

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INTITULO DE ECONOMIA

TESE DE DOUTORADO

Modelos autorregressivos funcionais: Uma aplicação aos dados da carga de energia elétrica horária.

Lucélia Viviane Vaz Raad

Tese apresentada ao corpo docente do Instituto de Economia da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários à obtenção do título de doutor em Ciências, em Economia.

Orientador: Prof. Getulio Borges da Silveira Filho

Março/2014

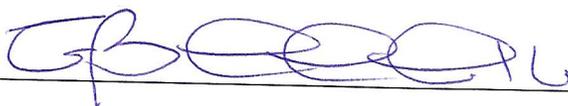
Rio de Janeiro

Lucélia Viviane Vaz Raad

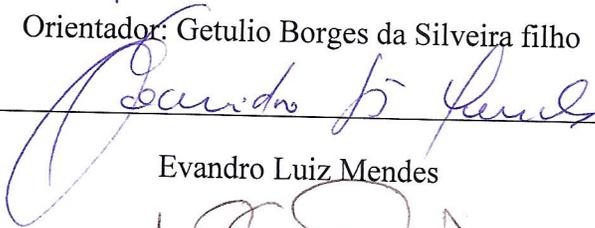
Modelos autorregressivos funcionais: Uma aplicação aos dados da carga de energia elétrica horária.

Tese apresentada ao corpo docente do Instituto de Economia da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários à obtenção do título de doutor em Ciências, em Economia.

BANCA EXAMINADORA:



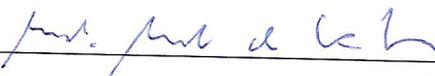
Orientador: Getulio Borges da Silveira filho



Evandro Luiz Mendes



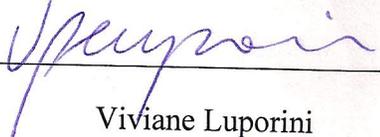
Luiz Otávio Façanha



Rudi Rocha



Victor Hugo Carvalho de Gouveia



Viviane Luporini

Março/2014

Rio de Janeiro

FICHA CATALOGRÁFICA

R111 Raad, Lucélia Viviane Vaz.

Modelos autorregressivos funcionais: uma aplicação aos dados da carga de energia elétrica horária / Lucélia Viviane Vaz Raad. – 2014.

223 f. ; 31 cm.

Orientador: Getulio Borges da Silveira Filho.

Tese (doutorado) – Universidade Federal do Rio de Janeiro, Instituto de Economia, Programa de Pós-Graduação em Economia, 2014.

Bibliografia: f. 110-113.

1. Análise de dados funcionais. 2. Modelos periódicos autorregressivos. 3. Curvas de carga de energia elétrica. I. Silveira Filho, Getulio Borges da. II. Universidade Federal do Rio de Janeiro. Instituto de Economia. III. Título.

CDD 333.7932

Resumo

As curvas de carga [que são definidas como a potência elétrica requerida por um sistema como função do tempo] exibem três características principais: (1) forte inercia, (2) comportamento funcional [significando que cada observação é uma função] e (3) movimentos cíclicos. Nesta tese consideramos modelos de regressão lineares apropriados para lidar com essas três características.

Os modelos que desenvolvemos podem ser classificados como simultâneos e não simultâneos. Tratamos de três tipos de modelos simultâneos. No primeiro tipo de modelo, lidamos com generalizações funcionais dos modelos periódicos autorregressivos [baseados nas evoluções diárias da carga]. No segundo tipo de modelo, os parâmetros autorregressivos variam de maneira sazonal [baseados nas evoluções semanais]. E no terceiro tipo de modelo [para as evoluções anuais da carga], a periodicidade dos parâmetros autorregressivos é induzida através dos chamados operadores de aceleração harmônica [ver Ramsay and Silverman (2010)]. Os modelos baseados nas evoluções anuais se mostraram os mais adequados pelo critério da RMSE.

Nos modelos do tipo simultâneo, o valor corrente do processo resposta se relaciona somente com o valor corrente do processo preditor. Versões mais gerais de modelos para dados funcionais podem ser encontradas em Senturk e Muller (2010). Nesses modelos o valor corrente do processo resposta $Y(t)$ no tempo t depende da história recente do processo preditor X em uma janela de tamanho Δ , de uma determinada forma. Propomos algumas alterações na forma de relacionamento das variáveis dependente e explicativa proposta por Senturk e Muller (2010). Para dar conta de tais alterações, apresentamos uma generalização funcional do estimador de posto reduzido. O novo modelo é chamado modelo funcional não simultâneo. A fim de testar o comportamento do estimador proposto, realizamos uma série de simulações.

Os dados, obtidos junto ao Operador Nacional do Sistema elétrico, são constituídos por observações horárias da carga e referem-se ao subsistema sudeste/centro-oeste para o período 01/03/2003 a 20/01/2011.

Palavras-chaves: Análise de dados funcionais, modelos periódicos autorregressivos, curvas de carga de energia elétrica.

Abstract

The load curves [which are defined as the electrical power required by a system as a function of time] exhibit three main features: (1) strong inertia, (2) functional behavior [meaning that each observation is a function] and (3) cyclical movements. In this thesis, we consider linear regression models to deal with these three characteristics.

The developed models can be classified as non-concurrent and concurrent. We deal with three types of concurrent models. In the first type of model, we deal with functional generalizations of periodic autoregressive models [they are based on daily evolutions of the load]. In the second type of model, the autoregressive parameters vary seasonally [they are based on weekly evolutions of the load]. And the third type of model [for annual evolutions of the load], the periodicity of the autoregressive parameters is induced by so-called harmonic acceleration operators[see Ramsay and Silverman (2010)]. The models based on annual evolutions were the most suitable based on RMSE criterion.

In concurrent linear models, the current value of the response process is related only with the current value of the covariate. More general version of models for functional data can be found in Senturk and Muller (2010). In these models, the current value of the response process $Y(t)$ at time t depends on the recent history of the predictor process in a sliding window of length Δ , in a certain way. We propose some changes in the form of the relationship of the dependent and explanatory variables considered by Senturk and Muller (2010). We do this through a functional generalization of the reduced-rank estimator. The new model is called non-concurrent functional model. In order to test the behavior of the proposed estimator, we conducted a small simulations study.

The database was obtained from the National Operator of the Electrical System, ONS, and consists of observations of hourly electricity load for southeast and midwest electrical subsystem of Brazil for the period 01/03/2003 to 20/01/2011.

Keywords: Functional data analysis, periodic autoregressive models, electricity load curves.

Agradecimentos

Gostaria de ter o dom de uma poetisa, para transcrever em palavras a minha alegria em terminar esta tese de doutorado. Mas, diante de toda minha limitação, escrevo algumas simples, mas sinceras palavras.

Em primeiro lugar, agradeço a Deus por ter me dado força nos momentos de angústia. Não foram poucos. Aos meus pais, Daniel e Adélia, pelo apoio incondicional, pelo amor e carinho de sempre. Eles sempre valorizaram a educação e me deram força para eu ir até o fim. Ao meu marido, Rodrigo, pela força e paciência. Aos meus irmãos, Sérgio e Carlos, pelo carinho e amor. Minha querida avó, Nina, que partiu ao longo dessa tese deixando saudade e culpa por não ter dedicado mais tempo a ela.

Agradeço especialmente ao meu orientador, professor Getulio Borges da Silveira Filho, pela dedicação. Ele foi certamente a pessoa que mais contribuiu para a minha formação profissional e me ensinou a buscar o rigor acadêmico.

Os meus amigos também foram muito importantes ao longo dessa caminhada: Wilson (o Calmon), Henrique, Melissa, Igor, Débora, Poliana, Aline, Deborah, o pessoal do Brumado (pelos momentos em que pude pensar em outras coisas que não a tese de doutorado).

Aos professores Luiz Otávio Façanha, Rudi Rocha, Viviane Luporini, Victor Hugo Carvalho de Gouveia e Eduardo Pontual Ribeiro, membros da banca examinadora do Exame de Qualificação e/ou da Defesa da Tese, pelas valiosas contribuições e direções apontadas para continuidade e conclusão desse estudo.

Ao Operador Nacional do Sistema Elétrico, na pessoa do engenheiro do Evandro Luiz Mendes, pelo acesso aos dados e pelos valiosos comentários. Finalmente, agradeço ao apoio financeiro da CAPES – Coordenação de Aperfeiçoamento do Pessoal de Nível superior.

Sumário

	Introdução	1
1	Modelos Periódicos funcionais	4
1.1	Modelos periódicos autorregressivos e periódicos funcionais	4
1.2	Splines	6
1.3	Testes de hipóteses em modelos de regressão funcionais	11
2	Aplicação aos dados da carga de energia elétrica horária	14
2.1	Trabalhos dedicados à análise das curvas de carga	15
2.2	Análise exploratória da base de dados	17
2.2.1	Carga	17
2.2.2	Temperatura	26
2.3	Modelagem da carga elétrica horária	29
2.3.1	Modelo funcional diário	29
2.3.2	Modelo funcional semanal	36
2.3.3	Modelo de regressão não-paramétrica	40
2.3.4	Modelo anual	46
3	Modelos funcionais não simultâneos	57
3.1	O estimador de posto reduzido	57
3.2	Análise de correlações canônicas para dados funcionais	64
3.3	Modelos funcionais não simultâneos	66
3.3.1	A generalização funcional do modelo de posto reduzido	71
3.4	Simulações	74
3.4.1	Modelo com uma variável explicativa	74
3.4.2	Modelo com duas variáveis explicativas	76
3.5	Modelo autorregressivo com variável exógena não simultânea	86

3.6	Continuidade no modelo funcional não simultâneo	88
4	Modelo funcional não simultâneo aplicado à carga elétrica horária	93
4.1	Modelo semanal	93
4.2	Modelo anual	97
5	Conclusão	108
6	Bibliografia	110
7	Apêndice 1	114
7.1	Formulas seção 4.4.1	114
7.2	Fórmulas seção 4.4.2	115
7.3	Fórmulas seção 4.5	116
7.4	Apêndice 2	118
7.5	Programação do modelo diário	118
7.6	Programação do modelo semanal	165
7.7	Programação do modelo anual	182
7.8	Modelos não simultâneos	191
7.8.1	Modelo semanal	191
7.8.2	Modelo anual	207

LISTA DE FIGURAS

1.1	Bases para splines cúbicos.	8
1.2	Carga de domingos de inverno e de verão.	13
1.3	Exemplo de saída da função F_{perm} .	13
2.1	Média Diária da Carga.	17
2.2	Carga ajustada pela carga média da semana anterior.	18
2.3	Curvas de carga para terças-feiras de inverno e de verão.	19
2.4	Efeito dos dias da semana sobre a curva de carga para os dias 28/11/2010 a 04/12/2010.	20
2.5	Efeito do feriado sobre o formato da curva de carga - Semana que vai de 05/09/2010 a 11/09/2010.	21
2.6	Curva média da carga semanal somada e subtraída da primeira componente principal.	24
2.7	Curva média da carga semanal somada e subtraída da segunda componente principal.	25
2.8	Curva média da carga semanal somada e subtraída da terceira componente principal.	25
2.9	Curva média da carga semanal somada e subtraída da quarta componente principal.	26
2.10	Média diária da temperatura.	27
2.11	Comportamento da carga e da temperatura ao longo do dia para os dias 15/11/2002 e 08/08/2008.	28
2.12	Coefficientes associados à carga defasada nos modelos dos domingos, quartas e sextas-feiras.	31
2.13	Coefficiente associado à primeira cossenóide nos modelos dos sábados, das quartas e das sextas-feiras.	31
2.14	Coefficiente associado à segunda senoide nos modelos dos sábados, das quartas e sextas-feiras.	32
2.15	Teste de permutação para a significância conjunta dos coeficientes do modelo diário para o domingo.	32
2.16	Teste de permutação para a significância conjunta dos coeficientes do modelo diário para a segunda-feira.	33
2.17	Valores previstos para o domingo 29/08/2012, a segunda-feira 30/08/2012, a terça-feira 31/08/2012 e a quarta-feira 01/09/2010.	35

2.18	Média das previsões feitas para todos os domingos, segundas, terças e quartas-feiras observadas na amostra.	35
2.19	RMSE por hora e dia do modelo diário.	36
2.20	Parâmetros estimados para a carga defasada das semanas em que vale o horário de verão e a constante.	37
2.21	Teste de permutação para significância conjunta dos coeficientes do modelo semanal.	38
2.22	RMSE's para cada hora da semana associadas ao modelo semanal.	39
2.23	Gráfico das RMSE's do modelo semanal.	39
2.24	Coefficientes da carga defasada no modelo semanal.	40
2.25	Ilustração dos pesos utilizados na estimação dos modelos semanais.	45
2.26	Constante estimada no modelo das semanas 2, 4, 26, 34, 50 e 51.	45
2.27	Coefficiente estimado para a carga defasada nos modelos das semanas 2,4,26,34,50 e 51.	46
2.28	Exemplos de variações de fase e amplitude.	48
2.29	Gráfico da função $\cos((2/365)\pi t) + \cos((2/7)\pi t)$ avaliada nos inteiros.	53
2.30	Teste de permutação para os coeficientes estimados no modelo anual.	53
2.31	Boxplot da raiz quadrada do erro quadrático médio do modelo anual para algumas horas selecionadas do dia.	54
2.32	BoxPlot da RQEQM algumas horas selecionadas nos dias úteis e nos finais de semana de acordo com o modelo diário.	55
2.33	BoxPlot do RQEQM para as horas de referências nos dias úteis e nos finais de semana de acordo com o modelo semanal.	56
2.34	Constante estimada nos modelos semanal e anual.	56
3.1	Alguns X's e Y's simulados.	77
3.2	HIF's estimadas para cada um dos períodos.	78
3.3	Beta estimado no modelo funcional não simultâneo.	78

3.4	HIF's estimadas para cada um dos períodos em um modelo com duas variáveis explicativas. Apenas a variável X entra no modelo de maneira não simultânea.	79
3.5	HIF's associadas à variável explicativa X, em um modelo com duas variáveis explicativas, X e Temp.	81
3.6	HIF's estimadas associadas a X, em um modelo com duas variáveis explicativas, que apresentam correlação entre elas.	82
3.7	HIF's estimadas associadas a Temp, em um modelo com duas variáveis explicativas, que apresentam correlação entre elas.	83
3.8	HIF's estimadas para a temperatura após o tratamento para a dependência entre as variáveis explicativas.	85
3.9	Constante avaliada em dois intervalos ao longo do ciclo.	87
3.10	Variável exógena avaliada em dois intervalos ao longo do ciclo.	88
3.11	Exemplo de patamares.	89
3.12	Função dependente avaliada em dois intervalos ao longo de um mesmo ciclo.	89
3.13	HIF's estimadas para madrugadas, manhãs, tardes e noites.	90
3.14	Exemplo de projeção em base de splines dos patamares dados por	92
4.1	HIF's estimadas no modelo semanal para cada período do dia e do ano.	96
4.2	RMSE's provenientes dos modelos simultâneo e não simultâneo semanal.	97
4.3	RMSE's provenientes dos modelos simultâneo e não simultâneo semanal.	98
4.4	RMSE's provenientes dos modelos simultâneo e não simultâneo semanal.	99
4.5	RMSE's provenientes dos modelos simultâneo e não simultâneo semanal.	100
4.6	HIF's estimadas para a temperatura no modelo anual.	102
4.7	RMSE's provenientes dos modelos simultâneo e não simultâneo anual.	104
4.8	RMSE's provenientes dos modelos simultâneo e não simultâneo anual.	105
4.9	RMSE's provenientes dos modelos simultâneo e não simultâneo anual.	106

4.10	RMSE's provenientes dos modelos simultâneo e não simultâneo anual.	107
------	--	-------	-----

Introdução

Modelos estatísticos são coleções de distribuições de probabilidade que, acreditamos, podem ter originado os dados que nos dispomos a investigar. Quando os dados são funções, dizemos lidar com modelos para dados funcionais. A modelagem de dados funcionais trata processos probabilísticos que geram dados da forma Y_1, Y_2, \dots, Y_n em que $\{Y_j\}_{j \leq n}$ é uma família de funções com o mesmo domínio. O uso de técnicas próprias de dados funcionais vem ganhando notoriedade em economia com aplicações, por exemplo, a curvas de juros [Silveira & Bessada (2003) e Caldeira & Torrent (2011)]. Destacam-se também aplicações a dados de mercado de trabalho [modelos de salários], através de generalizações funcionais dos "modelos de variáveis instrumentais" [Cai, Das, Xiong & Wu (2006)]. Aplicações em diversas áreas podem ser encontradas em Ramsay & Silverman (1997).

Dados econômicos apresentam, muitas vezes, inércia e algum tipo de comportamento cíclico. Diversos modelos foram desenvolvidos a fim de tratar tais características em séries de tempo univariadas. Nos modelos estruturais [Harvey, et.all (1993)], por exemplo, a inércia e os movimentos cíclicos são tratados através de componentes não observáveis com características de tendência sazonal e cíclica, respectivamente. Os modelos SARMA [Box & Jenkins(1970)] são uma particularização dos modelos Autorregressivos e de Médias Móveis para os casos em que o processo subjacente apresenta movimentos cíclicos. Neste caso, além dos polinômios autorregressivos [que visam lidar com a inércia] e de média móvel, são incluídos [de forma multiplicativa] dois polinômios relacionados à periodicidade e também, possivelmente, podem ser tomadas diferenças sazonais. Nos modelos periódicos autorregressivos os parâmetros variam de forma a captar o ciclo. Por exemplo, em dados trimestrais os parâmetros são específicos para cada trimestre. Franses & Paap (2004) modelaram diversas séries de consumo norte-americanas e evidenciaram a importância dessa classe de modelos na modelagem de séries econômicas sazonais.

Nesta tese consideramos modelos apropriados para lidar com dados que exibam três características básicas (1) inercia, (2) ciclos e (3) dados funcionais. Tratamos de três tipos modelos. No primeiro tipo de modelo, lidamos com generalizações funcionais dos modelos

periódicos autorregressivos. No segundo tipo de modelo, os parâmetros autorregressivos variam de maneira sazonal. E no terceiro tipo de modelo, a periodicidade dos parâmetros é induzida através dos chamados operadores de aceleração harmônica [ver Ramsay & Silverman (2010)].

Neste trabalho daremos especial atenção à aplicação dos modelos anteriores às curvas de carga de energia elétrica. Vale ressaltar que os modelos aqui desenvolvidos podem ser aplicados a outros processos, por exemplo, dados de vazões fluviais.

A curva de carga é definida como a potência elétrica requerida por um sistema como função do tempo. Apesar de vários trabalhos, que serão descritos no capítulo 3, terem se dedicado à modelagem da carga, a nossa proposta é inovadora principalmente na consideração do aspecto funcional da carga. As curvas de carga são funções. Por limitações dos instrumentos de medida, observamos as funções que descrevem a carga apenas em alguns pontos específicos de seus domínios. Mais especificamente, dispomos de observações horárias da carga, para o subsistema sudeste/centro-oeste, que foram obtidas junto ao Operador Nacional do Sistema elétrico.

O principal problema de não tratar dados que são pontos (ou discretizações) ao longo do domínio de uma função como tais é a perda da estrutura de correlação para observações de tempos próximos entre si. No caso da carga este problema é marcante nos modelos com uma equação para cada hora do dia [ver Ramanathan et. al. (1997)].

A tese está organizada em quatro capítulos além desta introdução. Iniciamos o primeiro capítulo apresentando os modelos periódicos autorregressivos e logo após apresentamos como, via *splines*, estimamos funções desconhecidas a partir de um conjunto de observações [ou discretizações] ao longo de seus domínios. Tendo obtido as funções estimadas, caracterizamos a variabilidade destas. Além disso, justificamos o uso dos modelos funcionais, os quais definimos formalmente, para explicar a variabilidade entre as funções. Por fim, fazemos uma breve explanação sobre a estimação dos parâmetros de um modelo funcional e apresentamos os testes de permutação que serão utilizados para testes de hipóteses apropriadas referentes aos coeficientes dos modelos periódicos funcionais.

O segundo capítulo trata da aplicação dos modelos periódicos funcionais aos dados da carga de energia elétrica. Para isso, apresentamos uma revisão da literatura dedicada à modelagem da carga, uma análise exploratória da base de dados e os três modelos estimados. O primeiro deles, é um modelo periódico funcional baseado nas evoluções diárias das curvas de carga. O segundo está baseado nas evoluções semanais da carga e o terceiro na evolução anual da carga.

Os modelos tratados nos primeiro e segundo capítulos são chamados modelos simultâneos. Isto significa que a variável dependente no tempo t se relaciona com as variáveis explicativas apenas no tempo t . No terceiro capítulo, a nossa proposta é que a variável dependente no tempo t passe a se relacionar com uma média ponderada da variável explicativa em um intervalo de tempo que contém t . Os pesos serão dados por uma função. Ideia semelhante pode ser encontrada em Senturk & Muller (2010). O estimador para a função que designa os pesos está baseado em uma generalização funcional do estimador de posto reduzido. Ainda no terceiro capítulo, conduzimos exercícios de simulação com vistas a demonstrar o comportamento do estimador de posto reduzido funcional. O modelo proposto será denominado modelo funcional não simultâneo.

No quarto capítulo, faremos a aplicação do modelo funcional não simultâneo aos dados da carga de energia elétrica. O modelo será aplicado às trajetórias semanais e anuais da carga.

Capítulo 1

Modelos Periódicos funcionais

Neste capítulo descrevemos formalmente os modelos periódicos autorregressivos univariados. A partir deles, apresentamos a generalização para o caso funcional. Nos modelos periódicos autorregressivos univariados a observação para cada unidade de tempo é um característico numérico. No contexto de dados funcionais, associamos a cada unidade de tempo¹, um dia, por exemplo, uma função.

As funções que estão associadas a cada unidade de tempo podem ser desconhecidas. No caso da carga de energia elétrica, conhecemos apenas discretizações das mesmas para cada hora do dia. Sendo assim, precisamos recuperar as funções que estão gerando as observações horárias. Isto será feito através do uso de *splines*, conforme será descrito nas seções seguintes.

Descrevemos também como são estimados os parâmetros de um modelo funcional e os testes de permutação que serão utilizados para testar a significância de um conjunto de covariadas.

