br/handle/11449/181646.

GILMOUR, S. G.; TRINCA, L. A. Fractional polynomial response surface models. **Journal of Agricultural, Biological, and Environmental Statistics**, [International Biometric Society, Springer], v. 10, n. 1, p. 50–60, 2005. Disponível em: http://www.jstor.org/stable/27595542. Acesso em: 15 de julho de 2022.

HASTIE, T. J.; TIBSHIRANI, R. J. **Generalized Additive Models**. Londres: Chapman & Hall, 1990.

HOAGLIN, D. C.; WELSCH, R. E. The hat matrix in regression and anova. **The American Statistician**, Taylor & Francis, v. 32, n. 1, p. 17–22, 1978. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/00031305.1978.10479237. Acesso em: 9 de outubro de 2022.

HOFFMANN, R. **Análise de Regressão: Uma Introdução à Econometria**. São Paulo: Portal de Livros Abertos da USP, 2016. Disponível em: https://www.esalq.usp.br/biblioteca/sites/default/files/Analise_Regress%C3%A3o.pdf. Acesso em: 17 de julho de 2022.

ISAACS, D.; ALTMAN, D. G.; TIDMARSH, C. E.; VALMAN, H. B.; WEBSTER, A. D. Serum immunoglobulin concentrations in preschool children measured by laser nephelometry: reference ranges for igg, iga, igm. **Journal of Clinical Pathology**, BMJ Publishing Group, v. 36, n. 10, p. 1193–1196, 1983. ISSN 0021-9746. Disponível em: https://jcp.bmj.com/content/36/10/1193.

KUTNER, M. C.; NACHTSHEIM, C. J.; NETER, J.; LI, W. **Applied Linear Statistical Models**. 5. ed. Nova Iorque: McGraw-Hill, 2005.

LUKE, A.; DURAZO-ARVIZU, R.; ROTIMI, C.; PREWITT, T. E.; FORRESTER, T.; WILKS, R.; OGUNBIYI, O. J.; SCHOELLER, D. A.; MCGEE, D.; COOPER, R. S. Relation between body mass index and body fat in black population samples from nigeria, jamaica, and the united states. **American Journal of Epidemiology**, v. 145, n. 7, p. 620–628, 04 1997. Disponível em: https://doi.org/10.1093/oxfordjournals.aje.a009159. Acesso em: 30 de outubro de 2022.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. **Introduction to Linear Regression Analysis**. 6. ed. Nova Iorque: Wiley, 2021.

MOOD, A. M.; GRAYBILL, F. A.; BOES, D. C. **Introduction to The Theory of Statistics**. Nova Iorque: McGraw-Hill, 1974.

OUNSTED, M.; MOAR, V.; SCOTT, A. Growth in the first four years: Iv. correlations with parental measures in small-for-dates and large-for-dates babies. **Early Human Development**, v. 7, n. 4, p. 357–366, 1982. ISSN 0378-3782. Disponível em: https://www.sciencedirect.com/science/article/pii/0378378282900378. Acesso em: 22 de maio de 2022.

PAULA, G. A. **Modelos de Regressão com Apoio Computacional**. São Paulo: IME/USP, 2012.

R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2022. Disponível em: https://www.R-project.org/. Acesso em: 30 de outubro de 2022.

ROYSTON, P. Model selection for univariable fractional polynomials. **The Stata Journal**, v. 17, n. 3, p. 619–629, 2017. Disponível em: https://journals.sagepub.com/doi/pdf/10.1177/1536867X1701700305.

ROYSTON, P.; ALTMAN, D. Using fractional polynomials to model curved regression relationships. **Stata Technical Bulletin**, v. 4, 02 1995.

ROYSTON, P.; ALTMAN, D. G. Regression using fractional polynomials of continuous covariates: Parsimonious parametric modelling. **Journal of the Royal Statistical Society: Series C (Applied Statistics)**, v. 43, n. 3, p. 429–453, 1994. Disponível em: https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2986270. Acesso em: 22 de maio de 2022.

ROYSTON, P.; ALTMAN, D. G. Approximating statistical functions by using fractional polynomial regression. **Journal of the Royal Statistical Society: Series D (The Statistician)**, v. 46, n. 3, p. 411–422, 1997. Disponível em: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9884.00093. Acesso em: 11 de dezembro de 2022.

ROYSTON, P.; SAUERBREI, W. Stability of multivariable fractional polynomial models with selection of variables and transformations: a bootstrap investigation. **Statistics in Medicine**, v. 22, n. 4, p. 639–659, 2003. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.1310. Acesso em: 15 de outubro de 2022.

ROYSTON, P.; SAUERBREI, W. A new approach to modelling interactions between treatment and continuous covariates in clinical trials by using fractional polynomials. **Statistics in Medicine**, v. 23, n. 16, p. 2509–2525, 2004. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.1815. Acesso em: 31 de julho de 2022.

ROYSTON, P.; SAUERBREI, W. **Multivariable Model-Building: A Pragmatic Approach to Regression Anaylsis Based on Fractional Polynomials for Modelling Continuous Variables**. Chichester, Reino Unido: John Wiley & Sons, 2008.

Sabané Bové, D.; HELD, L. Bayesian fractional polynomials. **Statistics and Computing**, v. 21, n. 3, p. 309–324, 2011. Disponível em: https://doi.org/10.1007/s11222-010-9170-7. Acesso em: 20 de dezembro de 2022.

SAUERBREI, W.; ROYSTON, P. Building multivariable prognostic and diagnostic models: Transformation of the predictors by using fractional polynomials. **Journal of the Royal Statistical Society. Series A (Statistics in Society)**, [Wiley, Royal Statistical Society], v. 162, n. 1, p. 71–94, 1999. Disponível em: http://www.jstor.org/stable/2680468. Acesso em: 30 de julho de 2022.

SAUERBREI, W.; ROYSTON, P.; LOOK, M. A new proposal for multivariable modelling of time-varying effects in survival data based on fractional polynomial time-transformation. **Biometrical Journal**, v. 49, n. 3, p. 453–473, 2007. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.200610328. Acesso em: 15 de julho de 2022.

SAUERBREI, W.; SCHUMACHER, M. A bootstrap resampling procedure for model building: Application to the cox regression model. **Statistics in Medicine**, John Wiley & Sons, v. 11, n. 16, p. 2093–2109, 1992. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780111607. Acesso em: 12 de outubro de 2022.

SILKE, B.; KELLETT, J.; ROONEY, T.; BENNETT, K.; ORIORDAN, D. An improved medical admissions risk system using multivariable fractional polynomial logistic regression modelling. **QJM: An International Journal of Medicine**, v. 103, n. 1, p. 23–32, 10 2009. Disponível em: https://doi.org/10.1093/qjmed/hcp149. Acesso em: 15 de julho de 2022.

STATA. **Fractional Polynomials**. 2022. Disponível em: https://www.stata.com/features/overview/fractional-polynomials/. Acesso em: 30 de outubro de 2022.

THERNEAU, T. M.; GRAMBSCH, P. M. **Modeling Survival Data: Extending the Cox Model**. New York: Springer, 2000. ISBN 0-387-98784-3.

WICKHAM, H. **ggplot2: Elegant Graphics for Data Analysis**. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. Disponível em: https://ggplot2.tidyverse.org.

WICKHAM, H. **stringr: Simple, Consistent Wrappers for Common String Operations**. [*S. l.*], 2022. R package version 1.4.1. Disponível em: https://CRAN.R-project.org/package=stringr.

ZHANG, M.; MICHOS, E. D.; WANG, G.; WANG, X.; MUELLER, N. T. Associations of Cord Blood Vitamin D and Preeclampsia With Offspring Blood Pressure in Childhood and Adolescence. **JAMA Network Open**, v. 3, n. 10, p. e2019046–e2019046, 10 2020. ISSN 2574-3805. Disponível em: https://doi.org/10.1001/jamanetworkopen.2020.19046. Acesso em: 22 de maio de 2022.

ZHOU, D.; PAN, E.; ZHANG, Y. Fractional polynomial function in stochastic response surface method for reliability analysis. **Journal of Mechanical Science and Technology**, v. 35, p. 121–131, 2021. Disponível em: https://doi.org/10.1007/s12206-020-1211-3. Acesso em: 15 de julho de 2022.