1.1 Modelos periódicos autorregressivos e periódicos funcionais

Os modelos periódicos autorregressivos tradicionais têm como principal característica o fato de os parâmetros variarem (periodicamente) visando captar movimentos cíclicos da série de tempo. Considere $y := \{Y_n\}_{n \in \mathbb{Z}}$ processo estocástico tomando valores em R . Defina $F_n := \sigma(Y_n, Y_{n-1}, \dots)$, em que $\sigma(Y_n, Y_{n-1}, \dots)$ é a σ -álgebra gerada por $\{Y_n, Y_{n-1}, \dots\}$. No restante da tese $\sigma(\cdot)$ indicará sempre a σ -álgebra gerada por $\{\cdot\}$. O processo y é dito

¹Fora de um contexto de séries de tempo, teríamos uma função associada a cada elemento amostral. Por exemplo, a função eletrocardiograma de cada indivíduo que compõe a amostra.

periódico autorregressivo quando, para algum $p > 0$ e $s > 0$, podemos escrever:

$$E(Y_n | \mathcal{F}_{n-1}) = \mu_n + \phi_{1n} Y_{n-1} + \cdots + \phi_{pn} Y_{n-p} \quad (1.1)$$

em que $\mu_n = \mu_{n+s}$, $\phi_{1n} = \phi_{1,n+s}$, $\phi_{2n} = \phi_{2,n+s}$, \cdots , $\phi_{pn} = \phi_{p,n+s}$. Defina $\varepsilon_n = Y_n - E(Y_n | \mathcal{F}_{n-1})$. Assumimos também que $\text{var}(\varepsilon_n) = \text{var}(\varepsilon_{n+s})$, para todo $n \in \mathbb{Z}$. O índice s representa o período subjacente e p é a ordem autorregressiva.

Uma generalização importante dos modelos periódicos autorregressivos é obtida quando permitimos que as unidades observacionais sejam funções e não mais valores aleatórios. Uma outra diz respeito à possível presença de variáveis exógenas no modelo. Tais generalizações são particularmente relevantes para a modelagem da carga, onde estamos lidando com curvas para as quais variáveis climáticas, principalmente a temperatura, afetam fortemente o seu comportamento.

Seja $\xi := (\xi_t, t \in T)$ uma família de elementos aleatórios definidos no espaço de probabilidades (Ω, \mathcal{A}, P) e tomando valores em um espaço mensurável (E, B) . Quando T é um intervalo em \mathbb{R} , ξ é um processo a tempo-contínuo. Se $(E, B) = (\mathbb{R}, B_{\mathbb{R}})$, em que $B_{\mathbb{R}}$ é a σ -álgebra de Borel de \mathbb{R} , ξ é dito um Processo Estocástico Real.

Em algumas aplicações $T = (0, \infty)$ é conveniente segmentarmos o domínio do processo ξ , gerando sequência $[Y_k, k = 0, 1, 2, \dots]$ com o mesmo domínio²:

$$Y_k(t) = \xi_{kh+t}, 0 \leq t \leq h; k = 0, 1, \dots$$

com $h > 0$, fixo. A sequência Y_0, Y_1, \dots de variáveis toma valores no espaço de funções reais com domínio $[0, h]$. Em nosso caso, iremos impor ainda que as Y 's são contínuas.

Considere as sequências $\{Y_n, n \geq 0\}$ e $\{W_n, n \geq 0\}_{n \geq 0}$ de processos definidos nos mesmos moldes que no paragrafo anterior. Faça $G_{n-1} := \sigma(Y_m, W_{m+1} : m \leq n-1)$. Diremos que o processo $\{Y_n\}_{n \in \mathbb{Z}}$ é periódico autorregressivo funcional com variáveis

²Esta é a abordagem adotada em Bosq(2000)

exógenas quando podemos escrever:

$$E(Y_n(t)|\mathcal{G}_{n-1}) = \mu_n(t) + \sum_{i=1}^p \phi_{in}(t)Y_{n-i}(t) + \beta_n(t)W_n(t) \quad (1.2)$$

em que $\mu_n(t) = \mu_{n+s}(t)$, $\phi_{1,n}(t) = \phi_{1,n+s}(t)$, $\phi_{2,n}(t) = \phi_{2,n+s}(t)$, \dots , $\phi_{p,n}(t) = \phi_{p,n+s}(t)$, $\beta_n(t) = \beta_{n+s}(t)$. Defina $\varepsilon_n(t) = Y_n(t) - E(Y_n(t)|\mathcal{G}_{n-1})$. Assumimos também que $Var(\varepsilon_n(t)) = Var(\varepsilon_{n+s}(t))$, para todo $n \in Z$ e $t \in [a, b]$. O índice s é o período subjacente e p é a ordem autorregressiva.

Os modelos como em (1.2) são chamados na literatura [Ramsay & Silverman (1997)] de modelos simultâneos [*concurrent models*], já que na equação de regressão de (1.2) entram somente os regressores avaliados no tempo t . Abordagens mais gerais podem ser encontradas em Senturk & Muller (2010), Bosq (2000) dentre outros.

As covariadas e as funções dependentes são observadas para alguns pontos de seus domínios [ou são observadas determinadas discretizações das mesmas]. Sendo assim, é preciso recuperar as funções desconhecidas que estão gerando os dados. Isso será feito mapeando as funções discretizadas em um conjunto de splines, conforme descreveremos na próxima seção.

1.2 Splines

Um *spline* é função $f : [a, b] \rightarrow R$, polinomial por partes. Mais especificamente, a cada *spline* f estão associados $a = t_0 < t_1 < \dots < t_n < t_{n+1} = b$ de modo que a restrição, P_j , de f a cada um dos $[t_j, t_{j+1}]$, $j = 0, 1, \dots, n$ é um polinômio e $Dom(f) = [a, b]$. O grau do spline f , dizemos, m , é o maior dos graus dos P_j 's. Impomos ainda que f e suas $m - 1$ primeiras derivadas sejam contínuas. Os pontos t_0, t_1, \dots, t_{n+1} são ditos os nós do spline f . Os nós t_1, \dots, t_n são ditos nós interiores.

Para m fixo, é claro que a classe dos splines de grau m e com nós $a = t_0 < t_1 < \dots < t_n < t_{n+1} = b$ é um subespaço vetorial de $C_{[a,b]}^{m-1}$, o espaço das funções com domínio $[a, b]$ e $m - 1$ derivadas contínuas. Splines cúbicos [grau= 3] são os mais utilizados em aplicações.

Neste caso, podemos escrever

$$P_j(t) = a_j + b_j(t - t_j) + c_j(t - t_j)^2 + d_j(t - t_j)^3$$

para constantes $a_j, b_j, c_j, d_j, j = 0, 1, \dots, n$, apropriadamente escolhidas.

A continuidade de f e de suas duas primeiras derivadas estabelece algumas restrições quanto à escolha das constantes acima. É fácil ver [Green & Silverman (1994)] que as $4(n+1)$ constantes $[a_0, \dots, d_n]$ devem satisfazer a $3n$ restrições [lineares] induzidas pela continuidade de f e suas derivadas nos nós interiores. Decorre daí que $[a_0, \dots, d_n]$ reside em um subespaço vetorial de $R^{4(n+1)}$ com dimensão $n+4$. Consequência trivial: a classe dos splines de grau m [domínio $[a, b]$ e nós $a = t_0 < t_1 < \dots < t_n < t_{n+1} = b$] é espaço vetorial de dimensão $n+4$.

Um *spline* é dito natural quando ambos P_0 e P_n têm grau 1 [são retas, portanto]. Quando é assim $c_0 = d_0 = 0 = c_n = d_n$. Quatro novas restrições, portanto. Isso faz com que (a_0, \dots, d_n) passe a residir em um subespaço n -dimensional. Seguem daí:

1. Qualquer *spline* cúbico natural é univocamente determinado pelos valores que assume em seus nós.
2. A classe dos *splines* cúbicos [cúbicos naturais, respectivamente] com nós $a = t_0 < t_1 < \dots < t_n < t_{n+1} = b$ é um subespaço vetorial de $C_{[a,b]}^2$ com dimensão $(n+4)$ [n , respectivamente].

Na figura (1.1), exibimos duas possíveis bases para *splines* cúbicos com nós $a = 0 < 20 < 40 < 80 < 110 = b$.

Dados $t_0 < t_1 < \dots < t_n < t_{n+1}$ e y_1, \dots, y_n números reais, iremos denominar o *spline* f com nós $a = t_0 < t_1 < \dots < t_n < t_{n+1} = b$ e tal que $f(t_j) = y_j$ de *spline* interpolante. Assim, dada $h : [a, b] \rightarrow R$ e $a < t_1 < \dots < t_n < b$ existe um único *spline* interpolante \tilde{h} tal que $\tilde{h}(t_j) = h(t_j), j = 1, \dots, n$. Para obter esse spline basta resolver um sistema de equações lineares.

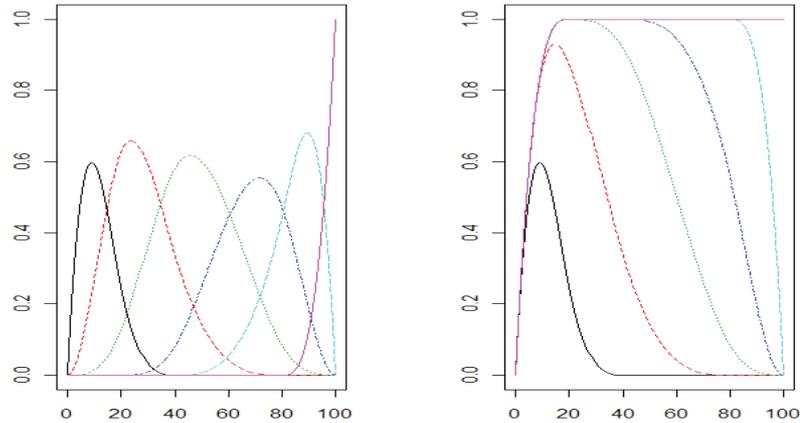


Figura 1.1: Bases para *splines* cúbicos

Splines interpolantes, salvo em casos onde a função original é muito suave, podem exibir ondulações indesejáveis. Isso é particularmente verdade, quando os valores da função original são observados com erros de medida.

O interesse no uso de *splines* no presente contexto é obter uma representação de funções, para as quais conhecemos apenas alguns pontos ao longo de seu domínio. A representação deve ser suave. A imposição de suavidade é o reconhecimento tácito da existência de erros de medida. Isso, por sua vez, justifica o mapeamento das funções discretizadas em funções suaves que não necessariamente interpolam os dados originais³. Mais especificamente, para um conjunto de pontos observados ao longo do domínio de uma função, queremos exibir uma combinação linear de *splines* de tal forma a se ajustar bem aos pontos observados e que atenda a determinados critérios de suavidade.

O bom ajuste aos dados é medido, como tradicionalmente é feito, em termos da soma de quadrados de resíduos. O critério de suavidade está baseado no quadrado da derivada

³No caso dos dados da carga alguns nós devem necessariamente ser mantidos. Eles são 7, 11,13,18,19,20. Nesses pontos ocorrem características importantes da curva de carga, por exemplo, mudanças de concavidade. Entre 18 e 20 horas ocorre o horário de pico e no horário de pico as variações da carga são de fato fortes, dessa forma permitiremos a não continuidade da derivada primeira nesses pontos.

segunda. Os dois objetivos podem ser resumidos em um problema de minimização de quadrado de resíduos sujeito a uma restrição de suavidade. Em termos matemáticos escrevemos:

$$\min \left\{ \sum_j [y_j - h(t_j)]^2 + \lambda \int_a^b [h''(t)]^2 dt : h(t) \in C_{[a,b]}^2 \right\} \quad (1.3)$$

O primeiro termo mede o ajuste e o segundo a curvatura da função estimada. A constante λ é o parâmetro de suavização. Quanto maior o valor de λ mais suaves serão as funções estimadas. Pode-se demonstrar [Reinsch, C. (1967)] que o único argumento que minimiza (1.3), para um λ fixo, é um *spline* natural cúbico com nós t_j , para $j = 1, \dots, n$. Este *spline* é denominado *spline* de suavização.

Na prática, observamos pontos do domínio de um conjunto n de funções, para cada conjunto de pontos do domínio de uma determinada função i obtemos um *spline* h_i que para um dado λ satisfaz (1.3).

Tendo em vista que em (1.3) penalizamos excesso de variabilidade local [possivelmente induzida pelos erros de medida], os *splines* h_i 's obtidos lidam com parte da variabilidade exibida pelos pontos observados ao longo do domínio da função y_i .

No caso da carga, por exemplo, esta técnica não é suficiente para exaurir toda a variabilidade das curvas. Isto significa que os *splines* h_i 's, $i = 1, \dots, n$, obtidos através de (1.3) exibem uma variabilidade que deve ser tratada.

Uma possível fonte de variabilidade para os h_i 's que representam a carga é sua resposta a variáveis climáticas, principalmente a temperatura. Um dos efeitos da temperatura sobre as curvas de carga é provocar deslocamentos verticais das mesmas. Na figura (1.2), temos as curvas de carga de alguns domingos de inverno e de verão. Podemos observar que, pelo menos até às 17 horas, as curvas de carga dos domingos de verão estão acima das curvas de carga dos domingos de inverno. Existem também alguns deslocamentos horizontais de características marcantes das curvas. Deslocamentos do horário de pico⁴, por exemplo. O

⁴O operador nacional do sistema elétrico define o horário de pico de consumo como sendo as três horas consecutivas do dia na qual ocorre a demanda máxima, exceção feita aos sábados, domingos e feriados

horário de pico no verão ocorre mais tarde do que no inverno e mesmo entre as curvas do verão, pode-se observar deslocamentos do horário de pico. Essa variabilidade remanescente será tratada através da especificação de modelos funcionais que serão explicados nas seções seguintes.

Na próxima seção apresentamos como se dá a estimação dos parâmetros de um modelo funcional quando as funções são representadas através de *splines*.

Estimação dos parâmetros de um modelo funcional

Para motivar a estimação dos parâmetros de um modelo funcional reescrevemos abaixo o modelo que aparece em (1.2):

$$E(Y_n(t)|\mathcal{G}_{n-1}(t)) = \mu(t) + \sum_{i=1}^p \phi_i(t)Y_{n-i}(t) + \beta(t)X_n(t)$$

em que $Y_n, Y_{n-1}, \dots, Y_{n-p}$ são expressas como combinações lineares de componentes de uma determinada base de splines e X_n como combinação linear de elementos de uma base genérica [por exemplo, uma base de Fourier, ver Ramsay & Silverman (2010)]. Por simplicidade de exposição, adotamos duas simplificações. A primeira é que basearemos nossa apresentação em um modelo sem defasagens e com apenas uma variável explicativa, digamos X . A segunda é que consideramos que Y_n e X_n estão centralizadas, isto é, subtraídas de suas respectivas médias. A estimação de médias funcionais é discutida em detalhes em Rice & Silverman (1991).

De maneira semelhante aos modelos tradicionais de regressão, nos modelos funcionais minimizamos uma espécie de soma de quadrados de resíduos. No caso tradicional, o vetor de resíduos é dado por $y - \beta x$. Também é assim para dados funcionais. A principal diferença é que as entradas do vetor $y - \beta x$ passam a ser funções. A notação fica mais clara quando deixamos explícita a dependência em t . O estimador natural, $\tilde{\beta}(t)$, de $\beta(t)$ é dado

nacionais.

por:

$$\tilde{\beta}(t) = \arg \min_{\beta \in C_{[a,b]}^2} \left\{ \int_0^T \|y(t) - \beta(t)x(t)\|^2 dt \right\} \quad (1.4)$$

em que $x(t) := [x_1(t), \dots, x_N(t)]^\top$ e $y(t) := [y_1(t), \dots, y_N(t)]^\top$ e x_n, y_n são as realizações de X_n e Y_n . As expressões de x_n e y_n nas respectivas bases de funções são dadas por:

$$x_n(t) = \sum_{j=1}^J g_{nj} \phi_j(t) \quad \text{e} \quad y_n(t) = \sum_{k=1}^K h_{nk} \varphi_k(t) \quad \text{para todo } t \in (a, b] .$$

É fácil ver que $\beta(t)$ pode ser escrito como:

$$\beta(t) = \sum_{j=1}^J \sum_{k=1}^K b_{jk} \phi_j(t) \varphi_k(t) = \phi^\top(t) B \varphi(t) \quad \text{para todo } t \in (a, b]$$

em que B é matrix $J \times K$. De forma mais compacta escrevemos:

$$x(t) = G\phi(t) \quad , \quad y(t) = H\varphi(t)$$

G, H são matrizes de dimensão $N \times J, N \times K$, respectivamente. Defina $\varepsilon(t) = y(t) - x(t)\beta(t) = H\varphi(t) - G\phi(t)\phi^\top(t)B\varphi(t)$. Escreva $J_\phi = \phi(t)\phi^\top(t)$ e $J_\varphi = \varphi(t)\varphi^\top(t)$, logo $\varepsilon(t) = (H - GJ_\phi B)\varphi(t)$. Usando que $\text{tr}(\varepsilon(t)\varepsilon^\top(t)) = \varepsilon^\top(t)\varepsilon(t)$, temos então que $\varepsilon^\top(t)\varepsilon(t) = \text{tr}\left((H - GJ_\phi(t)B)J_\varphi(t)(H - GJ_\phi(t)B)^\top\right)$. Logo o problema (1.4) pode ser reescrito como:

$$\min \left\{ \int_0^T \text{tr}\left((H - GJ_\phi(t)B)J_\varphi(t)(H - GJ_\phi(t)B)^\top\right) dt : B \in R^{JK} \right\}$$

cuja solução fornece os estimadores de β .

1.3 Testes de hipóteses em modelos de regressão funcionais

Em modelos de regressão, funcionais ou não, é natural testar a significância estatística das covariadas para explicar a variável dependente em questão. Nos modelos lineares clássicos isto é feito, em geral através da razão da soma de quadrados de resíduos [estatística F] da

seguinte forma:

$$F = \frac{RSS_0 - RSS_1}{RSS_1}$$

No caso clássico, dadas certas condições [ver, Davidson & MacKinnon (2004)] a estatística acima, sob a hipótese nula, tem distribuição F . Para o caso funcional Ramsay & Silverman (2010) propõem que sejam fixados pontos ao longo do domínio das funções, para os quais as estatísticas F de permutação serão avaliadas. Para cada um desses pontos são fornecidos p-valores obtidos como nos testes de permutação usuais, ver Good (2005). No pacote R isso é implementado através da função $Fperm$ cuja saída é um gráfico do tipo exibido na figura (1.3).

Nos testes de permutação a distribuição sob a hipótese nula é estimada diretamente a partir dos dados observados. A ideia é que se não existe relação entre a variável resposta e as covariadas então não faria diferença se aleatoriamente rearranjássemos os pares. A vantagem dos testes de permutação é que não precisamos assumir distribuições hipotéticas. A desvantagem é que não podemos fazer testes de significância de uma covariada individual em um modelo com mais de uma [covariada].

Mais especificamente, considere o seguinte modelo funcional linear:

$$Y_i(t) = \beta_0(t) + \beta_1(t)X_i(t) + \varepsilon_i(t) \quad (1.5)$$

para cada função dependente observada $Y_i(\cdot)$ existe uma função explicativa observada correspondente $X_i(\cdot)$. Para implementar o teste, fixamos alguns valores para t [t_1, \dots, t_R digamos], obtemos a soma dos quadrados dos resíduos referente a cada ponto e computamos as estatísticas F' s correspondentes a cada ponto, as quais chamamos F_{obs} . Para obter a distribuição aproximada da estatística de teste sob a hipótese nula, permutamos as funções dependentes observadas enquanto mantemos inalteradas as funções explicativas observadas. Para cada permutação, re-estimamos o modelo em (1.5) e computamos as correspondentes estatísticas F' s, obtendo a distribuição de permutação nos pontos fixados. Depois disso, comparamos as F_{obs} com a distribuição de permutação nos pontos fixados.

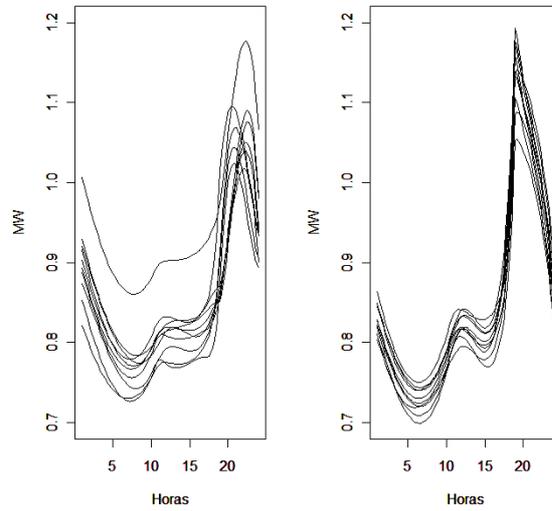


Figura 1-2: Carga de domingos de inverno e de verão.

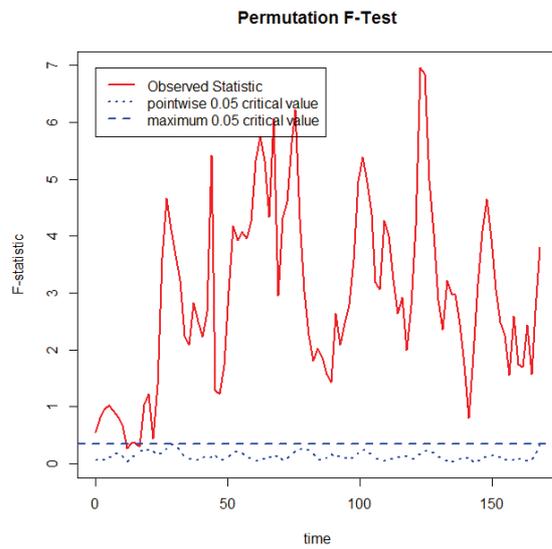


Figura 1-3: Exemplo de saída da função Fperm.

Capítulo 2

Aplicação aos dados da carga de energia elétrica horária

O objetivo central deste capítulo é proceder à modelagem da carga de energia elétrica horária. Iniciamos com uma breve revisão da literatura dedicada à modelagem da carga. Em seguida, conduzimos uma análise exploratória da base de dados, na qual incluímos análise de componentes principais funcionais. A descrição da estrutura de variabilidade dos dados ajuda a entender as estratégias de modelagem adotadas. Foram considerados três modelos. No primeiro deles, optamos por um modelo baseado nas evoluções diárias da carga. Depois disso, passamos a um modelo para as evoluções semanais e por fim um modelo para as evoluções anuais da carga. Em todos eles as variáveis explicativas são: uma função que desempenha o papel de constante e uma função que desempenha o mesmo papel que a *dummy* nos modelos tradicionais de regressão para os dias referentes a feriados, a carga defasada [em um dia], a temperatura do respectivo dia e eventualmente alguma componente para captar sazonalidade.

Começamos com um modelo para cada dia da semana, isso porque, como veremos na seção 2 deste capítulo, o dia da semana afeta o formato da curva de carga. A curva de carga dos sábados e domingos é diferente da curva de carga dos dias úteis, por exemplo. Mesmo entre os dias úteis existem diferenças [a curva de carga da segunda-feira apresenta um nível de consumo abaixo de todos os outros dias da semana durante as horas da madrugada]. Dessa forma, justifica-se a especificação de um modelo que permita parâmetros próprios para cada dia da semana. No modelo de cada dia da semana serão adicionadas componentes de seno e cosseno visando capturar as possíveis variações sazonais que não as induzidas

por temperatura [um exemplo de variação sazonal, não necessariamente captada por temperatura, é a luminosidade solar que tem efeito direto sobre a carga].