APÊNDICE A – FUNÇÃO PLOT_MFP

```
plot_fp <- function(mfp.object, data, y, x, xlab=NULL, ylab=NULL){
    variables <- rownames(mfp.object$powers)
    powers <- mfp.object$powers

    # Faz uma matriz de coeficientes
    coefficients <- c()
    coefficients <- append(coefficients, mfp.object$coefficients['Intercept'])
    coefficients <- append(coefficients, NA)
    for(variable in variables){
        parameters_for_variable <- (
            !is.na(str_extract(names(mfp.object$coefficients), variable))
        )
        n_parameters <- sum(parameters_for_variable)
        if(n_parameters == 0){
            parameter1 <- NA
            parameter2 <- NA
        } else if(n_parameters == 1){
            parameter1 <- mfp.object$coefficients[parameters_for_variable]
            parameter2 <- NA
        } else{
            parameter1 <- mfp.object$coefficients[parameters_for_variable][1]
            parameter2 <- mfp.object$coefficients[parameters_for_variable][2]
        }
        coefficients <- append(coefficients, parameter1)
        coefficients <- append(coefficients, parameter2)
    }
    coefficients <- matrix(coefficients,
                           nrow=(length(variables) + 1),
                           ncol=2,
                           byrow=TRUE)
    rownames(coefficients) <- append('Intercept', variables)
    colnames(coefficients) <- c('power1', 'power2')

    # Faz a função de cada variável
    functions <- c()
    fractional_polynomial <- function(power1,
                                      power2,
```

```
                                coefficient1,
                                coefficient2,
                                intercept){
force(power1)
force(power2)
force(coefficient1)
force(coefficient2)
force(intercept)

# Determina as funções
if(is.na(power1) & is.na(power2)){ # Quando a variável não é escolhida
    function(x) return(0)
} else if(is.na(power2)){ # Quando só tem p1
    if(power1 == 0){ # Quando p1 é 0
        function(x){return(intercept + coefficient1 * log(x))}
    } else{ # Quando p1 é diferente de 0
        function(x){return(intercept + coefficient1 * x^power1)}
    }
} else{ # Quando tem p1 e p2
    if(power1 == power2){ # Quando p1 e p2 são iguais
        if(power1 == 0){ # Quando ambos são 0
            function(x){return((intercept
                                + coefficient1 * log(x)
                                + coefficient2 * log(x)^2))}
        } else{ # Quando ambos não são 0
            function(x){return(intercept
                               + coefficient1 * x^power1
                               + coefficient2*(x^power1)*log(x))}
        }
    } else{ # Quando p1 e p2 são diferentes
        if(power1 == 0){ # Quando p1 é 0
            function(x){(intercept
                         + coefficient1 * log(x)
                         + coefficient2 * x^power1)}
        } else if(power2 == 0){ # Quando p2 é 0
            function(x){(intercept
                         + coefficient1 * log(x)
                         + coefficient2 * x^power2)}
        } else{ # Quando nem p1 nem p2 é 0
```

```
                function(x){(intercept
                            + coefficient1 * x^power1
                            + coefficient2*x^power2)}
            }
        }
    }
}
for(variable in variables){
    power1 <- powers[variable, 'power1']
    power2 <- powers[variable, 'power2']
    coefficient1 <- coefficients[variable, 'power1']
    coefficient2 <- coefficients[variable, 'power2']
    intercept <- coefficients['Intercept', 1]

    fractional_polynomial_variable <- fractional_polynomial(
        power1,
        power2,
        coefficient1,
        coefficient2,
        intercept
    )
    functions <- append(functions, fractional_polynomial_variable)
}
names(functions) <- variables

if(is.null(xlab)){
    xlab <- x
}
if(is.null(ylab)){
    ylab <- y
}

f <- functions[[x]]
# Plota
g <- ggplot(data=data, aes(x=eval(parse(text=x)), y=eval(parse(text=y)))) +
    geom_point(shape=1) +
    stat_function(fun=f, size=1, n=10000) +
    theme_minimal() +
    removeGridX() +
```

```
        removeGridY() +
        ylab(ylab) +
        xlab(xlab) +
        theme(axis.line = element_line(colour = "black"),
              axis.ticks = element_line(colour = "black"))
    return(g)
}
```

APÊNDICE B – FUNÇÃO BAGGING_DIAGNOSTICS

```
bagging_diagnostics <- function(mfp.object,
                                formula,
                                data,
                                bootstrap_size,
                                seed,
                                rescale=TRUE,
                                alpha=0.05,
                                select=0.05,
                                B=1000){
    variables <- rownames(mfp.object$powers)
    transformations = array(NA, c(B, 2, length(variables)),
                            dimnames=list(1:B, c('power1', 'power2'), variables))
    for(b in 1:B){
        set.seed(seed + b - 1)
        bootstrapped_data <- data[sample(1:nrow(data),
                                         size=bootstrap_size,
                                         replace=TRUE), ]
        # bootstrapped_data <- data
        rownames(bootstrapped_data) <- NULL
        bootstrapped_mfp <- mfp(
            formula=formula,
            data=bootstrapped_data,
            rescale=rescale,
            alpha=alpha,
            select=select
        )
        for(column in 1:ncol(bootstrapped_mfp$powers)){
            for(variable in variables){
                transformations[b,
                                column,
                                variable] <- bootstrapped_mfp$powers[variable,
                                                                     column]
            }
        }
    }

    # Calcula as funções bootstrap
```

```
functions <- c()
fractional_polynomial <- function(power1, power2, y){
    force(power1)
    force(power2)
    force(data)
    force(y)
    if(is.na(power1) & is.na(power2)){ # Quando a variável não é escolhida
        function(x) return(0)
    } else if(is.na(power2)){ # Quando só tem p1
        if(power1 == 0){ # Quando p1 é 0
            mean_value <- mean(log(y))
            force(mean_value)
            function(x){return(log(x) - mean_value)}
        } else{ # Quando p1 é diferente de 0
            mean_value <- mean(y^power1)
            force(mean_value)
            function(x){return(x^power1 - mean_value)}
        }
    } else{ # Quando tem p1 e p2
        if(power1 == power2){ # Quando p1 e p2 são iguais
            if(power1 == 0){ # Quando ambos são 0
                mean_value <- mean(log(y) + log(y)^2)
                force(mean_value)
                function(x){return(log(x) + log(x)^2 - mean_value)}
            } else{ # Quando ambos não são 0
                mean_value <- mean(y^power1 + (y^power1) * log(y))
                force(mean_value)
                function(x){
                    return(x^power1 + (x^power1)*log(x) - mean_value)
                }
            }
        } else{ # Quando p1 e p2 são diferentes
            if(power1 == 0){ # Quando p1 é 0
                mean_value <- mean(log(y) + y^power2)
                force(mean_value)
                function(x){log(x) + x^power1 - mean_value}
            } else if(power2 == 0){ # Quando p2 é 0
                mean_value <- mean(log(y) + y^power1)
                force(mean_value)
```

```
                function(x){log(x) + x^power2 - mean_value}
            } else{ # Quando nem p1 nem p2 é 0
                mean_value <- mean(y^power1 + y^power2)
                force(mean_value)
                function(x){x^power1 + x^power2 - mean_value}
            }
        }
    }
}
for(variable in variables){
    for(b in 1:B){ # Quando a variável não é escolhida
        functions <- append(functions,
                            fractional_polynomial(transformations[b,
                                                                  'power1',
                                                                  variable],
                            transformations[b, 'power2', variable],
                            data[, variable]))
    }
}
bootstrapped_functions <- array(functions,
                                c(B, length(variables)),
                                dimnames=list(1:B, variables))

# Calcula as bagged funcions
f_bag <- function(functions_for_variable){
    force(functions_for_variable)
    n_functions <- length(functions_for_variable)
    function(x){
        y <- 0
        for(b in 1:n_functions){
            y <- y + 1/n_functions * functions_for_variable[[b]](x)
        }
        return(y)
    }
}
f_bags <- c()
for(variable in variables){
    f_bags <- append(f_bags,
                     f_bag(bootstrapped_functions[, variable]))
```

```
}
names(f_bags) <- variables

# Determina as f_refs
f_ref <- c()
for(variable in variables){
    power1 <- mfp.object$powers[variable, 'power1']
    power2 <- mfp.object$powers[variable, 'power2']
    x <- data[, variable]
    f_ref <- append(f_ref, fractional_polynomial(power1, power2, x))
}
names(f_ref) <- variables

# Calcula as quantidades do diagnóstico
V <- c()
D2 <- c()
Vcond <- c()

n <- nrow(data)
for(variable in variables){
    V_variable <- 0
    Vcond_variable <- 0
    D2_variable <- 0
    for(i in 1:n){
        # Calcula o V_j
        x_ij <- data[i, variable]
        V_j <- 0
        Vcond_j <- 0

        # Calcula a frequência de inclusão da variável para calcular o Vcond
        n_r_x_j <- sum(!is.na(transformations[, 'power1', variable]))
        q_variable <- n_r_x_j/B

        for(i in 1:B){
            # Cálculo do V
            V_j <- (V_j
                    + 1/B
                    * (bootstrapped_functions[[b, variable]](x_ij)
                       - f_bags[[variable]](x_ij))^2)
```

```
                V_variable <- V_variable + 1/n * V_j

                # Cálculo do Vcond
                Vcond_j <- (Vcond_j
                            + 1/n_r_x_j
                            * (bootstrapped_functions[[b, variable]](x_ij)
                               - 1/q_variable * f_bags[[variable]](x_ij))^2)
                Vcond_variable <- Vcond_variable + 1/n * Vcond_j
            }
            # V_variable <- V_variable + 1/n * V_j
            # Vcond_variable <- Vcond_variable + 1/n * Vcond_j

            # Calcula o D2 da observação
            D2_variable <- (D2_variable
                            + 1/n
                            * (f_bags[[variable]](x_ij)
                               - f_ref[[variable]](x_ij))^2)
        }

        V <- append(V, V_variable)
        D2 <- append(D2, D2_variable)
        Vcond <- append(Vcond, Vcond_variable)
    }
    names(V) <- variables
    names(D2) <- variables
    names(Vcond) <- variables

    totals <- V + D2
    percentages_of_D2 <- D2/totals

    return(percentages_of_D2)
}
```