Pontencialmente, a principal desvantagem associada ao modelo diário é a descontinuidade dos parâmetros ao longo da semana. Se um modelo é especificado para as evoluções semanais da carga, então resolvemos o problema de descontinuidade dos parâmetros entre os dias da semana. Além disso, nesse modelo, que denominaremos de semanal, permitiremos que os parâmetros associados à carga defasada variem com as estações do ano.

No modelo semanal, observamos que a forma como os parâmetros variam ao longo do ano ocorre de maneira muito particular. Eles exibem uma certa suavidade. Isso nos levou a especificação de um outro modelo, agora para a evolução anual da carga. Para a especificação do modelo anual faremos uso dos chamados operadores de aceleração harmônica.

2.1 Trabalhos dedicados à análise das curvas de carga

A energia elétrica é um insumo básico de vários processos produtivos e é também utilizada diretamente pelas famílias através de serviços de iluminação, refrigeração dentre outros. Sendo assim, quaisquer fontes de custos neste setor afetam diretamente a renda real das famílias. A correta descrição da estrutura de variabilidade das curvas de carga ajuda a mitigar custos relacionados a despachos abaixo ou acima da verdadeira demanda. Quando a geração de energia é incapaz de suprir a demanda podem ocorrer blackouts. Neste caso, temos uma série de custos para a sociedade como um todo, por exemplo, distúrbios no trânsito, na segurança pública, equipamentos elétricos/eletrônicos danificados, dentre outros. Despachos acima da demanda levam ao desperdício de insumos [água, gás, óleo combustível, principais insumos da matriz brasileira]. Apesar de a geração hidráulica utilizar um insumo relativamente barato, associado ao uso da água, temos o custo de oportunidade dado pela utilização de insumos de alto custo [gás, óleo combustível] em períodos de escassez de chuvas.

A importância do conhecimento da série de carga e suas principais fontes de variabilidade é corroborada pela ampla gama de trabalhos dedicados à modelagem da mesma. Bunn & Farmer (1985) apresentam uma coleção de artigos dedicados ao assunto. Dentre estes, Ackerman (1985) compara três modelos alternativos. No primeiro, foi especificado um $AR(24)$ para os dados horários da carga. No segundo, os dados foram diferenciados e modelados como um $ARMA(24, 1)$. Gupta (1985) estimou um modelo no qual a série é explicada por suas defasagens e a temperatura através de um modelo com parâmetros variando no tempo.

Schneider et. al. (1985) consideraram cada hora do dia como sendo função de uma carga básica [específica da hora em questão], de uma componente que capta o desvio da temperatura em relação a uma temperatura base, de um termo de erro e um termo de correção de viés. Ramanathan et. al. (1997) desenvolveram um modelo de regressão multivariado com uma equação para cada hora. Seguindo essa linha, Dordonnat et. al. (2008), Cottet & Smith (2003) também consideraram modelos de séries de tempo univariadas para cada uma das horas do dia. Aplicações deste tipo de modelo a dados brasileiros podem ser encontrados em Soares & Medeiros (2008) e Soares & Souza (2006). Ambos utilizaram dados de concessionárias de energia elétrica localizadas na região sudeste do Brasil.

Harvey & Koopman (1993) modelaram a carga através de modelos estruturais utilizando splines com coeficientes variando no tempo para tratar as questões de periodicidade do consumo.

Alguns dos trabalhos anteriores consideram a associação da carga com fatores climáticos, especialmente, a temperatura. Também é assim em Engle et. al. (1996).

Os trabalhos aqui citados se tratam de modelos tradicionais de séries de tempo. A nossa proposta é uma generalização funcional de modelos de séries temporais, com vistas a tratar curvas de carga como funções.

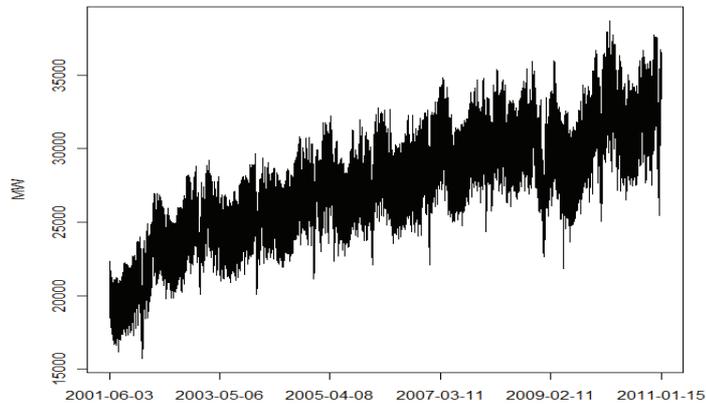


Figura 2.1: Média Diária da Carga

2.2 Análise exploratória da base de dados

2.2.1 Carga

A base de dados contém as observações horárias da carga [em MW] e da temperatura [em graus Celsius], referentes ao subsistema sudeste/centro-oeste [compreendendo as regiões de mesmos nomes] para o período 01 de janeiro de 2003 até 20 de janeiro de 2011. Na figura (2.1), podemos observar que a série da carga apresenta uma tendência de crescimento ao longo do tempo. Essa tendência se deve basicamente à resposta ao nível de atividade econômica. Aqui, não nos dedicaremos à modelagem da relação entre a carga e a atividade econômica. Mesmo assim é necessário lidar com a tendência de longo prazo.

Uma possível abordagem, que não advogamos como ótima, é transformar os dados de forma que eles não exibam tendência de longo prazo. Uma maneira de fazer isso é modelar as variações da carga em vez de seus níveis. Mais especificamente, iremos modelar a evolução das variações percentuais das cargas relativamente à carga horária média da semana imediatamente anterior. A figura (2.2) dá evidências da ausência de eventuais tendências de crescimento dos dados assim transformados.

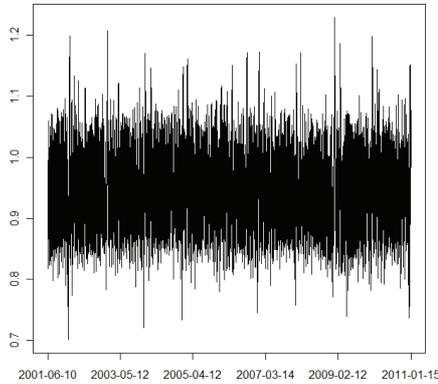


Figura 2.2: Carga ajustada pela carga média da semana anterior.

As curvas de carga apresentam periodicidade diária, semanal e anual conforme descreve a literatura [ver, por exemplo, Harvey & Koopman (1993)]. Um exemplo de comportamento com periodicidade diária é a ocorrência do horário de pico, afora nos dias mais quentes de verão, no início das noites e do horário de menor consumo durante as madrugadas. Na figura (2.3), observamos curvas de carga para algumas terças-feiras. As curvas referentes às terças-feiras de inverno apresentam um comportamento bastante similar entre elas. O horário de pico ocorre por volta das 19 horas. As terças-feiras de verão apresentam uma maior diversidade de formatos. Em pelo menos uma curva, o horário de pico ocorre durante a tarde enquanto nas outras o horário de pico ocorre entre 18 e 21 horas. Apesar da maior diversidade de formatos, cada curva de verão mostra dois vales bem definidos e pelo menos um pico, assim como as curvas do inverno.

A característica mais marcante da periodicidade semanal é a influência dos dias úteis e dos finais de semana sobre o formato da curva de carga, figura (2.4). As curvas de carga do sábado e do domingo são significativamente diferentes das curvas dos outros dias da semana. Geralmente, as curvas dos sábados e domingos apresentam um nível de carga mais baixo que os dias úteis da semana. Além disso, as madrugadas das segundas apresentam, em

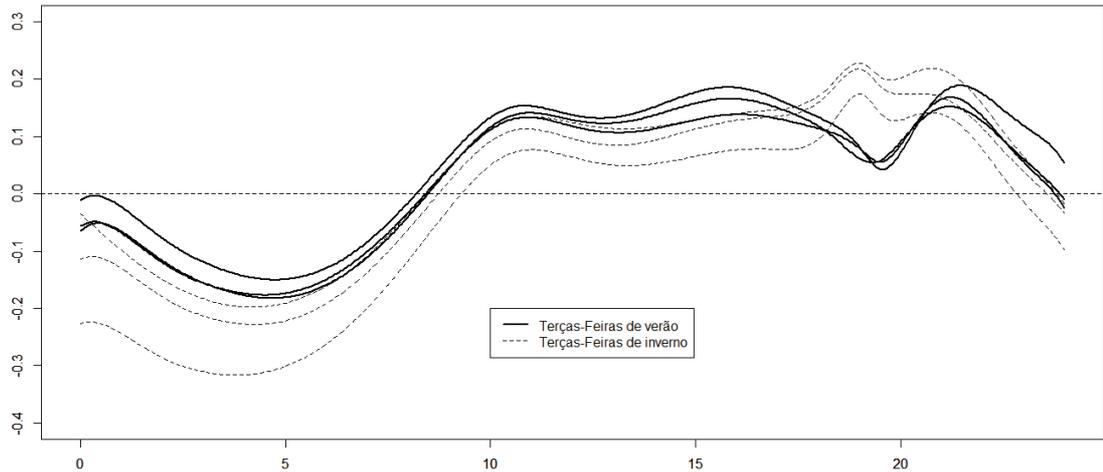


Figura 2-3: Curvas de carga para terças-feiras de inverno e de verão.

geral, um nível de carga abaixo daquele observado em todos os outros dias da semana.

A periodicidade anual (sazonalidade) é devida basicamente a fatores climáticos. Por exemplo, terças-feiras de inverno exibem formas típicas distintas das terças-feiras de verão [ver figura (2-3)].

Ressaltamos também o efeito dos feriados sobre o formato da curva de carga. A figura (2-5) mostra o gráfico das curvas de carga para cada um dos dias da semana que vai de 05/09/2010 a 11/09/2010. O feriado da independência comemorado no dia 07/09 ocorreu em uma terça-feira. Podemos observar a mudança no formato da curva típica de uma terça-feira [figura (2-4)], para a terça-feira do feriado de 07/09/2010. A curva de uma terça-feira típica é parecida com as curvas das quartas, quintas e sextas-feiras, enquanto o formato da curva da terça-feira de feriado é mais parecido com o formato dos dias do final de semana.

Os feriados e as periodicidades diária, semanal e anual são algumas das principais fontes de variabilidade das curvas de carga. Podem haver outras fontes de variabilidade

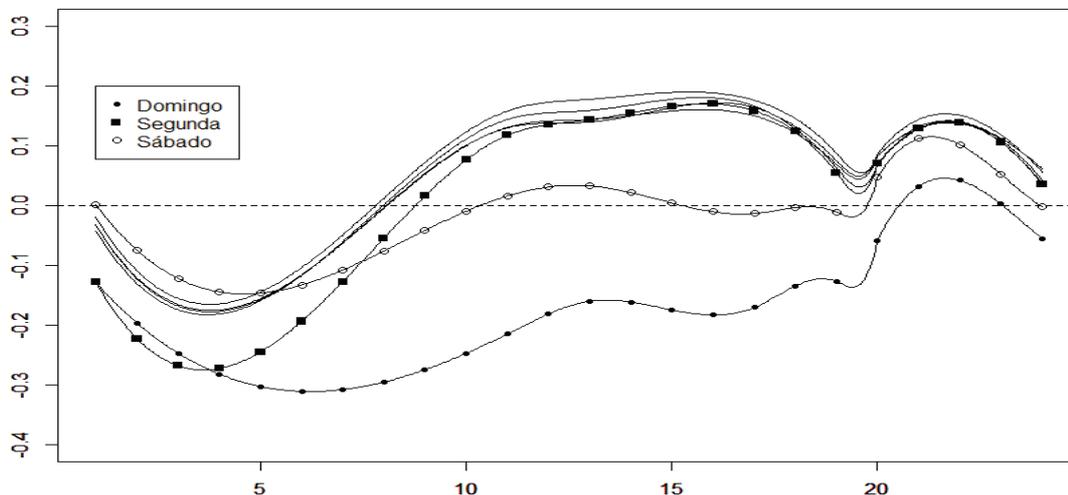


Figura 2-4: Efeito dos dias da semana sobre a curva de carga para os dias 28/11/2010 a 04/12/2010

não perceptíveis através da inspeção direta das curvas. Quando os dados são observações de vetores aleatórios de R^d , a análise de componentes principais é técnica importante para analisar a variabilidade dos dados [ver, por exemplo, Gnanadesikam (1997) e Anderson (1984) para uma exposição didática sobre o assunto]. Mais formalmente, seja X vetor aleatório p -dimensional com média zero. A análise de componentes principais está baseada em uma sequência ortonormal de autovetores ψ_1, \dots, ψ_p , com a propriedade que para cada k tal que $1 \leq k \leq p$ e para $\psi_1, \dots, \psi_{k-1}$, a distância:

$$d(k) = E \left\| X - \sum_{j=1}^k \psi_j \psi_j^\top X \right\|^2$$

s.a. $\|\psi_j\| = 1$

é minimizada para todas as possíveis escolhas de ψ_k . $\|x\|$ é a norma euclidiana de x . A ortonormalidade dos autovetores ψ_j 's implica que as variáveis aleatórias $\psi_j^\top X$, $\psi_k^\top X$ são não

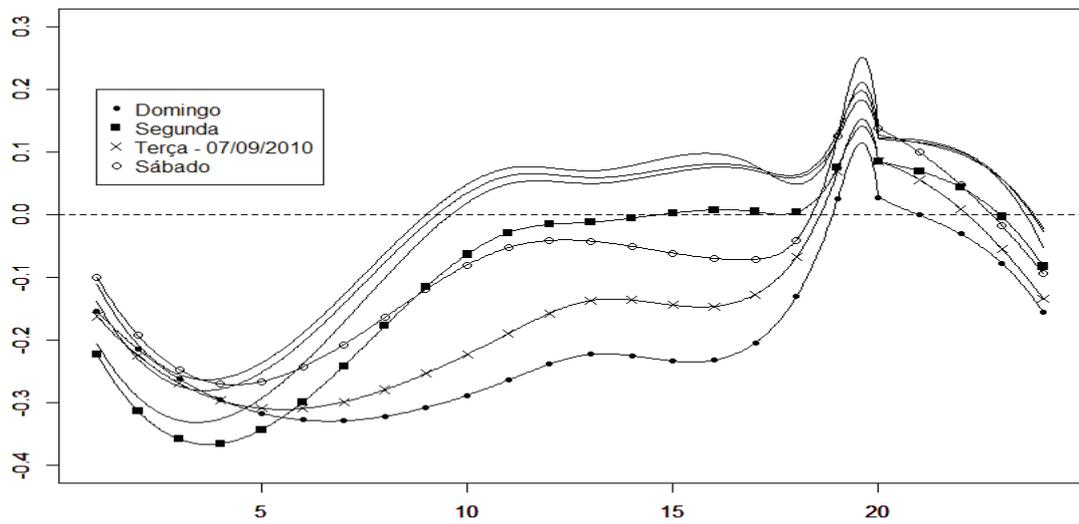


Figura 2-5: Efeito do Feriado sobre o formato da curva de carga - Semana que vai de 05/09/2010 a 11/09/2010

correlacionadas para todo $k \neq j$. As respectivas variâncias são dadas por $\theta_j = \text{var}(\psi_j^\top X)$ e formam uma sequência não negativa e não decrescente, isto é,

$$\theta_1 \geq \theta_2 \geq \dots \geq \theta_p \geq 0$$

A matriz de covariância de X pode ser dada por:

$$K = \sum_{j=1}^p \theta_j \psi_j \psi_j^\top$$

sendo assim, para cada j , $K\psi_j = \theta_j\psi_j$ para $1 \leq j \leq p$. Isto é, ψ_j é um autovetor de K com autovalor associado θ_j . A k -ésima componente principal é dada por $\psi_k^\top X$.

Aplicações econômicas de análise de componentes principais podem ser encontradas em Litterman & Scheinkman (1991). Eles analisaram a evolução das estruturas a termo de taxas de juros e identificaram 3 componentes capazes de explicar algo em torno de 98% da variabilidade das taxas implícitas a papéis de várias maturidades no mercado americano. As três componentes obtidas estavam relacionadas a movimentos no nível, na inclinação e na curvatura da curva de juros.

Quando as observações se tratam de funções, determinamos ζ_1, ζ_2, \dots funções que capturam, em ordem decrescente, as fontes de variabilidade dos dados. O procedimento é análogo ao descrito no caso clássico, para redefinições adequadas do produto interno e da norma.

De maneira mais formal, seja X função aleatória com suporte no intervalo compacto I e satisfazendo:

$$\int_I E [X(s)^2] ds < \infty \quad (2.1)$$

como decorrência de (2.1) a função de covariância de X admite decomposição espectral convergente em L^2 de I^2 dada por:

$$K(s, t) \equiv \text{cov} \{X(s), X(t)\} = \sum_{j=1}^{\infty} \theta_j \psi_j(s) \psi_j(t)$$

em que $\theta_1 \geq \theta_2 \geq \dots \geq 0$ são autovalores com autofunções ψ_j do operador linear com núcleo K . Cada função ψ_j pode ser definida indutivamente como sendo a função que minimiza:

$$d(k) = E \left\| X - \sum_{j=1}^k \psi_j \int_I \psi_j X \right\|^2$$

$$s.a. \|\psi_j\| = 1$$

em que $\|x\|$ é dada pela raiz quadrada de $\int_I [x(t)]^2 dt$. O j -ésimo escore principal é dado por $\theta_j^{-\frac{1}{2}} \int_I \psi_j(t) X(t) dt$ e a j -ésima componente principal, ζ_j , é dada por $\zeta_j = \psi_j$. A ζ_j se caracteriza por explicar a maior parte possível da variação de X , além do nível de variação explicado usando as ψ'_k s anteriores, isto é, $k < j$.

Nas figuras (2.6), (2.7), (2.8) e (2.9) temos a média funcional para todas as semanas observadas, somada e subtraída das quatro primeiras componentes. Elas são responsáveis por explicar 52%, 26%, 7% e 4% da variabilidade, respectivamente. O artifício de adicionar/subtrair as componentes principais à média foi originalmente utilizado por Ramsay & Silverman (1997) e contribuiu fortemente para a interpretação das componentes principais.

A primeira componente principal mostra que aproximadamente metade da variabilidade das curvas de carga em relação à média é devida a deslocamentos verticais das mesmas. Esses deslocamentos não afetam o formato da curva de carga.

A variabilidade associada à segunda componente principal está relacionada principalmente a deslocamentos próximos do horário de pico. Semanas com grandes escores na segunda componente principal mostram maiores picos quando comparadas com a curva média. Além disso, podemos observar uma espécie de mudança de escala temporal [tradução nossa para time shifting] próximo do horário de pico [para semanas com grandes escores na segunda componente principal o pico ocorre mais cedo quando comparado à curva média].

Semanas com grandes escores na terceira componente principal são tais que para os

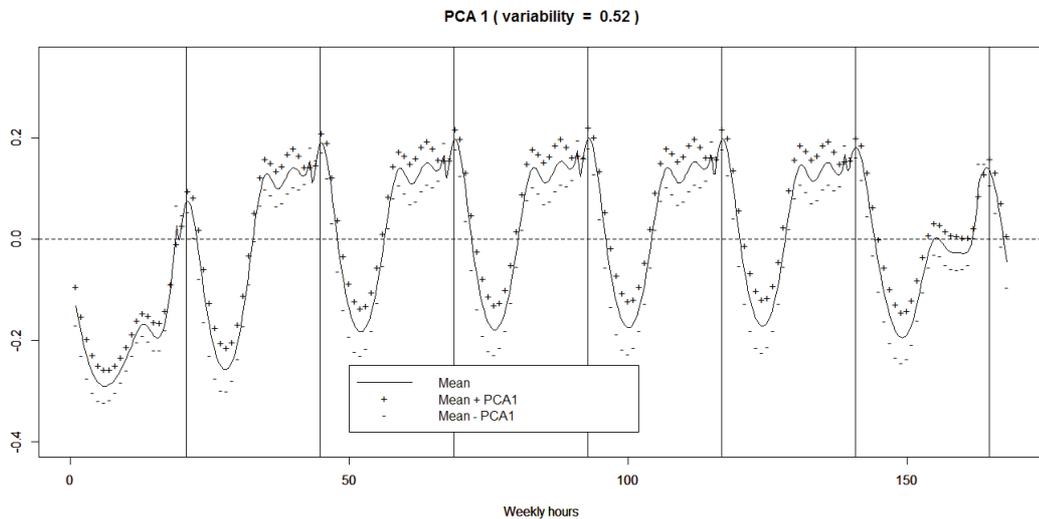


Figura 2-6: Curva média da carga semanal somada e subtraída da primeira componente principal. As linhas verticais marcam os dias da semana.

dias no início da semana [domingo, segunda e terça] as curvas de carga são maiores que a curva média. Elas apresentam ainda, nos dias próximos ao final de semana [quintas, sextas e sábados] curvas de carga menores que a curva média. A terceira componente principal está associada a uma mudança de inclinação da curva média de carga.

A variabilidade associada à quarta componente principal está relacionada à curvatura. Semanas com grandes escores na quarta componente principal apresentam uma grande curvatura nos seus extremos [domingos, segundas, sextas e sábados] e menores curvaturas nos dias centrais da semana [terças, quartas e quintas-feiras]

Assim como no caso clássico, cujo exemplo são as estruturas a termo da taxa de juros analisadas por Litterman & Scheinkman (1991), encontramos que a primeira, a terceira e a quarta componentes principais funcionais estão relacionadas a movimentos no nível, na inclinação e na curvatura das curva de carga, respectivamente. A segunda componente principal é mais típica de dados funcionais, tendo em vista que está relacionada a uma tipo de mudança de escala temporal.

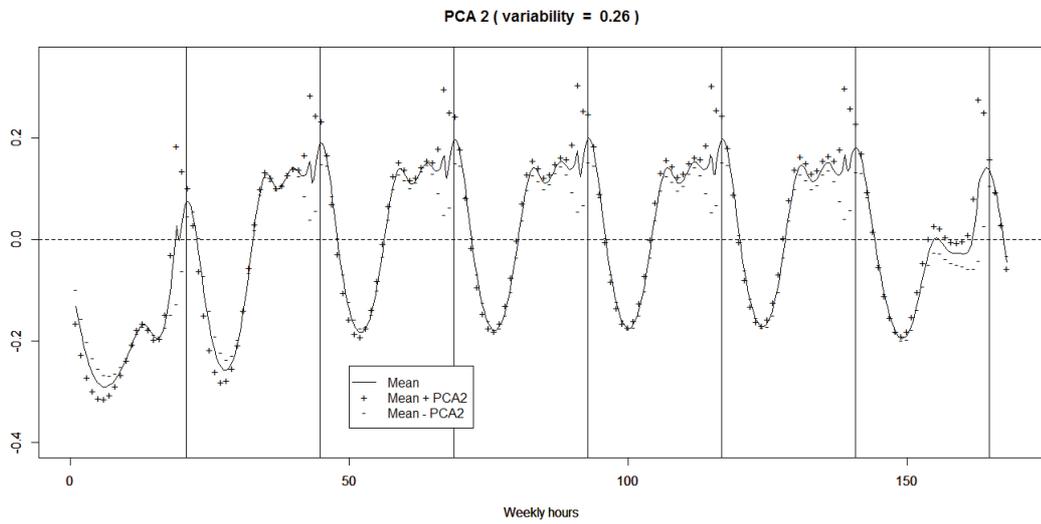


Figura 2-7: Curva média da carga semanal somada e subtraída da segunda componente principal. As linhas verticais marcam os dias da semana.