APÊNDICE C – FUNÇÃO PLOT_BAGGING

```
plot_bagging <- function(mfp.object,
                         formula,
                         data,
                         bootstrap_size,
                         x,
                         y,
                         xlab,
                         ylab,
                         seed,
                         alpha=0.05,
                         select=0.05,
                         B=1000){
    rescale <- mfp.object$rescale
    variables <- rownames(mfp.object$powers)
    transformations = array(NA, c(B, 2, length(variables)),
                            dimnames=list(1:B, c('power1', 'power2'), variables))
    coefficients <- array(NA, c(B, 2, length(variables) + 1),
                          dimnames=list(1:B,
                                        c('power1', 'power2'),
                                        append('Intercept', variables)))
    for(b in 1:B){
        set.seed(seed + b - 1)
        bootstrapped_data <- data[sample(1:nrow(data),
                                         size=bootstrap_size,
                                         replace=TRUE), ]
        # bootstrapped_data <- data
        rownames(bootstrapped_data) <- NULL
        bootstrapped_mfp <- mfp(
            formula=formula,
            data=bootstrapped_data,
            rescale=rescale,
            alpha=alpha,
            select=select
        )
        for(column in 1:ncol(bootstrapped_mfp$powers)){
            for(variable in variables){
                transformations[b,
```

```
                            column,
                            variable] <- bootstrapped_mfp$powers[variable,
                                                                 column]
        }
    }

    # Coeficientes
    coefficients[b,
                 'power1',
                 'Intercept'] <- bootstrapped_mfp$coefficients['Intercept']
    for(variable in variables){
        parameters_for_variable <- (
            !is.na(str_extract(names(bootstrapped_mfp$coefficients),
                               variable))
        )
        n_parameters <- sum(parameters_for_variable)
        if(n_parameters == 0){
            parameter1 <- NA
            parameter2 <- NA
        } else if(n_parameters == 1){
            parameter1 <- bootstrapped_mfp$coefficients[
                parameters_for_variable
            ]
            parameter2 <- NA
        } else{
            parameter1 <- bootstrapped_mfp$coefficients[
                parameters_for_variable
            ][1]
            parameter2 <- bootstrapped_mfp$coefficients[
                parameters_for_variable
            ][2]
        }
        coefficients[b, 'power1', variable] <- parameter1
        coefficients[b, 'power2', variable] <- parameter2
    }
}

# Calcula as funções bootstrap
functions <- c()
```

```
fractional_polynomial <- function(intercept,
                                  power1,
                                  power2,
                                  coefficient1,
                                  coefficient2){
    force(intercept)
    force(power1)
    force(power2)
    force(coefficient1)
    force(coefficient2)
    if(is.na(power1) & is.na(power2)){ # Quando a variável não é escolhida
        function(x) return(0)
    } else if(is.na(power2)){ # Quando só tem p1
        if(power1 == 0){ # Quando p1 é 0
            function(x){return(intercept + coefficient1 * log(x))}
        } else{ # Quando p1 é diferente de 0
            function(x){return(intercept + coefficient1 * x^power1)}
        }
    } else{ # Quando tem p1 e p2
        if(power1 == power2){ # Quando p1 e p2 são iguais
            if(power1 == 0){ # Quando ambos são 0
                function(x){return(intercept
                                   + coefficient1 * log(x)
                                   + coefficient2 * log(x)^2)}
            } else{ # Quando ambos não são 0
                function(x){
                    return(intercept
                           + coefficient1 * x^power1
                           + coefficient2 * (x^power1)*log(x))
                }
            }
        } else{ # Quando p1 e p2 são diferentes
            if(power1 == 0){ # Quando p1 é 0
                function(x){(intercept
                             + coefficient1 * log(x)
                             + coefficient2 * x^power1)}
            } else if(power2 == 0){ # Quando p2 é 0
                function(x){(intercept
                             + coefficient1 * log(x)
```

```
                            + coefficient2 * x^power2)}
            } else{ # Quando nem p1 nem p2 é 0
                function(x){(intercept
                            + coefficient1 * x^power1
                            + coefficient2 * x^power2)}
            }
        }
    }
}
for(variable in variables){
    for(b in 1:B){ # Quando a variável não é escolhida