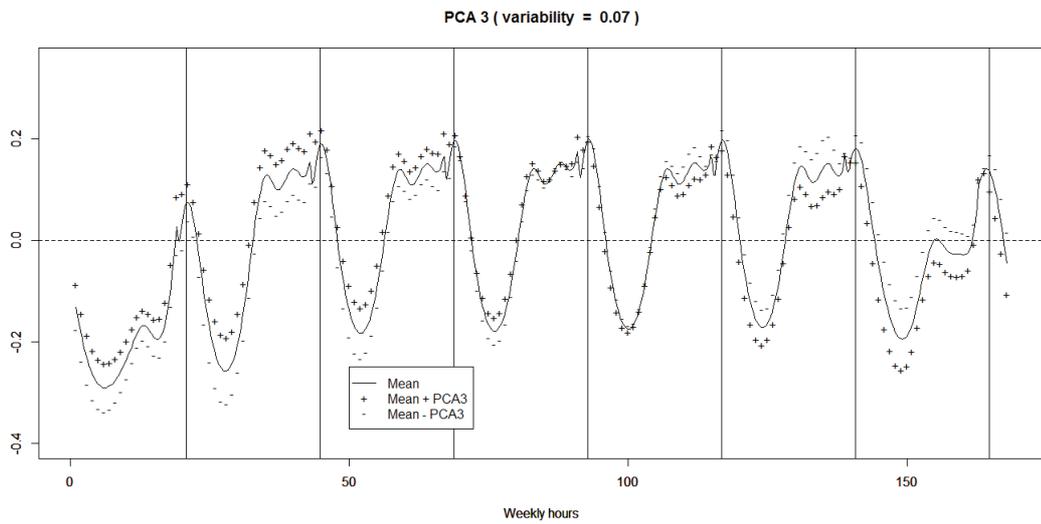


Figura 2-8: Curva média da carga semanal somada e subtraída da terceira componente principal. As linhas verticais marcam os dias da semana.

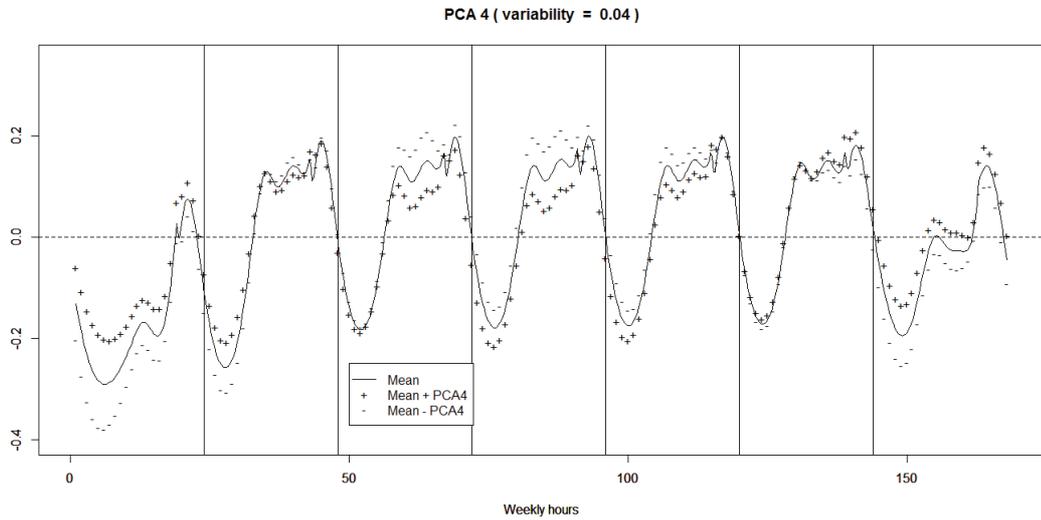


Figura 2-9: Curva média da carga semanal somada e subtraída da quarta componente principal. As linhas verticais marcam os dias da semana.

2.2.2 Temperatura

As observações da temperatura (em graus Celsius) são médias aritméticas ponderadas de medições realizadas em aeroportos localizados nas regiões sudeste e centro-oeste para o período 01 de janeiro de 2003 até 20 de janeiro de 2011. A figura (2-10) mostra o gráfico da média diária da temperatura. Por este gráfico, percebemos que a temperatura apresenta um óbvio padrão sazonal acentuado.

A resposta da carga à temperatura não ocorre de forma linear [Engle, Granger, Rice & Weiss (1986)]. Na figura (2-11), podemos observar a carga e a temperatura de dois dias escolhidos ao acaso e evidências de não linearidade da relação. A resposta de carga à temperatura é assimétrica, no sentido de que um grau que aumenta quando a temperatura é alta e um grau que aumenta quando a temperatura é baixa não tem o mesmo impacto sobre a carga [Valor, Meneu & Caselles (2001)]. A relação entre a carga e a temperatura é diferente para dias úteis e finais de semana [Smith (2000)]. Ela também pode depender da estação do ano [Hyde & Hodnett (1997)].

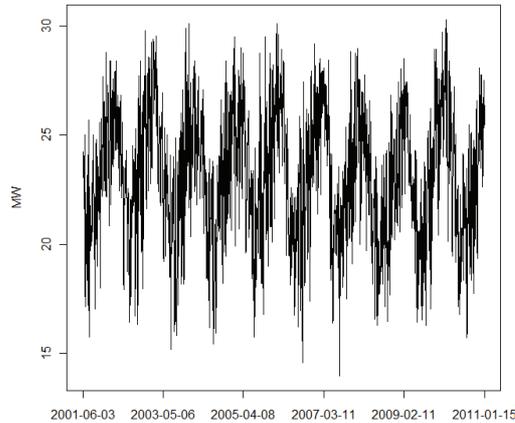


Figura 2-10: Média diária da temperatura

Diante dos possíveis tipos de relação entre a carga e a temperatura vamos avaliar três especificações diferentes. Iremos compará-las graficamente em termos de erro quadrático médio. A primeira delas é permitir que a temperatura observada hora a hora entre como um regressor funcional no modelo. Em geral, às 9:00 o comércio e os serviços iniciam suas atividades diárias, sendo assim, na segunda especificação, consideramos a temperatura observada neste horário como um regressor escalar¹ para explicar a carga. A terceira forma é amplamente utilizada, nela consideramos a relação entre a carga e a temperatura através das estatísticas denominadas HDD [heating degree days] e CDD [cooling degree days]. No Brasil, não é comum a utilização de equipamentos de calefação. Por isso, vamos considerar apenas a CDD para relacionar a carga à temperatura. O CDD é definido em termos de uma temperatura base [ou temperatura de conforto que é aquela a partir da qual é necessário o uso de equipamentos de resfriamento]. A temperatura base é aproximadamente 26° C. O

¹Mesmo que a temperatura seja um escalar o parâmetro a ela associado continua sendo funcional.

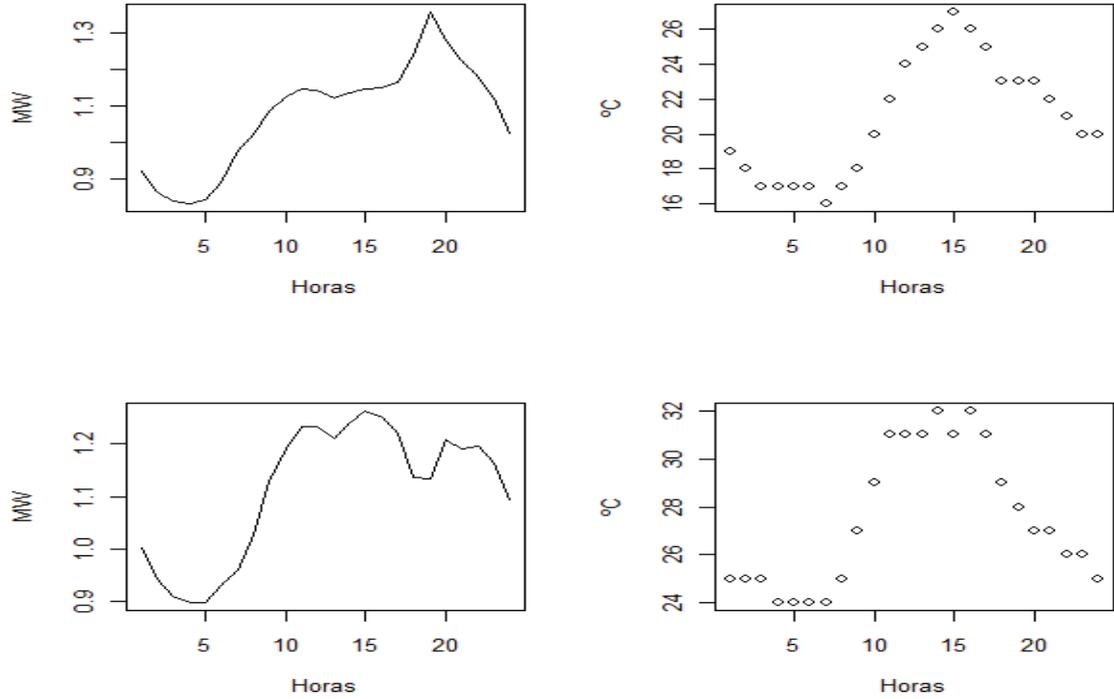


Figura 2-11: Comportamento da carga e da temperatura ao longo do dia para os dias 15/11/2002 e 08/08/2008

CDD é definido da seguinte maneira:

$$CDD = \begin{cases} 0 & \text{se } T_{\max} < T_{base} \\ \frac{T_{\max} - T_{base}}{4} & \text{se } \frac{T_{\max} + T_{\min}}{2} < T_{base} \\ \frac{T_{\max} - T_{base}}{2} - \frac{T_{base} - T_{\min}}{4} & \text{se } T_{\min} \leq T_{base} \\ \frac{T_{\max} - T_{\min}}{2} - T_{base} & \text{se } T_{\min} > T_{base} \end{cases}$$

em que T_{\max} e T_{\min} são as temperaturas máxima e mínima observadas, respectivamente, no dia.

2.3 Modelagem da carga elétrica horária

Nesta seção, apresentamos os modelos aqui denominados por: diário, semanal e anual, que foram estimados para a carga de energia elétrica horária. Além disso, fazemos uma breve revisão dos modelos de regressão não paramétrica para dados univariados e uma adaptação desses modelos para o caso funcional.

Conforme veremos em seções seguintes, o modelo para a evolução semanal da carga sugere que os parâmetros evoluem de forma contínua ao longo do ano. A fim de avaliar melhor o comportamento dos parâmetros ao longo do ano, especificamos modelos de regressão lineares locais [funcionais] para cada uma das semanas, baseados na ideia dos modelos de regressão não paramétrica para dados univariados. O comportamento dos parâmetros desses modelos, para cada uma das semanas, também sugere que eles evoluem ao longo do ano de forma suave.

Em todos os modelos que serão estimados neste trabalho, a componente autorregressiva será a carga defasada em 24 horas. Isto é, a carga do dia $d - 1$ na hora t será o regressor para a carga do dia d na hora t . Poderíamos, por exemplo, utilizar como regressor para a carga na hora t a carga da hora $t - 1$. Mas queremos manter o horizonte de previsão em 24 horas.

2.3.1 Modelo funcional diário

O modelo funcional diário pode ser escrito da seguinte forma:

$$E(Y_d(t)|\mathcal{G}_m) = \alpha_d(t) + \beta_d(t)Y_{d-1}(t) + \gamma_d(t)W_d(t) + D_d(t) \quad (2.2)$$

em que $d = 1, 2, \dots, n$, $\alpha_d(t) = \alpha_{d+7}(t)$, $\beta_d(t) = \beta_{d+7}(t)$, $\gamma_d(t) = \gamma_{d+7}(t)$ e $t \in [0, 24)$. A variável $W_d(\cdot)$ é a temperatura horária² e $Y_{d-1}(\cdot)$ é a carga do dia imediatamente anterior ao dia d . As variáveis $\alpha_d(\cdot)$ e $D_d(t)$ desempenham a mesma função que a constante e dummies de feriados em um modelo univariado de regressão, respectivamente. Temos também que

²A temperatura será escrita em termos de uma base de Fourier, como em Ramsay & Silverman (2010).

$G_m := \sigma(Y_{m-1}, X_m : m \leq d)$.

No modelo diário que aparece em (2.2), as variações sazonais que a carga apresenta são captadas apenas pela temperatura. Outras variáveis podem induzir variações sazonais na carga, por exemplo horas de luminosidade diária, e não aparecem em (2.2). Sendo assim, adicionamos à forma funcional acima duas componentes de seno e duas componentes de cosseno conforme (2.3), visando captar possíveis variações sazonais não induzidas por temperatura:

$$\begin{aligned}
 E(Y_d(t)|\mathcal{G}_m) &= \alpha_d(t) + \beta_d(t)Y_{d-1}(t) + \gamma_d(t)W_d(t) + D_d(t) + & (2.3) \\
 &\theta_{1d}(t) \cos\left(\frac{2\pi}{52} \times \left\lfloor \frac{d}{7} \right\rfloor\right) + \phi_{1d}(t) \text{sen}\left(\frac{2\pi}{52} \times \left\lfloor \frac{d}{7} \right\rfloor\right) + \\
 &\theta_{2d}(t) \cos\left(\frac{4\pi}{52} \times \left\lfloor \frac{d}{7} \right\rfloor\right) + \phi_{2d}(t) \text{sen}\left(\frac{4\pi}{52} \times \left\lfloor \frac{d}{7} \right\rfloor\right).
 \end{aligned}$$

em que $d = 1, \dots, n$ e $\lfloor \cdot \rfloor$ indica a função parte inteira.

Na figura (2.12), temos os coeficientes associados à carga defasada para os modelos dos domingos, das quartas-feiras e das sextas-feiras. Na seção de análise exploratória da base de dados, observamos que as curvas de carga das terças, quartas, quintas e sextas-feiras apresentavam um formato parecido entre si. Na figura (2.12) podemos observar que a forma de resposta da carga à carga defasada da quarta e da sexta-feira também são parecidas entre si quando comparadas com a forma do domingo.

Na figura (2.13), podemos observar o gráfico dos parâmetros associados à primeira cossenóide nos modelos dos sábados, das quartas-feiras e das sextas-feiras.

Na figura (2.14), temos o gráfico dos parâmetros associados à segunda senóide nos modelos dos sábados, das quartas-feiras e das sextas-feiras.

Nas figuras (2.15) e (2.16) temos os gráficos dos testes de permutação para os coeficientes da carga defasada, da temperatura, da dummy de feriados e das componentes de seno e cosseno dos modelos do domingo e da segunda-feira. Eles são conjuntamente significativos com 95% de confiança.

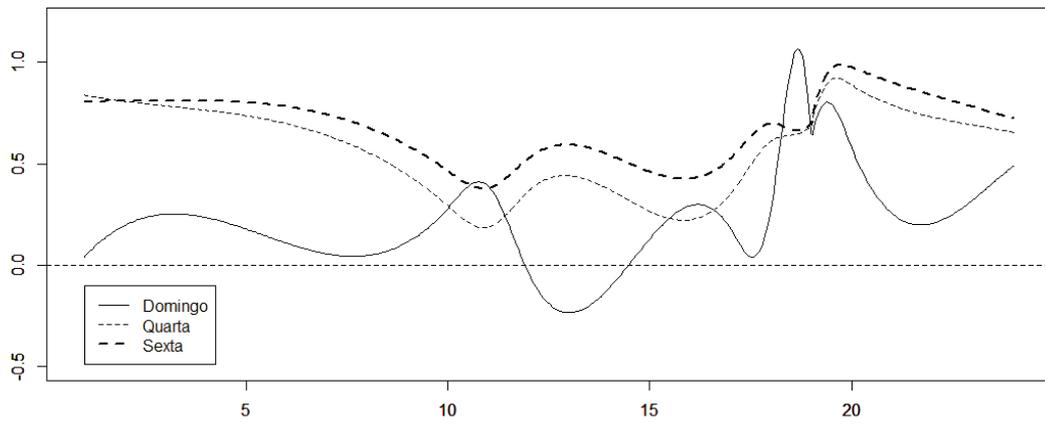


Figura 2-12: Coeficientes associados à carga defasada nos modelos dos domingos, quartas e sextas-feiras

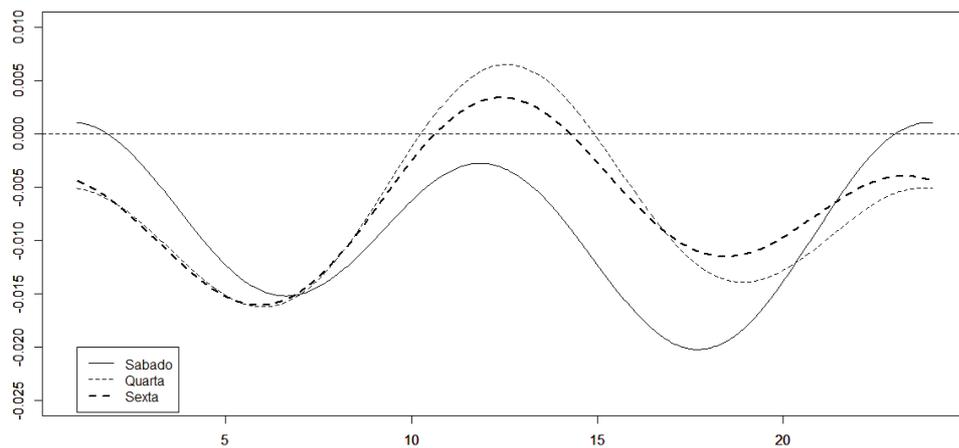


Figura 2-13: Coeficiente associado à primeira cossenóide nos modelos do sábados, quartas e sextas-feiras.

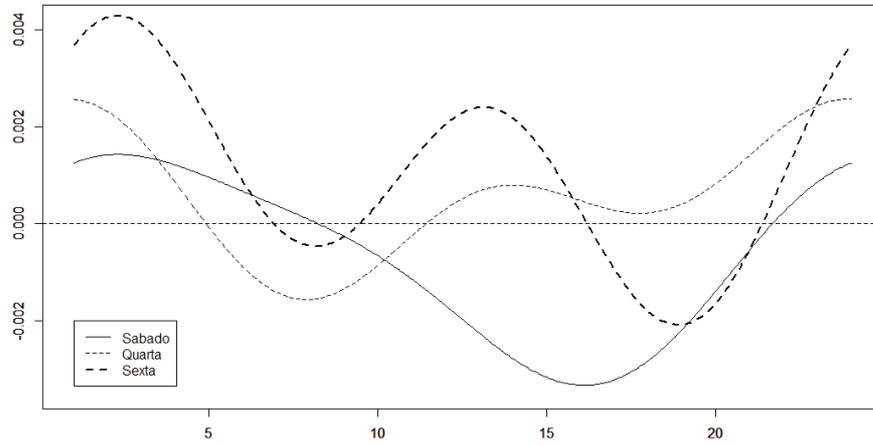


Figura 2.14: Coeficiente associado à segunda senóide nos modelos do sábados, quartas e sextas-feiras.

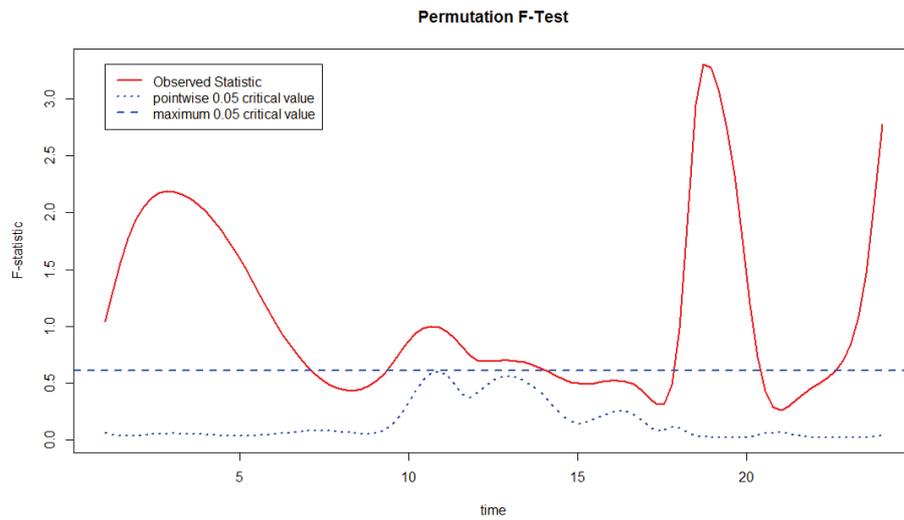


Figura 2.15: Teste de permutação para a significância conjunta dos coeficientes do modelo diário para o domingo

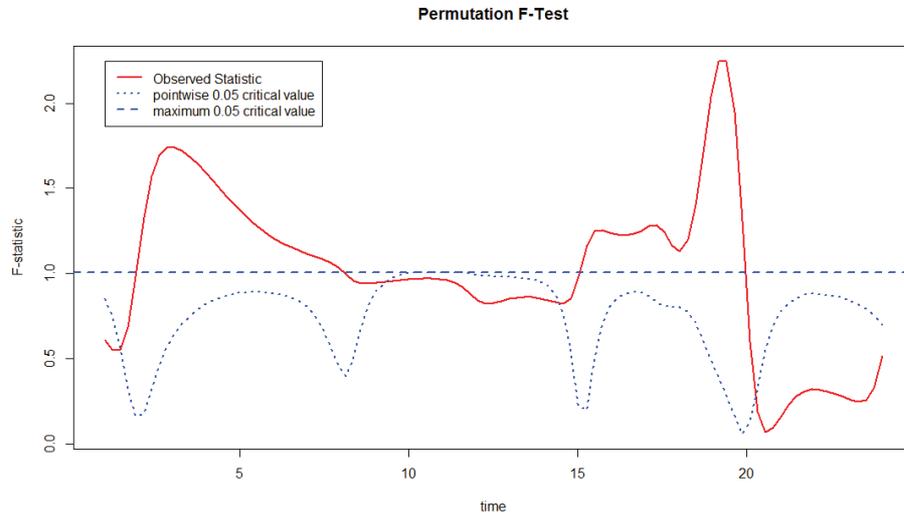


Figura 2-16: Teste de permutação para a significância conjunta dos coeficientes do modelo diário para a segunda-feira

O modelo diário apresenta bons resultados, em termos de raiz do erro quadrático médio, para as terças, quartas, quintas e sextas-feiras. Para indicar a raiz do erro quadrático médio utilizaremos a sigla RMSE [*Root Mean Squared Error*]. Os modelos para o sábado e o domingo apresentam RMSE's mais elevadas. Os maiores valores ocorrem nos momentos mais críticos que são os horários de pico de consumo. Os valores horários da RMSE para cada um dos dias da semana podem ser observados na tabela (2-19). Nessa tabela, os erros se referem a um modelo em que a informação da temperatura é aquela fornecida pela Cooling Degree Days e eles são muito próximos dos erros obtidos quando se considera a temperatura horária.

Especificamente em relação ao modelo da segunda-feira, temos que a curva de carga do domingo apresenta um formato bastante diferente do formato da curva da segunda-feira³. Diante disso, decidimos utilizar a média dos dias úteis da semana imediatamente

³ Isto pode ser verificado na seção de análise descritiva, particularmente na figura (2-4). Podemos observar que a carga da segunda-feira apresenta movimentos ao longo do dia que não são acompanhados pelos movimentos da carga do domingo.

anterior a cada segunda-feira como sua respectiva covariada em substituição à carga do domingo. Tal medida representa um relaxamento do modelo periódico funcional original. O novo modelo apresentou melhores resultados, em termos de RMSE. Sendo assim, para posteriores considerações, a média dos dias úteis da semana imediatamente anterior a cada segunda-feira será a covariada correspondente à parte autorregressiva do modelo da segunda-feira.

O modelo diário apresenta como potencial desvantagem o fato dos parâmetros não serem contínuos ao longo dos dias da semana. Pode ocorrer uma descontinuidade entre a previsão para o último minuto do domingo e o primeiro minuto da segunda, o último minuto da segunda e o primeiro minuto da terça-feira e assim por diante. Na figura (2.17), observamos o gráfico do valor previsto para o domingo 29/08/2012, a segunda-feira 30/08/2012, a terça-feira 31/08/2012 e a quarta-feira 01/09/2010. Na figura (2.18), temos a média das previsões feitas para todos os domingos, segundas, terças e quartas-feiras observadas na amostra. A descontinuidade não ocorre apenas em um caso isolado, ela persiste para a média das previsões. Tendo em vista tal descontinuidade e a necessidade de se avaliar novas formas de tratamento da sazonalidade que não a induzida por temperatura, passaremos descrever o modelo semanal.

Para fins de comparação, estimamos um modelo ARIMA $(0, 0, 0) \times (1, 0, 1)$. O polinômio autorregressivo sazonal é $(1 - \phi L^{24})$ e o polinômio sazonal de média móvel é $(1 - \theta L^{24})$. L é o operador de defasagem, isto é, $L^{24}x_t = x_{t-24}$ e $|\phi| < 1$. A base de dados é a mesma utilizada em (2.3). O polinômio autorregressivo aparece com zero em todas as suas entradas para que possamos manter a comparabilidade com o modelo funcional diário estimado. Em nossos modelos, temos o interesse de manter o horizonte de previsão em 24 horas. As RMSE's provenientes do modelo ARIMA sazonal estimado variam de 4% a 10% ao longo das horas do dia.

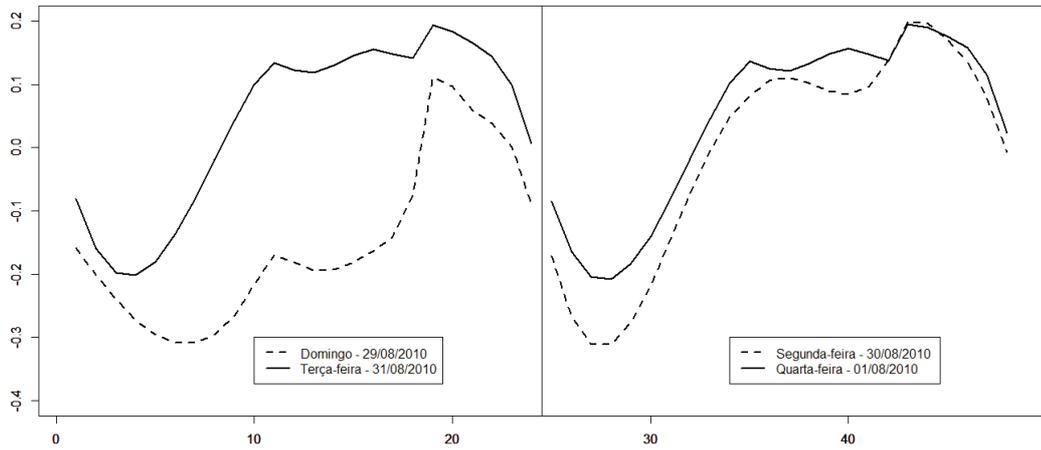


Figura 2-17: Valores previstos para o domingo 29/08/2012, a segunda-feira 30/08/2012, a terça-feira 31/08/2012 e a quarta-feira 01/09/2010.

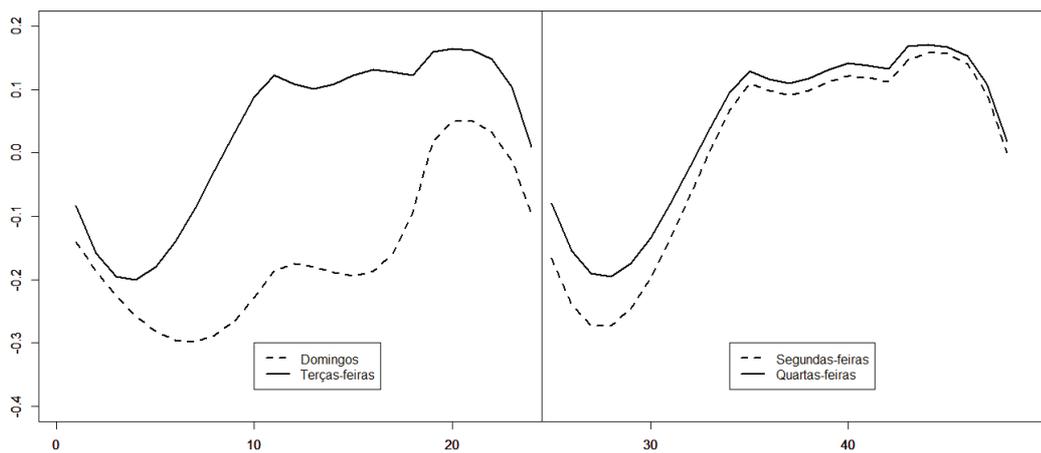


Figura 2-18: Média das previsões feitas para todos os domingos, segundas, terças e quartas-feiras observadas na amostra.

Domingo												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	4.29	4.05	3.89	3.73	3.64	3.7	3.95	4.35	4.81	5.31	5.72	5.58
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	5.46	5.41	5.43	5.47	5.47	5.62	6.17	5.9	5.47	4.98	4.56	4.13
Segunda												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	2.37	1.98	1.91	1.97	2.29	2.9	3.65	4.37	4.92	5.25	5.35	4.99
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	4.99	5.19	5.38	5.39	4.99	4.26	3.87	3.44	3.35	3.36	3.41	3.34
Terça												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	2.37	2.11	1.98	1.86	1.76	1.72	1.77	1.87	1.97	2.1	2.29	2.26
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	2.41	2.39	2.39	2.54	2.75	2.84	1.99	1.62	1.63	1.82	2.03	2.16
Quarta												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	2.09	1.9	1.84	1.78	1.66	1.63	1.69	1.82	1.96	2.14	2.34	2.34
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	2.61	2.84	3.07	3.3	3.4	3.28	2.37	1.7	1.7	1.99	2.28	2.5
Quinta												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	2.37	2.22	2.12	1.96	1.85	1.94	2.26	2.68	3.09	3.45	3.63	3.29
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	3.24	3.39	3.63	3.81	3.77	3.47	2.6	2.08	1.99	2.14	2.35	2.43
Sexta												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	2.18	1.95	1.88	1.81	1.76	1.79	1.92	2.1	2.27	2.47	2.69	2.55
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	2.65	2.85	3.08	3.25	3.29	3.07	2.38	1.96	1.9	2.1	2.4	2.75
Sábado												
Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	3.75	2.72	2.28	2.14	2.19	2.45	2.89	3.41	3.89	4.26	4.4	3.93
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	3.64	3.61	3.7	3.73	3.61	3.54	3.77	3.29	2.8	2.6	2.53	4.45

Figura 2-19: RMSE por hora e dia do modelo diário.

2.3.2 Modelo funcional semanal

No modelo semanal, o domínio das funções é $[0, 168)$. As primeiras 24 horas correspondem ao domingo, as 24 horas seguintes correspondem à segunda-feira e assim por diante. Defina:

$$G_{s,t} := \begin{cases} \sigma(\{Y_s(u); u \leq t - 24\}, Y_{s-1}, Y_{s-2}, \dots, X_s, \dots) & \text{se } t \in (24, 168] \\ \sigma(\{Y_{s-1}(u); u \leq t + 168 - 24\}, Y_{s-1}, Y_{s-2}, \dots, X_s, \dots) & \text{se } t \in (0, 24] \end{cases}$$

Iremos admitir que:

$$\begin{aligned} E(Y_s(t)|\mathcal{G}_{s,t}) &= \alpha(t) + \beta_1(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+}V(\vartheta) + \\ &\beta_2(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+}FMA(\vartheta) + \\ &\beta_3(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+}MJJ(\vartheta) + \\ &\beta_4(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+}ASO(\vartheta) + \gamma(t)W_s(t) + D_s(t) \end{aligned} \quad (2.4)$$

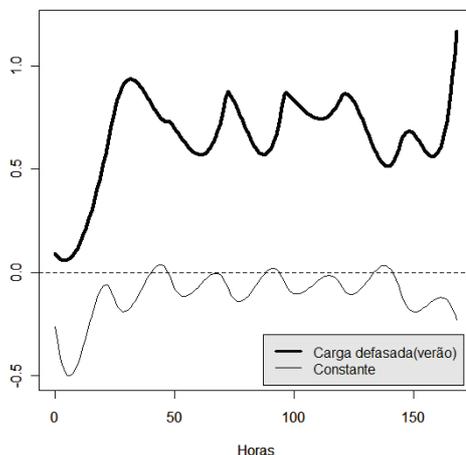


Figura 2·20: Parâmetros estimados para a carga defasada das semanas em que vale o horário de verão e a constante.

em que $s = 1, 2, \dots, N$, $\vartheta = 168 * \lfloor \frac{24}{t} \rfloor + t - 24$, $\lfloor \cdot \rfloor$ denota a função parte inteira e $(\cdot)_+$ denota a função parte positiva. $Y_{z,V}$ assume os valores da carga somente quando z indica uma semana na qual vale o horário de verão. Caso contrário, $Y_{z,V}$ assume valor zero. $Y_{z,FMA}$ assume os valores da carga somente quando z indica uma semana pertencente a fevereiro [e na qual não vale horário de verão], março e abril. $Y_{z,MJJ}$ segue a mesma regra para aquelas semanas pertencentes a maio, junho e julho. E por fim, $Y_{z,ASO}$ assume os valores da carga somente quando z indica uma semana pertencente a agosto, setembro e outubro [e na qual não vale horário de verão]. $W_s(t)$ é a temperatura. Novamente, $\alpha_s(\cdot)$ e $D_s(\cdot)$ desempenham a mesma função que uma constante e uma dummy de feriados nos modelos univariados de regressão.

Na figura (2·20), podemos observar o parâmetro funcional estimado para $Y_{z,V}(t - 24)$ e a constante $\alpha(t)$, que são contínuos ao longo da semana. E na figura (2·21) acessamos o gráfico do teste de permutação para significância conjunta dos coeficientes do modelo semanal. Os coeficientes são conjuntamente significativos com 95% de confiança.

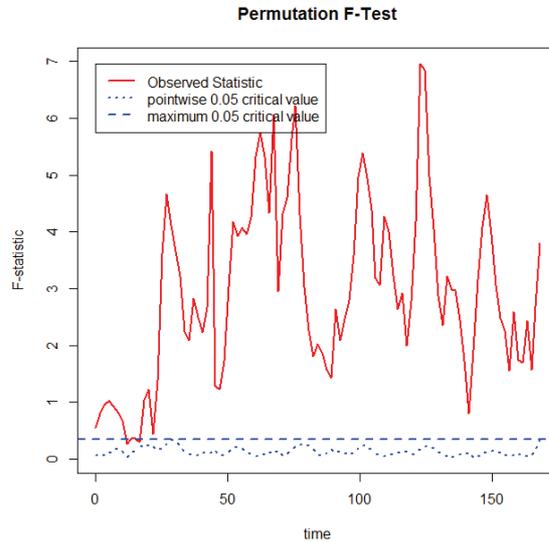


Figura 2-21: Teste de permutação para significância conjunta dos coeficientes do modelo semanal.

As RMSE's para o modelo semanal se encontram na tabela (2-22) e também na figura (2-23). O gráfico ajuda a localizar os dias da semana na tabela. Podemos observar que o modelo semanal gera RMSE's mais altas para os horários de pico do domingo e do sábado em relação aos outros dias da semana. O modelo chega a apresentar RMSE de 6% às 19 horas do domingo, enquanto ela fica em média em torno de 4% para este mesmo horário nas segundas, terças, quartas, quintas e sextas-feiras.

A figura (2-24) mostra o gráfico dos coeficientes da carga defasada estimados no modelo semanal. Podemos observar a existência de quatro padrões diferentes de resposta da carga à carga defasada. Apesar de diferentes, os padrões variam de forma suave ao longo do ano. Os parâmetros referentes a verão e FMA apresentam forma mais parecida entre si do que quando comparados aos parâmetros de verão e MJJ. Além disso, é desejável permitirmos que a resposta da carga às outras variáveis explicativas [temperatura, por exemplo] também varie ao longo das estações ou, mais que isso, ao longo dos meses. Sendo assim, se justifica a especificação de um modelo no qual a unidade de observação é a carga ao longo de um

Hora	1	2	3	4	5	6	7	8	9	10	11	12
RMSE	3,8	3,7	3,6	3,5	3,4	3,4	3,4	3,4	3,4	3,5	3,9	4,9
Hora	13	14	15	16	17	18	19	20	21	22	23	24
RMSE	4,3	3,8	4,3	4,7	4,7	4,8	6	5,2	5,5	5	3,9	3,2
Hora	25	26	27	28	29	30	31	32	33	34	35	36
RMSE	2,8	2,6	2,6	2,7	2,8	3	3	3,1	3,5	3,9	4	3,7
Hora	37	38	39	40	41	42	43	44	45	46	47	48
RMSE	3,4	3,7	3,9	4	3,8	3,4	3,6	3,7	3,2	2,9	2,8	2,8
Hora	49	50	51	52	53	54	55	56	57	58	59	60
RMSE	2,6	2,3	2,1	2,1	2,3	2,6	2,9	3,1	3,3	3,4	3,1	2,6
Hora	61	62	63	64	65	66	67	68	69	70	71	72
RMSE	3,2	2,8	2,8	3	3,1	3,1	3,3	3,6	2,2	1,9	2	2,3
Hora	73	74	75	76	77	78	79	80	81	82	83	84
RMSE	2,3	2,2	2	2	2,2	2,6	2,9	3,4	3,9	4,2	3,8	3
Hora	85	86	87	88	89	90	91	92	93	94	95	96
RMSE	3,4	3,3	3,3	3,5	3,5	3,4	3,8	3,9	2,3	2,1	2,5	2,8
Hora	97	98	99	100	101	102	103	104	105	106	107	108
RMSE	2,8	2,6	2,4	2,3	2,4	2,6	2,7	2,9	3,2	3,6	3,5	3,1
Hora	109	110	111	112	113	114	115	116	117	118	119	120
RMSE	2,9	3,1	3,3	3,5	3,6	3,7	4,1	4,2	2,7	2,5	2,7	2,8
Hora	121	122	123	124	125	126	127	128	129	130	131	132
RMSE	2,6	2,3	2	2,1	2,5	3	3,4	3,8	4,3	4,6	4,5	3,8
Hora	133	134	135	136	137	138	139	140	141	142	143	144
RMSE	3,6	3,9	4,1	4,1	4	4,1	5,4	4,7	3,6	3	2,6	2,6
Hora	145	146	147	148	149	150	151	152	153	154	155	156
RMSE	2,6	2,6	2,5	2,4	2,5	2,6	2,7	3,1	3,5	3,8	3,9	3,9
Hora	157	158	159	160	161	162	163	164	165	166	167	168
RMSE	3,5	3,5	4,3	5	4,9	4	5,7	5,7	3,7	3,1	3,8	3,8

Figura 2-22: RMSE's para cada hora da semana associadas ao modelo semanal.

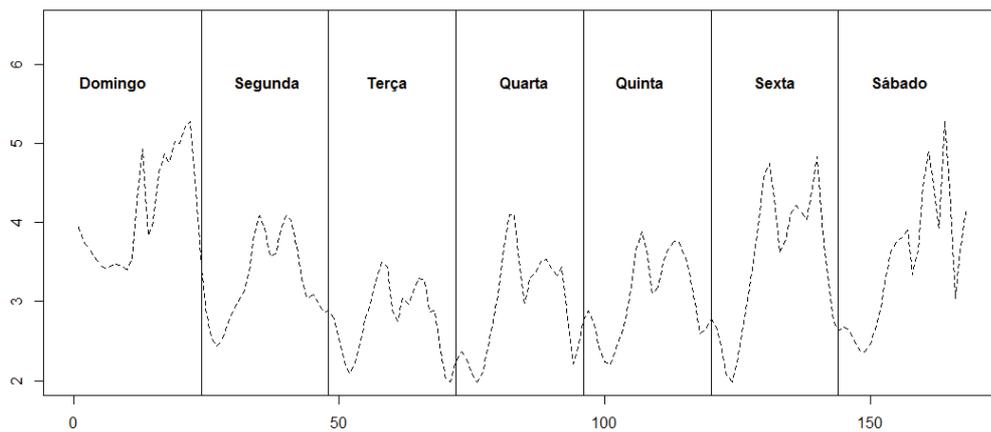


Figura 2-23: Grafico das RMSE's do modelo semanal.

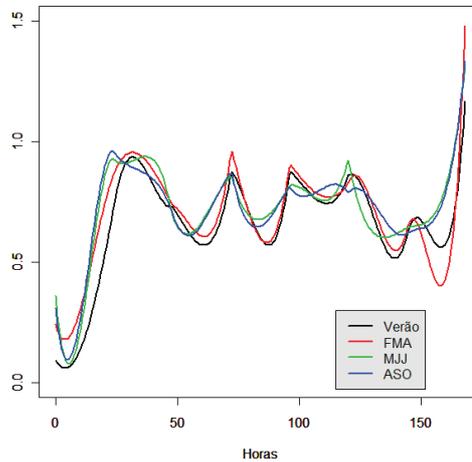


Figura 2-24: Coeficientes da carga defasada no modelo semanal

ano.

Para entender melhor como se dá a evolução dos parâmetros ao longo do ano vamos especificar modelos para cada semana. A ideia que fundamenta o desenvolvimento de tais modelos está baseada nos modelos tradicionais de regressão não paramétrica que passaremos a descrever.

2.3.3 Modelo de regressão não-paramétrica

Nesta subseção, descrevemos sumariamente a estimação não paramétrica de funções de regressão através de regressões locais. Optamos aqui pela parcimônia. Nossas definições e comentários foram escolhidos de modo a levar rapidamente aos modelos de regressão localmente lineares. Para maiores detalhes consulte Härdle (1990) [ver também, Györfi, Kohler, Krzyzak & Walk (2002) ou Wand & Jones (1995) ou Loader (1999)].

Sejam $Y \in L^2$ e X vetores aleatórios tomando valores em R^d e R^k , respectivamente. A

função de regressão de Y em X é definida como:

$$\begin{aligned} m & : \mathbb{R}^k \rightarrow \mathbb{R}^d \\ x & \mapsto E(Y|X = x) \end{aligned}$$

Modelos estatísticos onde a função de regressão é o objeto interesse são denominados modelos de regressão. Quando supomos:

$$m(x) = \mathbf{B}^\top x,$$

para alguma matriz de constantes $B_{k \times d}$ e dispomos de pares $(X_1, Y_1), \dots, (X_n, Y_n)$ tais que $(X_j, Y_j) \sim (X, Y)$, $j = 1, 2, \dots, n$, o modelo subjacente é dito de regressão linear (multivariado).

Modelos de regressão linear são exemplos de modelos paramétricos⁴ de regressão. Com os raros fenômenos sobre os quais estão disponíveis informação ou conhecimento suficientemente detalhados é legítimo lançarmos mão de modelos de regressão não-linear paramétricos [Bates & Watts (1988) ou Seber & Wild (2003)]. Não é assim na maioria das aplicações.

Uma alternativa importante aos modelos paramétricos de regressão não linear é fornecida pela estatística não-paramétrica através das regressões localmente lineares. A ideia é simples. Suponha que a função de regressão $m(\cdot)$ seja continuamente diferenciável. A expansão de Taylor em torno do ponto x_0 dá

$$m(x) = m(x_0) + m'(x_0) \cdot (x - x_0) + \|x - x_0\| \text{ resto}(x, x_0)$$

onde $m'(x_0) : \mathbb{R}^k \rightarrow \mathbb{R}^d$ é a derivada de m em x_0 e

$$\lim_{x \rightarrow x_0} \text{resto}(x, x_0) = 0.$$

Em outros termos: para valores de x próximos de x_0 , $m(x)$ é bem aproximada por

⁴A rigor semi-paramétricos.

$m(x_0) + m'(x_0) \cdot (x - x_0)$, uma transformação linear⁵. A aproximação acima justifica a adoção da linearidade local para a função de regressão. Resta precisar o conceito de proximidade e a formalização do estimador.

É conveniente aqui sermos um pouco mais formais: supor $(x_1, y_1), \dots, (x_n, y_n)$ realizações de (X, Y) . Iremos estimar a função de regressão num ponto x_0 , arbitrário, considerando somente os pontos em $S(x_0) := \{(x_j, y_j) : x_j \in N(x_0)\}$, onde $N(x_0)$ é uma vizinhança de x_0 ⁶. Para isso obtemos estimativa de mínimos quadrados ponderados do seguinte modelo auxiliar de regressão linear:

$$y_j = \alpha + \mathbf{B}^\top (x_j - x_0) + \varepsilon_j$$

somente para as observações em $S(x_0)$. A estimativa da função de regressão em x_0 é dada por $\hat{m}(x_0) := \hat{\alpha}$.