        power1 <- transformations[b, 'power1', variable]
        power2 <- transformations[b, 'power2', variable]
        coefficient1 <- coefficients[b, 'power1', variable]
        coefficient2 <- coefficients[b, 'power2', variable]
        intercept <- coefficients[b, 'power1', 'Intercept']
        functions <- append(functions,
                            fractional_polynomial(intercept,
                                                  power1,
                                                  power2,
                                                  coefficient1,
                                                  coefficient2))
    }
}
bootstrapped_functions <- array(functions,
                                c(B, length(variables)),
                                dimnames=list(1:B, variables))

# Determina a função de referência
# Coeficientes da função de referência
coefficients_ref <- matrix(NA,
                           nrow=2,
                           ncol=length(variables) + 1)
rownames(coefficients_ref) <- list('power1', 'power2')
colnames(coefficients_ref) <- append('Intercept', variables)
coefficients_ref['power1',
                 'Intercept'] <- mfp.object$coefficients['Intercept']
for(variable in variables){
    parameters_for_variable <- (
```

```
        !is.na(str_extract(names(mfp.object$coefficients),
                           variable))
    )
    n_parameters <- sum(parameters_for_variable)
    if(n_parameters == 0){
        parameter1 <- NA
        parameter2 <- NA
    } else if(n_parameters == 1){
        parameter1 <- mfp.object$coefficients[
            parameters_for_variable
        ]
        parameter2 <- NA
    } else{
        parameter1 <- mfp.object$coefficients[
            parameters_for_variable
        ][1]
        parameter2 <- mfp.object$coefficients[
            parameters_for_variable
        ][2]
    }
    coefficients_ref['power1', variable] <- parameter1
    coefficients_ref['power2', variable] <- parameter2
}

f_ref <- c()
for(variable in variables){
    intercept <- mfp.object$coefficients['Intercept']
    power1 <- mfp.object$powers[variable, 'power1']
    power2 <- mfp.object$powers[variable, 'power2']
    coefficient1 <- coefficients_ref['power1', variable]
    coefficient2 <- coefficients_ref['power2', variable]
    fp <- fractional_polynomial(
        intercept,
        power1,
        power2,
        coefficient1,
        coefficient2
    )
    f_ref <- append(f_ref, fp)
```

```
}
names(f_ref) <- variables

# Calcula as bagged functions
f_bag <- function(functions_for_variable){
    force(functions_for_variable)
    n_functions <- length(functions_for_variable)
    function(x){
        y <- 0
        for(b in 1:n_functions){
            y <- y + 1/n_functions * functions_for_variable[[b]](x)
        }
        return(y)
    }
}
f_bags <- c()
for(variable in variables){
    f_bags <- append(f_bags,
                     f_bag(bootstrapped_functions[, variable]))
}
names(f_bags) <- variables

# Plota
if(is.null(xlab)){
    xlab <- x
}
if(is.null(ylab)){
    ylab <- y
}

g <- ggplot(data=data, aes(x=eval(parse(text=x)), y=eval(parse(text=y)))) +
    geom_point(shape=1)
for(b in 1:B){
    f <- bootstrapped_functions[[b, x]]
    g <- g +
        stat_function(fun=f, n=10000, aes(color='Bootstrap',
                                          alpha='Bootstrap'))
}
f <- f_bags[[x]]
```

```
    g <- g +
        stat_function(fun=f, n=10000, aes(color='Bagging',
                                          alpha='Bagging',
                                          size='Bagging'))
    f <- f_ref[[x]]
    g <- g +
        stat_function(fun=f, n=10000, aes(color='Modelo',
                                                alpha='Modelo',
                                                size='Modelo')) +
        scale_color_manual(name='Estimação',
                           breaks=c('Modelo', 'Bootstrap', 'Bagging'),
                           values=c('Modelo'='#00FDFF',
                                    'Bootstrap'='black',
                                    'Bagging'='#FF00C8')) +
        scale_alpha_manual(name='Estimação',
                           breaks=c('Modelo', 'Bootstrap', 'Bagging'),
                           values=c('Modelo'=1,
                                    'Bootstrap'=min(1/B*10, 1),
                                    'Bagging'=1)) +
        scale_size_manual(name='Estimação',
                          breaks=c('Modelo', 'Bootstrap', 'Bagging'),
                          values=c('Modelo'=1,
                                   'Bootstrap'=0.5,
                                   'Bagging'=1)) +
        theme_minimal() +
        removeGridX() +
        removeGridY() +
        ylab(ylab) +
        xlab(xlab) +
        theme(axis.line = element_line(colour = "black"),
              axis.ticks = element_line(colour = "black"))
    return(g)
}
```

APÊNDICE D – SCRIPT R USADO NA APLICAÇÃO

```
# Bibliotecas
require(mfp)
require(tidyverse)
require(ggExtra)

# Funções que serão utilizadas
plot_variable <- function(data, x, lab){
        g <- ggplot(data = data, mapping = aes(eval(parse(text=x)))) +
                stat_boxplot(geom = "errorbar") +
                geom_boxplot(outlier.shape = 21) +
                theme_minimal() +
                theme(axis.ticks.x = element_blank(),
                          axis.title.x = element_blank(),
                          axis.text.x = element_blank(),
                          axis.line = element_line(colour = "black")) +
                removeGrid(x = TRUE, y = TRUE) +
                xlab(lab) +
                coord_flip()

        return(g)
}

plot_variables <- function(data, x, y, xlab, ylab){
        g <- ggplot(
                data = data,
                mapping = aes(x = eval(parse(text=x)), y = eval(parse(text=y)))
        ) +
                geom_point(shape = 21) +
                theme_minimal() +
                theme(axis.ticks.x = element_blank(),
                          axis.line = element_line(colour = "black")) +
                removeGrid(x = TRUE, y = TRUE) +
                xlab(xlab) +
                ylab(ylab)

        return(g)
}
```

```
# Análise exploratória

plot_variable(bodyfat, "bmi", expression(IMC ~ (em ~ kg/m^2)))
plot_variable(bodyfat, "pbfm", "Taxa de gordura corporal")

bodyfat %>%
        summarize(
                minimo = min(pbfm),
                q1 = quantile(pbfm, .25),
                mediana = median(pbfm),
                media = mean(pbfm),
                q3 = quantile(pbfm, .75),
                maximo = max(pbfm),
                dp = sd(pbfm),
                cv = sd(pbfm)/mean(pbfm)
        )
bodyfat %>%
        summarize(
                minimo = min(bmi),
                q1 = quantile(bmi, .25),
                mediana = median(bmi),
                media = mean(bmi),
                q3 = quantile(bmi, .75),
                maximo = max(bmi),
                dp = sd(bmi),
                cv = sd(bmi)/mean(bmi)
        )

plot_variables(
        bodyfat,
        "bmi",
        "pbfm",
        expression(IMC ~ (em ~ kg/m^2)),
        "Porcentagem de gordura corporal"
)
cor.test(bodyfat$bmi, bodyfat$pbfm)
```

```
# Modelo linear
fit1 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)),
        formula = pbfm ~ bmi
); summary(fit1)

# Gráfico
ggplot(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)),
        aes(x = bmi, y = pbfm)
) +
        geom_point(shape = 21) +
        stat_function(
                fun = function(x)(
                        fit1$coefficients[1]
                        + fit1$coefficients[2] * x
                )
        ) +
        #xlab("IMC (centralizado na média, em kg/m") +
        xlab(expression(IMC ~ centralizado ~ na ~ média ~ em ~ kg/m^2)) +
        ylab("Porcentagem de gordura corporal") +
        theme_minimal() +
        theme(axis.line = element_line(colour = "black")) +
        removeGrid(x = TRUE, y = TRUE)


# Diagnóstico
envel.norm(fit1)
diag.norm(fit1, iden = 1)

# Obs influentes
bodyfat %>% slice(97)
bodyfat %>% slice(304)
bodyfat %>% slice(315)

# Sem a 97
fit1_97 <- lm(
```

```
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-97),
        formula = pbfm ~ bmi
); summary(fit1_97)

# Sem a 304
fit1_304 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-304),
        formula = pbfm ~ bmi
); summary(fit1_304)

# Sem a 315
fit1_315 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-315),
        formula = pbfm ~ bmi
); summary(fit1_315)

# Sem a 97 e a 304
fit1_97_304 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-c(97, 304)),
        formula = pbfm ~ bmi
); summary(fit1_97_304)

# Sem a 97 e a 315
fit1_97_315 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-c(97, 315)),
        formula = pbfm ~ bmi
); summary(fit1_97_315)

# Sem a 304 e a 315
```

```
fit1_304_315 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-c(304, 315)),
        formula = pbfm ~ bmi
); summary(fit1_304_315)

# Sem as três
fit1_97_304_315 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-c(97, 304, 315)),
        formula = pbfm ~ bmi
); summary(fit1_97_304_315)



#------------------#
# Modelo quadrático
#------------------#
fit2 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)),
        formula = pbfm ~ bmi + I(bmi^2)
); summary(fit2)

# Gráfico
ggplot(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)),
        aes(x = bmi, y = pbfm)
) +
        geom_point(shape = 21) +
        stat_function(
                fun = function(x)(
                        fit2$coefficients[1]
                        + fit2$coefficients[2] * x
                        + fit2$coefficients[3] * x^2)
        ) +
        xlab(expression(IMC ~ centralizado ~ na ~ média ~ (em ~ kg/m^2))) +
```

```
        ylab("Porcentagem de gordura corporal") +
        theme_minimal() +
        theme(axis.line = element_line(colour = "black")) +
        removeGrid(x = TRUE, y = TRUE)

# Diagnóstico
envel.norm(fit2)
diag.norm(fit2, iden = TRUE)

# Obs influentes
bodyfat %>% slice(123)
bodyfat %>% slice(304)

fit2_123 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-123),
        formula = pbfm ~ bmi + I(bmi^2)
); summary(fit2_123)

fit2_304 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-304),