Os pesos associados aos estimadores de mínimos quadrados ponderados acima são regidos pelo seguinte princípio: quanto mais distante de x_0 estiver x_j , menor o peso a ser atribuído à observação (x_j, y_j) . É claro que esse princípio traduz a ideia local do estimador. A operacionalização desse princípio é facilmente conduzida com o auxílio de uma função integrável $W : R^k \rightarrow [0, \infty)$. O peso associado à observação j é dado por

$$\omega_{0j} := \frac{1}{b} W \left(\frac{1}{b} (x_j - x_0) \right),$$

onde $b > 0$ é quantidade denominada de largura de banda ou parâmetro de suavização. O caráter local fica mais evidenciado quando W é escolhido de modo que $W(x)$ seja função decrescente de $|x|$. Uma escolha popular de W : a densidade de uma normal multivariada com valor esperado nulo.

Relativamente às vizinhanças existem, grosso modo, dois tipos: vizinhanças métricas e vizinhanças ordinais:

⁵Mais precisamente uma transformação afim.

⁶Nada impede que $N(x_0) = \{x_1, \dots, x_n\}$.

1. Fazem parte de uma vizinhança métrica de x_0 todos os pontos (x_j, y_j) para os quais a distância entre x_0 e x_j for menor que um valor pré-estabelecido.
2. Fazem parte de uma vizinhança ordinal de x_0 todos os pontos (x_j, y_j) para os quais $\|x_j - x_0\| \leq \|x_{(\gamma n)} - x_0\|$, onde $\gamma \in (0, 1)$ é uma fração pré estabelecida e

$$\|x_{(1)} - x_0\| \leq \|x_{(2)} - x_0\| \leq \dots \|x_{(n)} - x_0\|$$

é o conjunto $\{\|x_1 - x_0\|, \dots, \|x_n - x_0\|\}$ ordenado.

A fim de explorarmos a possível sazonalidade nos coeficientes de nossos modelos semanais conduzimos regressões locais, uma para cada semana. Replicamos os princípios, acima apresentados, que governam as metodologias tradicionais de regressões não paramétricas para o contexto funcional com uma diferença importante. Aqui, o peso a ser atribuído à semana j vai ser definido pela distância entre o número da semana sendo modelada⁷ e o número no ano correspondente à família j . Dessa forma, as vizinhanças são definidas pela ordem das semanas e não pelos valores da temperatura, por exemplo.

Em termos mais práticos, temos: no modelo de regressão linear local da semana 1, a própria semana 1 recebe o maior de todos os pesos, as semanas 2 e 52 [do ano imediatamente anterior] recebem os segundos maiores pesos, as semanas 3 e 51 [do ano imediatamente anterior] recebem os terceiros maiores pesos e assim por diante até que as semanas 17 e 36 [do ano imediatamente anterior] são as últimas a receberem pesos diferentes de zero. No modelo da semana 2, essa recebe o maior de todos os pesos, as semanas 1 e 3 recebem os segundos maiores pesos, as semanas 52 [do ano imediatamente anterior] e 4 recebem os terceiros maiores pesos e assim por diante até que as semanas 18 e 37 [do ano imediatamente anterior] são as últimas a receberem pesos diferentes de zero.

O modelo da semana 1 pode ser escrito da seguinte forma:

$$y_1(t) = \alpha(t) + \beta_1(t)y_1(t - 24) + \gamma_1(t)x_1(t) + \varepsilon_1(t)$$

⁷Bem entendido: número= ordem da semana no ano. Valores possíveis: 1 a 52.

em que $\alpha(t)$ desempenha o papel da constante, $\beta_1(t)$ é o coeficiente da carga defasada, $\gamma_1(t)$ é o coeficiente da temperatura e $\varepsilon_1(t)$ é o termo de erro.

Na figura (2.25), temos os pesos para algumas semanas. Esses pesos foram gerados pela seguinte função⁸:

$$\left(1 - \left(\frac{d_{i,j}}{18}\right)^2\right)^2$$

em que:

$$d_{ij} = d(i-j) = \begin{cases} |i-j| & , \text{ se } |i-j| \leq 18 \\ 18 & , \text{ se } |i-j| \in [18, 26] \end{cases}$$

e

$$d((i-j) + 52) = d(i-j)$$

Na figura (2.26), exibimos a constante estimada para um conjunto de semanas. Podemos perceber que as curvas referentes às semanas 2 e 4, que são aquelas em que vale o horário de verão, são muito próximas entre si. Na sequência, temos as constantes das semanas 26 e 34. Apesar de distintas, percebemos que elas são mais semelhantes entre si do que quando comparadas com as primeiras. As semanas 50 e 51 voltam a ter um comportamento parecido com aquele observado nas semanas 2 e 4. Outra característica que podemos observar é que os picos da constante [que correspondem aos picos de consumo ao longo da semana] ocorrem mais cedo no inverno do que nas semanas em que vale o horário de verão.

Na figura (2.27), observamos o coeficiente da carga defasada. Podemos perceber a formação de três grupos de coeficientes. Os coeficientes pertencentes aos dois grupos

⁸Este tipo de peso é utilizado no método de estatística não paramétrica LOESS (*Locally weighted scatter plot smoothing*), para maiores informações sobre este método veja Fan & Gijbels (1996).

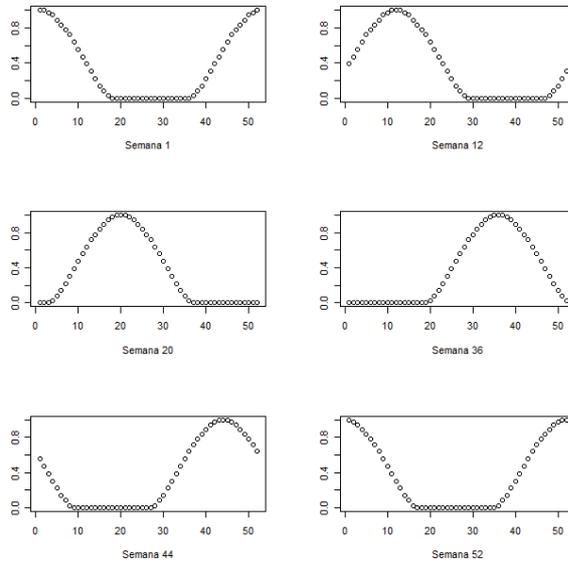


Figura 2.25: Ilustração dos pesos utilizados na estimação dos modelos semanais.

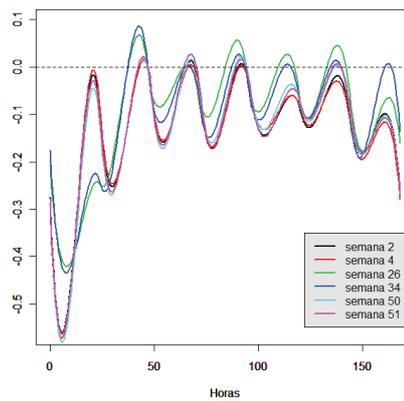


Figura 2.26: Constante estimada nos modelos das semanas 2,4,26, 34,50 e 51

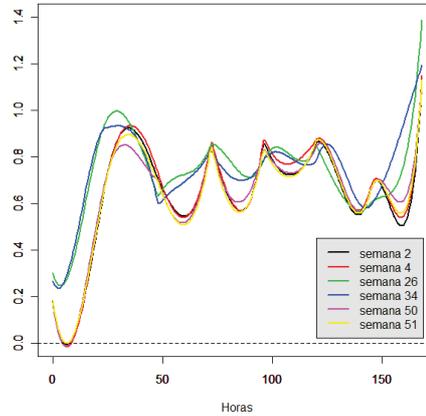


Figura 2.27: Coeficiente estimado para a carga defasada nos modelos da semanas 2,4,26,34,50 e 51.

formados pelas semanas 2, 4 e 50, 51 apresentam um formato mais próximo entre si, quando comparados com os coeficientes das semanas 26 e 34.

O modelo não paramétrico, no qual cada observação da variável dependente é a função carga de uma semana, indica que a função que desempenha o papel da constante, o coeficiente da carga defasada e da temperatura variam ao longo do ano. Essa variação ocorre de forma suave. Decidimos então desenvolver um modelo anual não paramétrico, em que a unidade de observação é a carga de um ano.

2.3.4 Modelo anual

No modelo baseado na evolução anual das curvas de carga, temos uma redução drástica do número de observações. Enquanto, no modelo diário temos 2920 dias observados, no modelo anual temos oito anos observados. Ressaltamos, porém que com as oito observações é possível estimar os parâmetros do modelo. Isso porque, eles devem satisfazer uma série de restrições que lhes conferem estrutura e regularidade. Por exemplo, a função que desempenha o papel da constante será escrita como combinação linear de uma base de *splines*. Dessa forma, a constante [que é combinação linear dos splines] deve satisfazer as

mesmas condições que estes. Ou seja, a função que desempenha o papel da constante será contínua e não só isso, terá primeira derivada contínua [permitimos que a segunda derivada seja descontínua no horário de pico quando ele ocorre no início da noite].

Além da estrutura herdada da base de splines, exigiremos outras estruturas dos coeficientes. A carga exhibe, por exemplo, periodicidade semanal. Isto é, a cada sete dias, temos a recorrência de um determinado padrão de consumo. Sendo assim, é desejável que o parâmetro se adapte ao padrão imposto pelo dia da semana. Isto será feito através do uso de operadores de aceleração harmônica que serão descritos nas seções seguintes.

Quando estamos lidando com dados funcionais uma possível fonte de variabilidade é a associada a flutuações na amplitude e na fase. Para entender melhor considere a figura (2.28). Podemos observar que características importantes das curvas como, por exemplo, os pontos nos quais a derivada primeira é zero, isto é, os máximos e mínimos da função não estão alinhados. Esse tipo de variabilidade das curvas é denominado variação de fase. Um outro tipo de variabilidade, denominado variação de amplitude, corresponde a deslocamentos verticais em alguns intervalos do domínio das curvas. Na figura (2.28), observamos deslocamentos verticais nas regiões próximas aos pontos críticos das curvas.

Em certos tipos de modelagem, no nosso caso modelos de regressão, as variações de amplitude e fase podem implicar algum tipo de viés. Uma técnica para lidar com essas variabilidades antes da modelagem é denominada de registro. O registro consiste em transformar o argumento t das funções assim como os seus valores $x(t)$. Na seção seguinte, deixamos claro como as evoluções anuais da carga envolvem variações de fase e amplitude e as suas implicações para os modelos funcionais.

O Registro das curvas de carga

Considere o seguinte modelo funcional:

$$Y_n(t) = \alpha(t) + \beta(t)X_n(t) + \varepsilon_n(t)$$

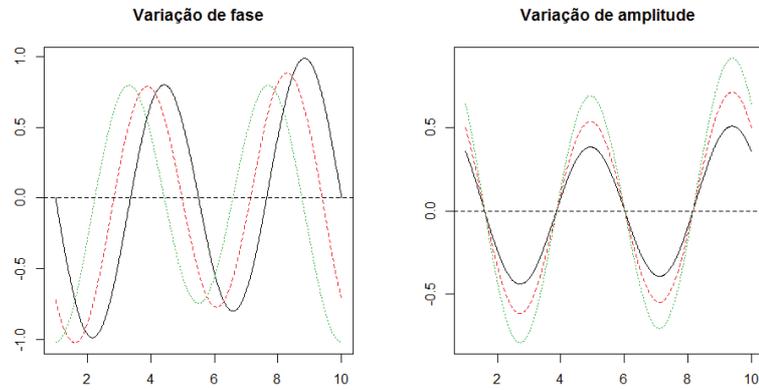


Figura 2-28: Exemplos de variações de fase e amplitude

em que $Y_n(t)$ é a carga observada no ano n para a hora t , $n = 1, \dots, N$, $t \in (0, 365 \times 24)$, $X_n(t)$ é uma variável explicativa qualquer para a carga e $\varepsilon_n(t)$ é um termo de erro. Suponha que dentre as curvas observadas estejam as curvas para os anos de 2006 e 2007. O primeiro dia do ano de 2006 foi um domingo enquanto o primeiro dia do ano de 2007 foi uma segunda-feira. Na estimação de $\alpha(t)$ e $\beta(t)$ para $t = 15$ contribuem tanto $Y_{2006}(15)$, que é a carga observada para às 15 horas de um domingo quanto $Y_{2007}(15)$ que é a carga observada para às 15 horas de uma segunda-feira. Na análise exploratória da base de dados vimos que o dia da semana tem impacto sobre a curva de carga e assim $Y_{2006}(15)$ e $Y_{2007}(15)$ devem ser comparados com certo cuidado em $t = 15$, porque eles exibem comportamentos diferentes.

Os argumentos anteriores são entendidos de maneira mais fácil quando nos atemos à estimação da função $\alpha(t)$ que desempenha o mesmo papel que a constante nos modelos tradicionais de regressão. Sem perda de generalidade e ainda visando o entendimento dos argumentos que vamos utilizar, suponha que $X_n(\cdot)$'s são expressas como desvios em relação

à sua média⁹. Assim como no caso tradicional $\alpha(t)$ será dada pela média aritmética das Y'_n s observadas na hora t . Esta média está sendo tomada para valores correspondentes a domingos, segundas-feiras, terças-feiras e assim por diante. Como as curvas de carga das terças, quartas, quintas e sextas-feiras apresentam um formato mais parecido entre si, quando comparadas aos sábados e domingos, o estimador de $\alpha(t)$ tenderá a ter um comportamento mais parecido com as curvas das terças, quartas, quintas e sextas-feiras. O que caracteriza um tipo de viés. Os sábados e domingos apresentariam um pior ajuste.

Dentre as possíveis técnicas [ver Ramsay & Silverman (2010)] para lidar com o viés descrito no paragrafo anterior, nós escolhemos o registro de deslocamento [*shift registration*] que consiste basicamente em realizar um alinhamento das funções na escala do tempo.

Considere as funções $\{Y_1, \dots, Y_n\}$ com domínio $[a, b]$ e $[A, B] \supset [a, b]$ o intervalo no qual as funções Y'_i s serão registradas. Para uma determinada curva de carga Y_i queremos encontrar um parâmetro δ_i tal que:

$$\tilde{Y}_i(t) = Y_i(t + \delta_i)$$

em que o parâmetro δ_i é escolhido de forma a alinhar as curvas $\{Y_1, \dots, Y_N\}$.

No caso da carga, nós conhecemos os valores de δ_i . As observações da carga são para o anos de 2003 a 2010. O ano de 2006 começa em um domingo e será o ano tal que $\delta_i = 0$. O ano de 2003 começa em uma quarta-feira, fazendo $\delta_{2003} = 96$, teremos que $\tilde{Y}_{2003}(1)$ e $\tilde{Y}_{2006}(1)$ são as primeiras horas de dois domingos. Os valores de δ_i para os outros anos são definidos da seguinte forma: $\delta_{2004} = 72$, $\delta_{2005} = 24$, $\delta_{2007} = 144$, $\delta_{2008} = 120$, $\delta_{2009} = 72$ e $\delta_{2010} = 48$. Para $t \in ((365 \times 24) - 144, 365 \times 24)$, nos definimos $\tilde{Y}_i(t) = Y_{i+1}(t + \delta_i)$. Para os anos de 2004 e 2008 as horas as últimas 24 horas serão excluídas.

Sabemos que o registro, através de deslocamento das funções, faz com que o primeiro de dia de um ano não esteja alinhado com o primeiro dia do ano seguinte e assim por diante. É fácil ver que o desalinhamento máximo entre dois dias de dois anos tomados ao acaso é

⁹bem entendido: $X_j(t) = x_j(t) - \left[\frac{x_1(t) + \dots + x_n(t)}{n} \right]$

de sete dias.

A grande influência dos dias do ano sobre o comportamento das curvas de carga ocorre em decorrência dos feriados fixos. O efeito dos feriados no formato das curvas de carga será controlado através do uso de variáveis *dummy*.

Afora o efeito feriado, importa mais ser terça-feira de verão do que ser a terça-feira que é o sétimo ou o décimo-quarto dia do ano. Evidentemente, o registro pode estar gerando algum tipo de viés nas estimativas dos parâmetros dos modelos funcionais. Acreditamos, porém que este viés seja menor do que aquele gerado pelo desalinhamento dos dias da semana. Isto porque as curvas de carga apresentam inércia e, portanto não ocorrem mudanças bruscas de padrão no intervalo de sete dias. Evidentemente, podem ocorrer mudanças de comportamento induzidas por chegada de frentes frias, por exemplo, mas essas mudanças estão controladas pela temperatura.

Operadores de aceleração harmônica

Em alguns casos de regressão funcional os parâmetros estimados podem exibir excesso de variabilidade local, Ramsay & Silverman (2010). Uma forma de corrigir tal problema, conforme apresentado no capítulo 1, é através da inclusão na soma de quadrados dos resíduos de um termo que penalize a o excesso de variabilidade. Dessa maneira os parâmetros [funcionais] seriam os argumentos que solucionam o problema abaixo:

$$\min \left\{ \int_I \|y_j(t) - \beta(t)X_j(t)\|^2 + \lambda \int_I [\beta(t)]^2 dt : \beta(t) \in C_I^2 \right\} \quad (2.5)$$

A constante λ é o parâmetro de suavização. Quanto maior o valor de λ mais suaves serão as funções estimadas. Em alguns casos é necessário corrigir o possível excesso de variabilidade local e também induzir uma determinada periodicidade nos parâmetros. Isto pode ser feito através dos operadores de aceleração harmônica, ver Ramsay & Dalzell (1991). Mais formalmente, considere a função $g(t) = \cos(\frac{2\pi t}{7}) + \text{sen}(\frac{2\pi t}{7})$ que é periódica com período 7 e é também solução da equação diferencial $L_1 g = 0$, em que $L_1 g = (\frac{2\pi}{7})^2 D^2 g + g$. O símbolo $D^n g$ indica n-ésima derivada de g em relação a t . Considere a seguinte soma de

quadrados penalizada:

$$SQP = \int_I \|Y(s) - \alpha(s) - \beta(s)X(s)\|^2 ds + \lambda \left[\int_I (L_1\beta(t))^2 dt \right].$$

O valor mínimo do segundo termo da SQP é atingido quando $L_1\beta = 0$. Se fazemos $\lambda \rightarrow \infty$, então a SQP assumirá valor mínimo em $L_1\beta = 0$, cuja solução é exatamente $\beta(t) = \cos(\frac{2\pi t}{7}) + \text{sen}(\frac{2\pi t}{7})$. Assim na soma de quadrados acima, penalizamos [de acordo com o tamanho do λ] possíveis desvios do comportamento senoidal descrito por $\cos(\frac{2\pi t}{7}) + \text{sen}(\frac{2\pi t}{7})$ na estimação da função $\beta(t)$. Com essa medida induzimos uma periodicidade no parâmetro estimado visando capturar a periodicidade semanal dos dados da carga.

A carga exibe também periodicidade anual induzida basicamente por questões climáticas. Podemos permitir que os parâmetros do modelo se adaptem de forma a captar essa periodicidade. Para isso, considere o operador diferencial $L_2 = (\frac{365}{2\pi})^2 D^2 + 1$ e a composição $L_2 \circ L_1 = [(\frac{365}{2\pi})^2] [(\frac{7}{2\pi})^2] D^4 + [(\frac{365}{2\pi})^2] + [(\frac{7}{2\pi})^2] D^2 + 1$. A equação $L^*\beta = 0$, em que o operador $L^* = L_2 \circ L_1$, tem como uma possível solução:

$$\cos(\frac{2}{365}\pi t) + \cos(\frac{2}{7}\pi t)$$

Na figura (2.29), fica claro que solução da equação $L^*\beta = 0$ é capaz de captar tanto os ciclo anual e semanal da carga. Podemos exigir também um comportamento aproximadamente senoidal da função $\alpha(t)$ bastando para isso acrescentar, na SQP , um termo de penalização para a função $\alpha(t)$.

Um exemplo do uso de operadores de aceleração harmônica pode ser encontrado em Ramsay & Silverman (2010). Esses dois autores utilizam o operador:

$$L\beta = \left(\frac{2\pi}{365}\right)^2 D\beta + D^3\beta$$

em que β é o coeficiente funcional de um modelo para explicar o log da precipitação anual de chuva através da temperatura diária. Os dados são para diversas estações meteorológicas situadas em regiões do Canadá.

Forma funcional do modelo anual

A fim de formalizar o modelo funcional anual defina:

$$\mathcal{G}_{m,t} := \begin{cases} \sigma(\{Y_m(u); u \leq t - 24\}, Y_{m-1}, Y_{m-2}, \dots, W_m, \dots) & \text{se } t \in (24, 8760] \\ \sigma(\{Y_{m-1}(u); u \leq t + 8760 - 24\}, Y_{m-1}, Y_{m-2}, \dots, W_m, \dots) & \text{se } t \in (0, 24] \end{cases}$$

escrevemos:

$$E(Y_m(t)|\mathcal{G}_{m,t}) = \alpha(t) + \beta_1(t)\tilde{Y}_{\tilde{m}} + \gamma(t)W_m(t) + D_m(t)$$

em que,

$$\begin{aligned} \tilde{Y}_{\tilde{m}} &= Y_m(t - 24) & \text{if } t \in (24, 8760] \\ \tilde{Y}_{\tilde{m}} &= Y_{m-1}(8760 + t - 24) & \text{if } t \in (0, 24] \end{aligned}$$

e $m = 1, \dots, N$, $t \in [0, 8760)$; 8760 é o número de horas observadas ao longo dos 365 dias do ano. A variável $\tilde{Y}_{\tilde{m}}$ representa a carga corrente defasada em 24 horas. A função $W_m(\cdot)$ é a temperatura. Novamente, $\alpha(\cdot)$ e $D_m(\cdot)$ desempenham o mesmo papel que a constante e dummies para feriados nos modelos tradicionais de regressão.

Na figura (2-30), temos o teste de permutação para o coeficientes estimados para o modelo anual. Na figura (2-31), podemos observar o boxplot dos erros provenientes do modelo anual para algumas horas de cada um dos dias úteis observados ao longo do ano: 4, 5, 12, 13, 17, 18, 19 e 20. Podemos observar também o *boxplot* das RMSE's para as mesmas horas nos finais de semana. As horas 4, 5, 12, 13, 17, 18, 19 e 20 foram escolhidas por apresentarem características importantes. Entre 4 e 5 horas o consumo de energia elétrica começa a aumentar em decorrência do início das atividades diárias. As horas 12 e 13 marcam algumas variações na carga em decorrência do horário do almoço. As horas 17, 18, 19 e 20 apresentam grande variabilidade da carga em função da ocorrência do horário de pico.

Nas figuras (2-32) e (2-33), podemos observar os boxplots das RMSE's dos modelos diário e semanal, separados por dias úteis e finais de semana. As RMSE's do modelo diário associadas às horas 4, 5, 12, 13, 17, 18, 19 e 20 apresentam distribuições cujas medianas ficam

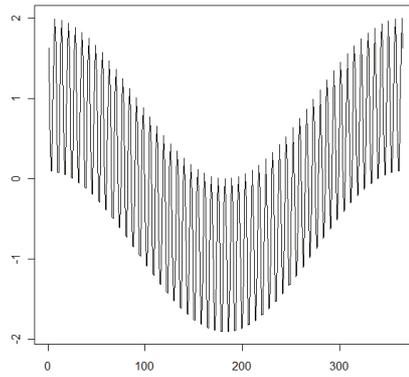


Figura 2-29: Gráfico da função $\cos(\frac{2}{365}\pi t) + \cos(\frac{2}{7}\pi t)$ avaliada nos inteiros.

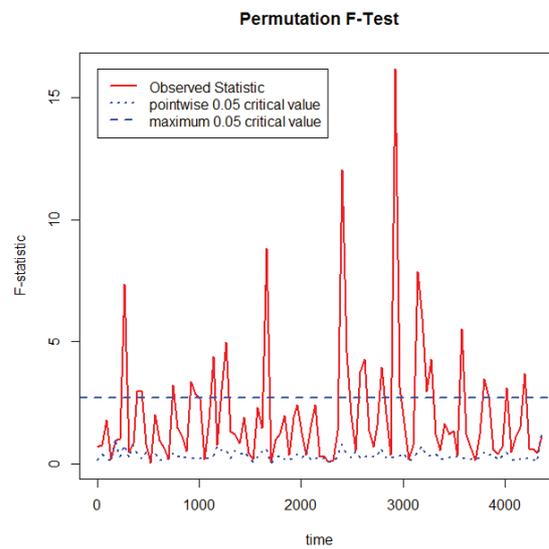


Figura 2-30: Teste de permutação para os coeficientes estimados no modelo anual.

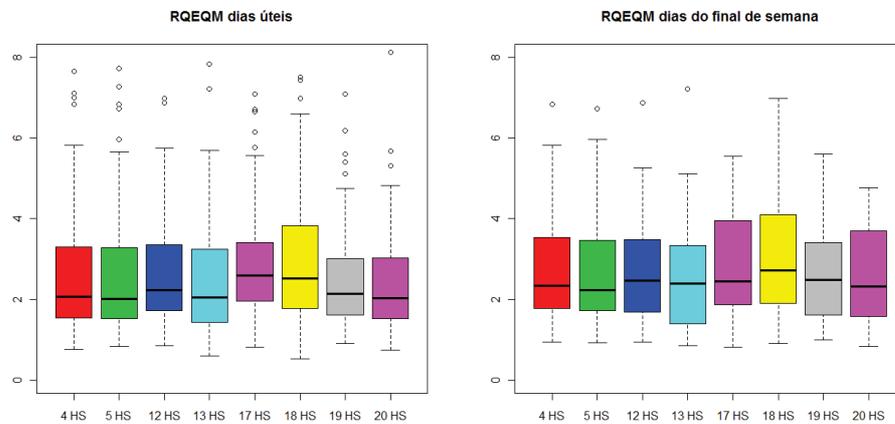


Figura 2-31: *Boxplot* da raiz quadrada do erro quadrático médio do modelo anual para algumas horas selecionadas do dia.

próximas a 2% para as horas de carga baixa [horas da madrugada] e entre 3 e 4% para as horas restantes. Nos finais de semana, as distribuições apresentam medianas em torno de 3% para as horas de carga baixa e acima de 5% para as horas restantes. As RMSE's do modelo semanal apresentam distribuições cujas medianas ficam entre 3 e 5% para os dias úteis. As RMSE's associadas aos finais de semana apresentam medianas semelhantes. As RMSE's do modelo anual apresentam distribuições cujas medianas ficam entre 2 e 3% tanto para os dias úteis quanto para os finais de semana. Em comparação aos modelos diário e semanal, o modelo anual rende resultados menos difusos. Isto é, as distribuições das RMSE's nos finais de semana e nos dias uteis são mais parecidas entre si. Além disso, as medianas são menores do que aquelas associadas aos outros dois modelos.

Na figura (2-34), temos o gráfico da constante estimada no modelo semanal. Na mesma figura, encontra-se a constante estimada no modelo anual avaliada para as horas entre 1 e 24 e entre 3865 e 4033. No modelo semanal, entram juntamente na estimativa da constante semanas pertencentes a inverno e verão. A forma típica das curvas de verão é diferente das formas típicas das curvas de inverno. A constante [que é uma espécie de média] fica com uma forma intermediária entre as formas do inverno e do verão.

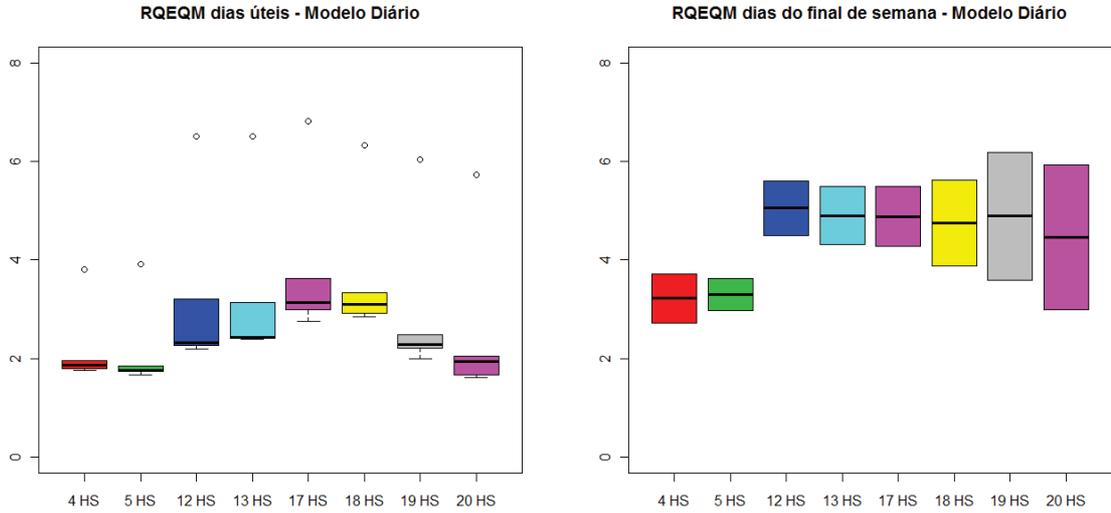


Figura 2-32: *BoxPlot* da RMSE para algumas horas selecionadas nos dias úteis e nos finais de semana de acordo com o modelo diário.

No modelo anual, para a primeira semana do ano, entram na estimativa da constante apenas curvas pertencentes ao verão. Para a 24^a semana do ano entram na estimativa apenas semanas pertencentes ao inverno. Isto leva a identificação de dois movimentos. O primeiro deles é um deslocamento para baixo da constante estimada para a 24^a semana do ano em comparação à estimativa para a 1^a semana do ano. O segundo movimento é uma ampliação da diferença entre as duas estimativas à medida que se aproxima do final da semana. Esses dois movimentos foram captados pela primeira e pela terceira componentes principais [cf. seção 2.2.1]. Elas estavam relacionadas ao nível e à inclinação das curvas de carga. Observamos também uma ligeira mudança de escala temporal na constante estimada para a 1^a semana e a 24^a. Esta característica foi captada pela 2^a componente principal [cf. seção 2.2.1]. Concluímos que as estimativas para a constante, fornecidas pelo modelo anual, refletem melhor as expectativas geradas pela análise de componentes principais funcionais.

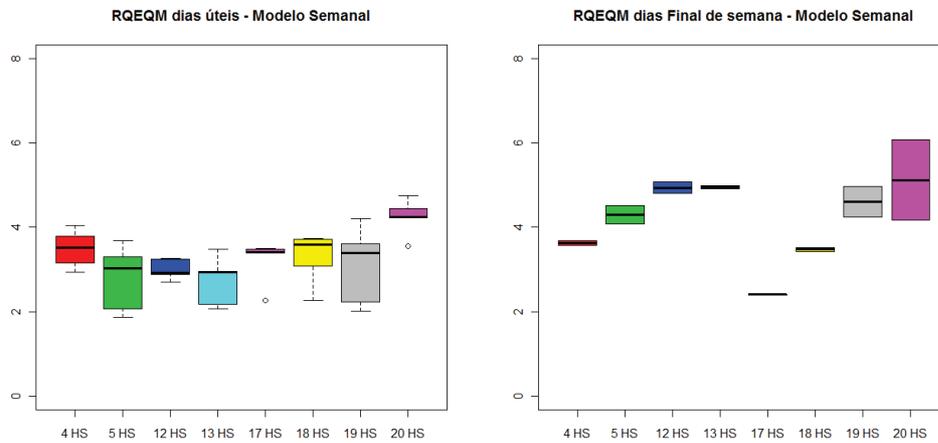


Figura 2-33: *BoxPlot* da RMSE para algumas horas selecionadas nos dias úteis e nos finais de semana de acordo com o modelo semanal.

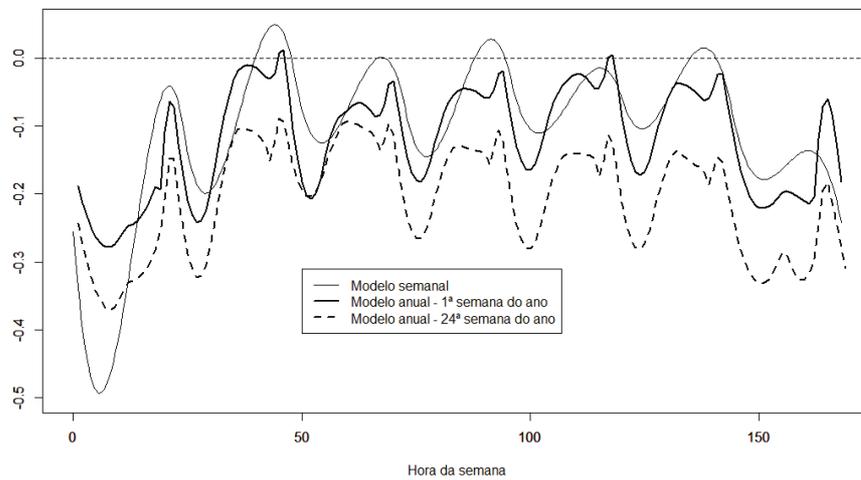


Figura 2-34: Constante estimada nos modelos semanal e anual. A constante estimada no modelo anual apresenta um formato mais parecido com os formatos observados para a curva diárias de carga. Ela se adapta para captar as mudanças que ocorrem na carga ao longo do ano.

Capítulo 3

Modelos funcionais não simultâneos

Nos modelos funcionais tratados até agora, os simultâneos, o valor corrente do processo resposta se relaciona somente com o valor corrente do processo preditor [ver, por exemplo, Ramsay & Silverman (2010)]. Senturk & Muller (2010) e Ramsay & Silverman (1997) propuseram modelos alternativos, nos quais o valor corrente do processo resposta passa a depender também dos valores passados do processo preditor. Neste capítulo, apresentamos uma generalização desses modelos alternativos, a qual pode ser aplicada aos dados da carga.

A estimação da generalização que propomos pode ser entendida como uma versão para dados funcionais do estimador de posto reduzido. Esse estimador está diretamente relacionado a conceitos de análise de correlação canônica. Em função disso, iniciamos o capítulo apresentando a derivação das correlações e das variáveis canônicas para o caso tradicional [não funcional]. Em seguida, apresentamos, também para o caso tradicional, a derivação do estimador de posto reduzido. Apresentamos também as correlações e variáveis canônicas para o caso funcional. Feito isso, descrevemos os modelos propostos por Senturk & Muller (2010) e Ramsay & Silverman (1997) e a nossa proposta de generalização.

Ainda neste capítulo, a fim de testar o comportamento do estimador de posto reduzido funcional conduzimos alguns exercícios de simulação.

3.1 O estimador de posto reduzido

O estimador de posto reduzido é utilizado tradicionalmente em um contexto de regressão multivariada com p variáveis dependentes, q variáveis independentes e no qual a matriz de coeficientes de regressão, B [com dimensão $p \times q$], tem posto $k < \min(p, q)$. Ele aparece também em um contexto de modelo de séries de tempo não estacionárias. Para maiores

detalhes ver Anderson (2001).

O estimador de posto reduzido está baseado em k variáveis canônicas correspondendo às k maiores correlações canônicas. Descrevemos sucintamente a derivação das variáveis canônicas e das correlações canônicas, maiores detalhes podem ser encontrados em Anderson (1984).

Na análise de correlação canônica, procuramos obter sequencialmente combinações lineares de variáveis que pertencem a dois grupos distintos. As combinações lineares são determinadas de forma a maximizar a correlação entre elas. A r -ésima combinação linear é construída de forma a exibir a r -ésima maior correlação possível com a correspondente combinação linear dentro do outro grupo e não ser correlacionada com nenhuma das combinações lineares anteriores. De maneira mais formal, seja X vetor aleatório com p componentes e matriz de covariância Σ , que por hipótese é positiva definida. Por simplicidade de exposição, assumimos que $EX = 0$. Considere a seguinte partição de X :

$$X = \begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix}$$

em que $X^{(1)}$ tem p_1 componentes e $X^{(2)}$ tem p_2 componentes e por conveniência $p_1 \leq p_2$.

A matriz de covariância é similarmente particionada:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

Considere as combinações lineares $U = \alpha^\top X^{(1)}$ e $V = \gamma^\top X^{(2)}$ em que α e γ são tais que U e V tenham variância unitária. Isto é:

$$\begin{aligned} 1 &= \alpha^\top \Sigma_{11} \alpha \\ 1 &= \gamma^\top \Sigma_{22} \gamma \end{aligned} \tag{3.1}$$

A correlação [note: $VAR(U) = VAR(V) = 1$] entre U e V é dada por:

$$EUU = \alpha^\top \Sigma_{12} \gamma \quad (3.2)$$

O lagrangeano associado a maximização de (3.2) sujeito as restrições descritas em (3.1) pode ser escrito como:

$$\Psi = \alpha^\top \Sigma_{12} \gamma - \frac{1}{2} \lambda \left(\alpha^\top \Sigma_{11} \alpha - 1 \right) - \frac{1}{2} \mu \left(\gamma^\top \Sigma_{22} \gamma - 1 \right)$$

em que λ e μ são multiplicadores de Lagrange. Diferenciando Ψ em relação a α e γ obtemos:

$$\begin{aligned} \frac{\partial \Psi}{\partial \alpha} &= \Sigma_{12} \gamma - \lambda \Sigma_{11} \alpha = 0 \\ \frac{\partial \Psi}{\partial \gamma} &= \Sigma_{12}^\top \alpha - \mu \Sigma_{22} \gamma = 0 \end{aligned} \quad (3.3)$$

Multiplicando à esquerda a primeira equação em (3.3) por α^\top e a segunda por γ^\top obtemos:

$$\begin{aligned} \alpha^\top \Sigma_{12} \gamma - \lambda \alpha^\top \Sigma_{11} \alpha &= 0 \\ \gamma^\top \Sigma_{12}^\top \alpha - \mu \gamma^\top \Sigma_{22} \gamma &= 0. \end{aligned} \quad (3.4)$$

Como $\alpha^\top \Sigma_{11} \alpha = 1$ e $\gamma^\top \Sigma_{22} \gamma = 1$, então $\lambda = \mu = \alpha^\top \Sigma_{12} \gamma$. Temos também que:

$$\begin{bmatrix} -\lambda \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & -\lambda \Sigma_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \gamma \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.5)$$

como Σ é positiva definida, $|\Sigma_{11}| |\Sigma_{22}| \neq 0$ e o determinante abaixo:

$$\begin{vmatrix} -\lambda \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & -\lambda \Sigma_{22} \end{vmatrix} = 0 \quad (3.6)$$

tem raízes, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p > 0$, para maiores detalhes ver Anderson (1984).

Como $\lambda = \alpha^\top \Sigma_{12} \gamma$ então λ é a correlação entre $U = \alpha^\top X^{(1)}$ e $V = \gamma^\top X^{(2)}$ quando α e γ satisfazem (3.6) para algum valor de λ . Nós queremos a máxima correlação então

tomamos $\lambda = \lambda_1$. Seja $(\alpha^{(1)}, \gamma^{(1)})$ a solução de (3.5) para $\lambda = \lambda_1$ e $U_1 = \alpha^{(1)}X^{(1)}$ e $V = \gamma^{(1)}X^{(2)}$. (U_1, V_1) é o primeiro par de variáveis canônicas com correlação canônica dada por λ_1 .

No segundo passo, visamos encontrar U_2 e V_2 , combinações lineares de $X^{(1)}$ e de $X^{(2)}$, respectivamente, não correlacionadas com U_1 e V_1 e que apresentem a maior correlação possível. No passo três, o objetivo é encontrar U_3 e V_3 , combinações lineares de $X^{(1)}$ e de $X^{(2)}$, respectivamente, não correlacionadas com U_1, V_1, U_2 e V_2 e que apresentem a maior correlação possível. Procedemos assim até que $m = p_1$ variáveis canônicas sejam encontradas. Para isso, além das restrições que aparecem em (3.1) adicionamos as seguintes restrições:

$$\begin{aligned} EUU_i &= 0 = \alpha^\top \Sigma_{11} \alpha^{(i)} \\ EUV_i &= 0 = \alpha^\top \Sigma_{12} \gamma^{(i)} \\ EVV_i &= 0 = \gamma^\top \Sigma_{22} \gamma^{(i)} \\ EVU_i &= 0 = \gamma^\top \Sigma_{21} \alpha^{(i)} \end{aligned} \quad (3.7)$$

Para maximizar $EU_{r+1}V_{r+1}$ escolhemos α, γ de modo que eles satisfaçam (3.1) e (3.7) para $i = 1, 2, \dots, r$ e montamos o seguinte lagrangeano:

$$\Psi = \alpha^\top \Sigma_{12} \gamma - \frac{1}{2} \lambda (\alpha^\top \Sigma_{11} \alpha - 1) - \frac{1}{2} \mu (\gamma^\top \Sigma_{22} \gamma - 1) + \sum_{i=1}^r \nu_i \alpha^\top \Sigma_{11} \alpha^{(i)} + \sum_{i=1}^r \theta_i \gamma^\top \Sigma_{22} \gamma^{(i)}$$

em que $\lambda, \mu, \nu_1, \dots, \nu_r, \theta_1, \dots, \theta_r$ são multiplicadores de lagrange. Derivando em relação a α e a γ e igualando a zero obtemos:

$$\begin{aligned} \frac{\partial \Psi}{\partial \alpha} &= \Sigma_{12} \gamma - \lambda \Sigma_{11} \alpha + \sum_{i=1}^r \nu_i \alpha^\top \Sigma_{11} \alpha^{(i)} = 0 \\ \frac{\partial \Psi}{\partial \gamma} &= \Sigma_{21} \alpha - \mu \Sigma_{22} \gamma + \sum_{i=1}^r \theta_i \gamma^\top \Sigma_{22} \gamma^{(i)} = 0 \end{aligned} \quad (3.8)$$

multiplicando a primeira equação de (3.8) à esquerda por $\alpha^{(j)}$ e a segunda equação à

esquerda por $\gamma^{(j)}$ obtemos:

$$\begin{aligned} 0 &= \nu_j \alpha^{(j)\top} \Sigma_{11} \alpha^{(j)} = \nu_j \\ 0 &= \theta_j \alpha^{(j)\top} \Sigma_{22} \alpha^{(j)} = \theta_j. \end{aligned} \quad (3.9)$$

Substituindo estes valores em (3.8) obtemos exatamente as mesmas equações de 3.3. Segue daí que necessariamente $\lambda^{(r+1)} = \mu^{r+1} = \alpha^{(r+1)\top} \Sigma_{12} \gamma^{(r+1)}$ que é igual a $EU_{r+1}V_{r+1}$. Concluimos daí que λ é a r -ésima maior raiz de 3.6. U_{r+1} e V_{r+1} são as $(r+1)$ -ésimas variáveis canônicas associadas à $\lambda^{(r+1)}$ -ésima correlação canônica.

Para estabelecer a relação entre o estimador de posto reduzido e as correlações canônicas, considere o modelo abaixo:

$$Y_i = X_i M + E_i \quad (3.10)$$

em que X_i e Y_i são vetores $1 \times q$ e $1 \times p$, respectivamente, e $i = 1, \dots, n$. Suponha que $1 \times q$ e $1 \times p$ vetores de médias tenham sido subtraídos de cada uma das n observações de X e Y respectivamente. O objetivo é estimar a matriz M , $q \times p$, de coeficientes sujeita à restrição de que $\text{posto}(M) = s < \min(q, p)$. Esta restrição pode ser entendida como: somente s combinações lineares dos regressores de fato importam para a predição de Y . Os erros são indicados pelo vetor E_i e apresentam média zero e matriz de covariância Σ positiva definida.

Assumindo que E_i tem distribuição normal multivariada, isto é:

$$E_i \sim N(0, \Sigma).$$

O problema de estimação de M pode ser escrito como o seguinte problema de maximização de verossimilhança:

$$2 \log L = \text{const} + n \log [\det(\Sigma^{-1})] - \text{tr} \left\{ (Y - XM) \Sigma^{-1} (Y - XM)^\top \right\} \quad (3.11)$$

sujeito à restrição $\text{posto}(M) = s$.

Observando que o estimador de máxima verossimilhança de Σ condicional a M é dado por:

$$\hat{\Sigma}^*|M = n^{-1} (Y - XM)^\top (Y - XM)$$

temos que o problema (3.11) pode ser reescrito como:

$$f(M) = \det \left((Y - XM)^\top (Y - XM) \right) \quad (3.12)$$

sujeito à posto(M) = s .

A restrição sobre o posto de M implica que podemos escrever:

$$M = AB$$

em que A é uma matriz $q \times s$. As colunas de A fornecem os pesos para as combinações lineares das componentes de X_i . Essas combinações lineares serão os novos regressores para Y_i . As colunas de B transferem a informação dos novos regressores para a variável dependente Y_i , isto é, são os parâmetros na regressão com regressores transformados. Para maiores detalhes ver Tso (1981).

Os estimadores de A e B estão baseados em correlações canônicas. Para ver isso, considere X e Y como sendo matrizes tais que:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \text{ e } Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}.$$

A matriz A deve satisfazer a seguinte normalização:

$$P^\top P = I_s \text{ em que } P = XA \quad (3.13)$$

isto é, as colunas de P são combinações lineares ortogonais das variáveis de X . Sendo

assim a expressão (3.12) pode ser reescrita como:

$$f(P, B) = \det \left([Y - PB]^\top [Y - PB] \right) \quad (3.14)$$

Tso (1981) demonstra que o valor mínimo de (3.14) é:

$$(\det(\gamma^{-1}))^2 (1 - \lambda_1)(1 - \lambda_2) \cdots (1 - \lambda_s) \quad (3.15)$$

em que $\{\lambda_i, i = 1, \dots, s\}$ são as s maiores correlações canônicas entre X e Y . A matriz γ , $p \times p$, vem do problema de determinação das correlações canônicas e é tal que $Y = V\gamma^{-1}$, em que as colunas de V são ortogonais.