        formula = pbfm ~ bmi + I(bmi^2)
); summary(fit2_304)

fit2_123_304 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)) %>%
                slice(-123, -304),
        formula = pbfm ~ bmi + I(bmi^2)
); summary(fit2_123_304)


#--------------#
# Modelo cúbico
#--------------#
fit3 <- lm(
```

```
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)),
        formula = pbfm ~ bmi + I(bmi^2) + I(bmi^3)
); summary(fit3)

# Diagnóstico
envel.norm(fit3)

f <- function(x) return(fit3$coefficients[1]
+ fit3$coefficients[2]*x
+ fit3$coefficients[3]*(x^2)
        + fit3$coefficients[4]*(x^3))
# Gráfico com a curva
ggplot(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi)),
        aes(x = bmi, y = pbfm)
) +
        geom_point(shape = 21) +
        stat_function(
                fun = function(x) fit3$coefficients[1]
                        + fit3$coefficients[2]*x
                        + fit3$coefficients[3]*(x^2)
                        + fit3$coefficients[4]*(x^3)
        ) +
        xlab(expression(IMC ~ centralizado ~ na ~ média ~ (em ~ kg/m^2))) +
        ylab("Porcentagem de gordura corporal") +
        theme_minimal() +
        theme(axis.line = element_line(colour = "black")) +
        removeGrid(x = TRUE, y = TRUE)

diag.norm(fit3, iden = TRUE)

# Análise confirmatória
# Sem o ponto 304
fit3_304 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi))
        %>% slice(-304),
```

```
        formula = pbfm ~ bmi + I(bmi^2) + I(bmi^3)
); summary(fit3_304)

# Sem o 315
fit3_315 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi))%>% slice(-315),
        formula = pbfm ~ bmi + I(bmi^2) + I(bmi^3)
); summary(fit3_315)

# Sem o 315 e o 304
fit3_304_315 <- lm(
        data = bodyfat %>%
                mutate(bmi = bmi - mean(bmi))%>% slice(-c(304, 315)),
        formula = pbfm ~ bmi + I(bmi^2) + I(bmi^3)
); summary(fit3_304_315)

#----------#
# Modelo FP
#----------#
fit4_mfp <- mfp(
        data = bodyfat,
        formula = pbfm ~ fp(bmi),
        alpha = 0.05,
        select = 0.05,
        rescale = TRUE
)
fit4 <- lm(
        data = bodyfat,
        formula = pbfm ~ I(bmi^(-1))
); summary(fit4)

# Gráfico com a curva
plot_fp(
        fit4,
        bodyfat,
        y = "pbfm",
        x = "bmi",
        xlab = expression(IMC ~ (em ~ kg/m^2)),
```

```
        ylab = "Porcentagem de gordura corporal"
)

# Diagnóstico
envel.norm(fit4)
diag.norm(fit4, iden= 1)

# Tirando as observações influentes
fit4_185 <- lm(
        data = bodyfat %>% slice(-188),
        formula = pbfm ~ I(bmi^(-1))
); summary(fit4_185)

# Comparacao
AIC(fit2); AIC(fit3); AIC(fit4)
BIC(fit2); BIC(fit3); BIC(fit4)

# Diagnóstico do MFP
bagging_diagnostics(
        fit4_mfp,
        pbfm ~ fp(bmi),
        bodyfat,
        bootstrap_size = 150,
        seed = 474276
)
plot_bagging(fit4_mfp,
                        pbfm ~ fp(bmi),
                        bodyfat,
                        bootstrap_size=150,
                        x='bmi',
                        y='pbfm',
                        xlab=expression(IMC ~ (em ~ kg/m^2)),
                        ylab='Porcentagem de gordura corporal',
                        seed=474276,
                        alpha=0.05,
                        select=0.05,
                        B=1000
)
```

```
# Todos os modelos no mesmo gráfico
fit1b <- lm(
        data = bodyfat,
        formula = pbfm ~ bmi
)
fit2b <- lm(
        data = bodyfat,
        formula = pbfm ~ bmi + I(bmi^2)
)
fit3b <- lm(
        data = bodyfat,
        formula = pbfm ~ bmi + I(bmi^2) + I(bmi^3)
)

ggplot(
        data = bodyfat,
        aes(x = bmi, y = pbfm)
) +
        geom_point(shape = 21) +
        stat_function(
                fun = function(x)(
                        + fit1b$coefficients[1]
                        + fit1b$coefficients[2] * x
                ),
                size = 1,
                n = 1e5,
                aes(color = "Linear")
        ) +
        stat_function(
                fun = function(x)(
                        + fit2b$coefficients[1]
                        + fit2b$coefficients[2] * x
                        + fit2b$coefficients[3] * x^2
                ),
                size = 1,
                n = 1e5,
                aes(color = "Quadrático")
        ) +
        stat_function(
```

```
        fun = function(x)(
                + fit3b$coefficients[1]
                + fit3b$coefficients[2] * x
                + fit3b$coefficients[3] * x^2
                + fit3b$coefficients[4] * x^3
        ),
        size = 1,
        n = 1e5,
        aes(color = "Cúbico")
) +
stat_function(
        fun = function(x)(
                + fit4$coefficients[1]
                + fit4$coefficients[2] * x^(-1)
        ),
        size = 1,
        n = 1e5,
        aes(color = "Polinomial fracionário")
) +
scale_color_manual(
        name = "Modelo",
        breaks = c("Linear", "Quadrático", "Cúbico", "Polinomial fracionário"),
        values = c(
                "Linear" = "#f691b2",
                "Quadrático" = "#c3af62",
                "Cúbico" = "#3cc5af",
                "Polinomial fracionário" = "#88b3fd"
        )
) +
xlab(expression(IMC ~ (em ~ km/m^2))) +
ylab("Porcentagem de gourdura corporal") +
theme_minimal() +
theme(axis.line = element_line(colour = "black")) +
removeGrid(x = TRUE, y = TRUE)
```