Formas funcionais explícitas para A e B , os argumentos que minimizam (3.12), podem ser encontradas em Izenman (1975):

$$A = \Sigma_{Y_1 Y_1}^{-\frac{1}{2}} \left[V_1 : \cdots : V_s \right]$$

$$B = \begin{bmatrix} V_1^T \\ \vdots \\ V_s^T \end{bmatrix} \Sigma_{Y_1 Y_1}^{\frac{1}{2}} \Sigma_{Y_1 X_1} \Sigma_{X_1 X_1}^{-1}$$

em que

$$\Sigma = \begin{bmatrix} \Sigma_{Y_1 Y_1} & \Sigma_{Y_1 X_1} \\ \Sigma_{X_1 Y_1} & \Sigma_{X_1 X_1} \end{bmatrix}$$

é a matriz de covariância de:

$$\begin{bmatrix} Y_1 \\ X_1 \end{bmatrix}$$

e V_1^T, \dots, V_s^T são os autovetores de $\Sigma_{Y_1 Y_1}^{-\frac{1}{2}} \Sigma_{Y_1 X_1} \Sigma_{X_1 X_1}^{-1} \Sigma_{X_1 Y_1} \Sigma_{Y_1 Y_1}^{-\frac{1}{2}}$.

3.2 Análise de correlações canônicas para dados funcionais

As variáveis canônicas para o caso funcional são análogas ao caso clássico com algumas adaptações, por exemplo, de produtos internos. A interpretação delas também se mantém a mesma. Nos parágrafos seguintes apresentamos, de forma sucinta, a derivação das variáveis canônicas funcionais e maiores detalhes podem ser obtidos em Leurgans, Moyeed & Silverman (1993).

Considere as funções aleatórias $Y, X : J \rightarrow \mathbb{R}$, em que $J = [0, T]$. Por simplicidade de exposição, consideramos que Y e X estão centralizadas, isto é, subtraídas de suas respectivas médias. Considere também as seguintes funções de covariância.

$$\Gamma_{11}(s, t) = E[X(s)X(t)]$$

$$\Gamma_{22}(s, t) = E[Y(s)Y(t)]$$

$$\Gamma_{12}(s, t) = E[X(s)Y(t)]$$

Considere as funções $f, g : J \rightarrow \mathbb{R}$ e $\Gamma : J \times J \rightarrow \mathbb{R}$, escrevemos $f^\top g$ para designar¹ o produto interno $\int f(s)g(s)ds$ e Γg para denotar $\int \Gamma(s, t)g(t)dt$.

Assim como em boa parte da análise de dados funcionais, a penalização de rugosidade é também uma ideia central na determinação das variáveis canônicas. Na análise de correlação canônica funcional, a penalização de rugosidade é gerada por uma forma quadrática, escrita por convenção como $[f, g]$. Em que $[f, g] := \int f''(t)g''(t)dt$. Uma medida da rugosidade é obtida fazendo $f = g$ na forma quadrática anterior, de onde vem $\int [f''(t)]^2 dt$. Se g satisfaz condições adicionais de suavidade [quadrado integrável e quarta derivada contínua] então a integração por partes $\int f''g'' = \int fg^{(iv)}$. Como decorrência, podemos definir $[fg]$ como $f^\top D^4 g$, em que D^4 o operador de derivada de

¹A existência das integrais é assumida implicitamente.

quarta ordem.

Mais especificamente, para a determinação das variáveis canônicas funcionais, considere duas funções u e v . Definimos $\gamma(u, v)$ como sendo o quadrado das correlações de $u^\top X$ e $v^\top Y$ da seguinte forma:

$$\gamma(u, v) = \frac{[u^\top \Gamma_{12} v]^2}{[u^\top \Gamma_{11} u][v^\top \Gamma_{22} v]} \quad (3.16)$$

Analogamente ao caso discreto, o nosso objetivo é encontrar funcionais lineares $u^\top X$ de X e $v^\top Y$ de Y que maximizem $\gamma(u, v)$, o que é equivalente a maximizar:

$$\iint [u(s)\Gamma_{12}(s, t)v(t)]^2 ds dt$$

sujeito às restrições:

$$\iint u(s)\Gamma_{11}(s, t)u(t) ds dt = 1 \text{ e } \iint v(s)\Gamma_{22}(s, t)v(t) ds dt = 1. \quad (3.17)$$

A adição de penalização de rugosidade pode ser feita modificando as restrições em (3.17) para:

$$\iint u(s)\Gamma_{11}(s, t)u(t) ds dt + \alpha_1[u, u] = 1 \text{ e } \iint v(s)\Gamma_{22}(s, t)v(t) ds dt + \alpha_2[v, v] = 1 \quad (3.18)$$

em α_1 e α_2 são parâmetros de suavidade positivos. O problema de maximizar (3.16) sujeito a (3.18) é equivalente a maximizar:

$$\tilde{\gamma}(u, v) = \frac{[u^\top \Gamma_{12} v]^2}{[u^\top \Gamma_{11} u + \alpha_1[u, u]][v^\top \Gamma_{22} v + \alpha_2[v, v]]} \quad (3.19)$$

O par ordenado $\left((u^{(1)})^\top, (v^{(1)})^\top\right)$ que maximiza (3.19), está associado à autofunção

correspondente ao maior valor de ρ , ρ_1 digamos, que satisfaz o sistema de equações²:

$$\begin{bmatrix} 0 & \Gamma_{12} \\ \Gamma_{21} & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \rho \begin{bmatrix} \Gamma_{11} + \alpha_1 D^4 & 0 \\ 0 & \Gamma_{22} + \alpha_2 D^4 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.20)$$

As segundas variáveis canônicas funcionais $(u^{(2)})^\top$ e $(v^{(2)})^\top$, que maximizam (3.19) e são ortogonais a $(u^{(1)})^\top$ e $(v^{(1)})^\top$, estão associadas ao segundo maior valor de ρ , ρ_2 digamos que satisfaz o sistema de equações (3.20). As variáveis canônicas funcionais $(u^{(1)})^\top$, $(v^{(1)})^\top$, $(u^{(2)})^\top$, $(v^{(2)})^\top, \dots$ dependem do parâmetro de suavização α .

Versões empíricas para o problema acima são obtidas quando Γ_{11} , Γ_{22} e Γ_{12} são substituídos por seus análogos amostrais.

3.3 Modelos funcionais não simultâneos

Nos modelos funcionais simultâneos [*concurrent models*] o valor corrente do processo resposta se relaciona somente com o valor corrente do processo preditor [ver, por exemplo, Ramsay & Silverman (2010)]. Também é assim para várias outras classes de modelos que se destinam ao tratamento de dados funcionais. Um exemplo são os modelos de análise de dados funcionais para dados longitudinais [Yao, Muller & Wang (2005)]. Seja $X(t)$ uma função aleatória com $E(X(t)) = \mu(t)$, $cov(X(t), X(s)) = G(s, t)$ e domínio compacto J . Assuma que $G(s, t)$ possui expansão ortogonal [no sentido de L^2], em termos de autofunções ϕ_k e autovalores $\lambda_1 > \lambda_2 > \dots > \lambda_k > \dots$, $G(s, t) = \sum_k \lambda_k \phi_k(s) \phi_k(t)$, $t, s \in J$. A função $X(t)$ pode ser expressa como [para maiores detalhes consultar Yao, Muller & Wang (2005)]:

$$X(t) = \mu(t) + \sum_k \xi_k \phi_k(t)$$

em que ξ_k são variáveis aleatórias não correlacionadas com média zero e variância $E(\xi_k^2) = \lambda_k$, com $\sum_k \lambda_k < \infty$.

A fim de levar em conta questões de estimação, consideramos, a partir de agora, uma

² ρ_1 é analogo ao autovalor encontrado no caso clássico.

amostra de X , denotada por $\{X_i, i = 1, \dots, n\}$. Faça Y_{ij} a observação da função aleatória $X_i(\cdot)$ no tempo T_{ij} , $j = 1, \dots, N_i$. Temos então que:

$$Y_{ij} = \mu(T_{ij}) + \sum_{k=1}^M \xi_{ik} \phi_k(T_{ij}) + \varepsilon_{ij}.$$

em que ε_{ij} é um termo de erro tal que para todo $i \neq m$, ε_{ij} é independente ε_{mw} . O termo de erro ε_{ij} é independente do coeficiente aleatório ξ_{ik} , para todo $k \leq M$.

Senturk & Muller (2010) propuseram um modelo alternativo ao de Yao, Muller & Wang (2005), no qual o valor corrente do processo resposta passa a depender também dos valores passados do processo preditor, através da chamada função de índice histórico, HIF [*History Index Function*]. Neste modelo, o valor corrente do processo resposta $Y(t)$ no tempo t depende da história recente do processo preditor X em uma janela de tamanho Δ , da seguinte forma:

$$E[Y(t)|X(s) \text{ } s \in (0, T)] = \beta_0(t) + \beta_1(t) \int_0^{\Delta} \gamma(u) X(t-u) du \quad (3.21)$$

para $t \in [\Delta, T]$. A função de índice histórico [HIF] determina a influência do passado recente do processo preditor sobre a variável resposta. Podemos observar que a variável explicativa passa a ser uma espécie de média ponderada dos valores passados e presente do processo preditor em que os pesos são dados por $\gamma(u)$.

Dentro dessa linha, de relacionar o valor corrente do processo resposta não só com o valor corrente do processo preditor, temos também o modelo desenvolvido em Ramsay & Silverman (2010). Considere X_i e Y_i funções com domínio I_x e I_y , respectivamente e o seguinte modelo linear:

$$E(Y_i(t)/X_i(t)) = \hat{Y}_i(t) = \alpha(t) + \int_{I_x} X_i(s) \beta(s, t) ds \quad (3.22)$$

Para estabelecer a relação entre o modelo de Senturk & Muller (2010) e Ramsay & Silverman (2010), suponhamos que $\beta(s, t) = \vartheta(s) \varphi(t)$. A equação (3.22) pode ser reescrita

como:

$$\hat{Y}_i(t) = \alpha(t) + \int_{I_x} X_i(s) \vartheta(s) \varphi(t) ds = \varphi(t) \int_{I_x} X_i(s) \vartheta(s) ds. \quad (3.23)$$

Em (3.21) o valor da integral depende de t , o que faz dela uma espécie de média móvel da função explicativa. Além disso, a integral é tomada para intervalos específicos do domínio para cada t . Em (3.23) a integral não depende de t , portanto se torna um escalar que é o mesmo para todo t no domínio da função dependente.

O modelo de Senturk & Muller é desenvolvido em um contexto, no qual as funções Y_i e X_i são observadas em um número aleatório, N_i , de pontos de seu domínio. Já em Ramsay & Silverman é desenvolvido em um contexto no qual para cada uma das funções Y_i e X_i temos um número fixo de observações em pontos determinados dos respectivos domínios. Além disso, o estimador de Ramsay & Silverman é derivado a partir da representação das funções Y_i e X_i em base de funções. Tais características do modelo de Ramsay & Silverman se aproximam do contexto de nosso interesse. Sendo assim seguiremos os passos trilhados por Ramsay & Silverman para derivação do nosso estimador. Mas manteremos a notação de Senturk & Muller que, acreditamos, é mais adequada para os nossos objetivos.

Em ambos os modelos, Senturk & Muller e Ramsay & Silverman, para cada t fixado obtemos uma espécie de média ponderada da variável explicativa como regressor para Y no ponto t . No modelo de Ramsay & Silverman, a média ponderada é tomada ao longo de todo o domínio, I_x , da variável explicativa. Em Senturk & Muller, a média é tomada para valores $X(u)$ tais que u pertence a $(t - \Delta, t)$. A nossa proposta é tomar a média ponderada em intervalos apropriadamente escolhidos ao longo do domínio da variável explicativa.

Considere o intervalo (T_1, T_2) contido no domínio das variáveis X e Y . Faça $\Delta_2 = T_2 - T_1$, iremos considerar modelos nos quais supomos que para todo $t \in (T_1, T_2)$, a variável dependente Y é explicada pela média ponderada de X nesse mesmo intervalo e os pesos são dados por $\gamma(u)$, conforme expressão abaixo:

$$E[Y(t)|X(s) \text{ } s \in I_x] = \beta_0(t) + \beta(t) \int_0^{\Delta_2} \gamma(u) X(T_2 - u) du. \quad (3.24)$$

O domínio, $[0, T]$, das funções que aparecem em (3.24) será particionado em k subintervalos disjuntos. Para cada subintervalo temos uma versão de (3.24) com novos intervalos de integração. Dessa forma, o importante para explicar $Y(t)$ para qualquer hora t no intervalo (T_1, T_2) é o patamar dado por $\int_0^{\Delta_2} \gamma(u)X(T_2 - u)du$. A forma de transferência da informação desse patamar para a variável dependente na hora t será determinada por $\beta(t)$.

Ao longo deste trabalho, temos uma especial atenção à modelagem da série de carga de energia elétrica horária. Caso no qual temos intervalos naturais a serem considerados: madrugada, manhã, tarde e noite. Sendo assim, iremos considerar modelos com quatro HIF's, $\gamma_1(\cdot)$, $\gamma_2(\cdot)$, $\gamma_3(\cdot)$ e $\gamma_4(\cdot)$. O modelo para carga pode ser escrito como:

$$\begin{aligned}
 Y_i(t) = & \beta(t) \int_0^6 \gamma_1(u)X_i(6 - u)du \times I_{(0,6]}(t) + \\
 & \beta(t) \int_0^6 \gamma_2(u)X_i(12 - u)du \times I_{(6,12]}(t) + \\
 & \beta(t) \int_0^6 \gamma_3(u)X_i(18 - u)du \times I_{(12,18]}(t) + \\
 & \beta(t) \int_0^6 \gamma_4(u)X_i(24 - u)du \times I_{(18,24]}(t) + \varepsilon_i(t)
 \end{aligned}$$

em que:

$$I_{(a,b]}(t) = \begin{cases} 1 & \text{se } t \in (a, b] \\ 0 & \text{se } t \notin (a, b] \end{cases}$$

Admitimos que:

$$\begin{aligned} \lim_{t \rightarrow 6^-} \beta(t) \int_0^6 \gamma_1(u) X_i(6-u) du &= \lim_{t \rightarrow 6^+} \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du \\ \lim_{t \rightarrow 12^-} \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du &= \lim_{t \rightarrow 12^+} \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du \\ &\text{e} \\ \lim_{t \rightarrow 18^-} \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du &= \lim_{t \rightarrow 18^+} \beta(t) \int_0^6 \gamma_4(u) X_i(24-u) du \end{aligned}$$

Suponhamos que o regressor X que aparece na expressão acima seja a temperatura. O modelo desenvolvido neste capítulo poderia ser traduzido em termos da relação entre a carga e a temperatura da seguinte forma: a carga do dia i para qualquer hora t da manhã, isto é, $Y_i(t)$ para $t \in (6, 12]$, é explicada por uma espécie de média ponderada da temperatura da manhã do dia i . Os pesos são dados por $\gamma_2(\cdot)$, isto é, a temperatura média é $\int_0^6 \gamma_2(u) X_i(12-u) du$. A transferência da informação da temperatura observada em todo o período da manhã para a carga da hora s , $Y_i(s)$, se dá através da multiplicação da temperatura média ponderada da manhã por $\beta(s)$, $\beta(s) \int_0^6 \gamma_2(u) X_i(12-u) du$ em que $s \in (6, 12]$.

Propomos um estimador que possa levar em consideração as alterações propostas para o modelo de Senturk & Muller e Ramsay & Silverman. A fim de motivar o estimador, na equação (3.21), passamos o termo $\beta_1(t)$ para dentro da integral e ficamos com:

$$E[Y(t)|X(u) \ u \in (0, \Delta)] = \beta_0(t) + \int_0^{\Delta} \alpha_t(u) X(t-u) du \quad (3.25)$$

com $\alpha_t(u) = \beta_1(t)\gamma(u)$ para $u \in (0, \Delta)$. Para fins didáticos, suponha que Δ é um número inteiro e sem perda de generalidade que $\beta_0(t) = 0$. O análogo discreto da integral que aparece em (3.25) para a i -ésima função observada é dado por:

$$Y_i(t) = \alpha_t(0)X_i(t) + \alpha_t(1)X_i(t-1) + \cdots + \alpha_t(\Delta)X_i(t-\Delta) + \varepsilon(t) \quad (3.26)$$

no tempo s temos que:

$$Y_i(s) = \alpha_s(0)X_i(s) + \alpha_s(1)X_i(s-1) + \cdots + \alpha_s(\Delta)X_i(s-\Delta) + \varepsilon(s)$$

obtemos então que:

$$\frac{\alpha_t(0)}{\alpha_s(0)} = \frac{\beta(t)\gamma(0)}{\beta(s)\gamma(0)} = \frac{\alpha_t(1)}{\alpha_s(1)} = \frac{\beta(t)\gamma(1)}{\beta(s)\gamma(1)}$$

O que acontece de fato no modelo escrito em (3.21) é a imposição velada de uma série de restrições sobre os parâmetros funcionais $\alpha's$. Em um modelo de regressão multivariada, alguns conjuntos de restrições lineares sobre os parâmetros podem ser entendidos como uma redução do posto da matriz de coeficientes, conforme descrito no modelo de regressão de posto reduzido [c.f. seção 3.1]. As restrições sobre os $\alpha's$ funcionais serão tratadas de maneira análoga e por isso propomos uma versão funcional do modelo de regressão de posto reduzido.

3.3.1 A generalização funcional do modelo de posto reduzido

O modelo tradicional de posto reduzido foi desenvolvido para um contexto no qual as observações são vetores aleatórios. Algumas adaptações desse modelo devem ser feitas para o caso no qual as unidades observacionais são funções. Para isso, considere o modelo abaixo:

$$\begin{aligned}
Y_i(t) = & \beta(t) \int_0^6 \gamma_1(u) X_i(6-u) du \times I_{(0,6]}(t) + \\
& \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du \times I_{(6,12]}(t) + \\
& \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du \times I_{(12,18]}(t) + \\
& \beta(t) \int_0^6 \gamma_4(u) X_i(24-u) du \times I_{(18,24]}(t) + \varepsilon_i(t)
\end{aligned} \tag{3.27}$$

suponha que Y_i seja a função resposta, X_i seja a função explicativa e que ambas estejam centralizadas, ou seja, tiveram suas respectivas médias subtraídas. Assumimos também que Y_i e X_i são *iid's*. Assim como no capítulo 2, assumimos que as funções Y_i e X_i podem ser escritas em termos de componentes de determinadas bases como abaixo:

$$X_n(t) = \sum_{j=1}^J g_{nj} \phi_j(t) \quad \text{e} \quad Y_n(t) = \sum_{k=1}^K h_{nk} \varphi_k(t) \quad \text{para todo } t \in (a, b].$$

As funções de covariância das funções X_i e Y_i , $G_X(s, t)$ e $G_Y(s, t)$, são dadas por:

$$G_X(s, t) = E \left(\sum_{j=1}^J g_{nj} \phi_j(s), \sum_{j=1}^J g_{nj} \phi_j(t) \right)$$

e

$$G_Y(s, t) = E \left(\sum_{k=1}^K h_{nk} \varphi_k(s), \sum_{k=1}^K h_{nk} \varphi_k(t) \right)$$

que podem ser reescritas em formato matricial como:

$$\Gamma_{X_i X_i} = E [g^\top \mathcal{J}_\phi g]$$

$$\Gamma_{Y_i Y_i} = E [h^\top \mathcal{J}_\varphi h]$$

em que $g = [g_1, \dots, g_J]^\top$, $h = [h_1, \dots, h_K]^\top$ e \mathcal{J}_ϕ e \mathcal{J}_φ são matrizes $J \times J$ e $K \times K$,

respectivamente, dadas por:

$$\begin{aligned}\mathcal{J}_\phi &= \phi(s)\phi(t)^\top \\ &e \\ \mathcal{J}_\varphi &= \varphi(s)\varphi(t)^\top\end{aligned}\tag{3.28}$$

com $\phi(s) = [\phi_1(s), \dots, \phi_J(s)]^\top$ e $\varphi(s) = [\varphi_1(s), \dots, \varphi_K(s)]^\top$.

A derivação das correlações canônicas e do estimador de posto reduzido, conduzida ao longo da seção 3.1, depende também das matrizes de covariâncias cruzadas que são dadas por:

$$\begin{aligned}\Gamma_{X_i Y_i} &= E \left[g^\top \mathcal{J}_{\phi\varphi} h \right] \\ \Gamma_{Y_i X_i} &= E \left[h^\top \mathcal{J}_{\varphi\phi} g \right]\end{aligned}$$

em que:

$$\begin{aligned}\mathcal{J}_{\phi\varphi} &= \phi(s)\varphi(t)^\top \\ &e \\ \mathcal{J}_{\varphi\phi} &= \varphi(s)\phi(t)^\top\end{aligned}\tag{3.29}$$

Definidas as matrizes de variância e covariância $\Gamma_{Y_i Y_i}$, $\Gamma_{X_i X_i}$, $\Gamma_{Y_i X_i}$ e $\Gamma_{X_i Y_i}$, o estimador de posto reduzido funcional tem formas explícitas para A e B dadas por [conforme Izenman (1975)]:

$$\begin{aligned}A &= \Gamma_{Y_i Y_i}^{-1/2} V_1 \\ B &= \begin{bmatrix} V_1^T \\ \vdots \\ V_k^T \end{bmatrix} \Gamma_{Y_i Y_i}^{\frac{1}{2}} \Gamma_{Y_i X_i} \Sigma_{X_i X_i}^{-1}\end{aligned}\tag{3.30}$$

em que V_j é o j -ésimo autovetor de $\Gamma_{Y_i Y_i}^{-1/2} \Gamma_{Y_i X_i} \Sigma_{X_i X_i}^{-1} \Gamma_{X_i Y_i} \Gamma_{Y_i Y_i}^{-1/2}$. A interpretação de Izenman (1975) para o modelo de posto reduzido é que a informação de X para Y pode ser transmitida utilizando $k \leq q$ canais. Vamos admitir aqui que a informação média pode ser transmitida utilizando apenas um canal, que é a temperatura média ponderada com pesos dados por $\gamma(\cdot)$. Em termos das correlações canônicas significa que estamos utilizando apenas a primeira variável canônica de X .

3.4 Simulações

Nesta seção conduzimos alguns exercícios de simulação que visam testar o comportamento do estimador de posto reduzido funcional. Começamos com um modelo simples com apenas uma variável explicativa. Logo após, introduzimos duas variáveis explicativas não correlacionadas entre si. Depois disso, simulamos um modelo no qual as duas variáveis explicativas apresentam uma determinada dependência. E por último, propomos uma forma de estimar as HIF's corrigindo a dependência entre as variáveis.

3.4.1 Modelo com uma variável explicativa

Considere o seguinte modelo:

$$\begin{aligned}
 Y_i(t) = & \beta(t) \int_0^6 \gamma_1(u) X_i(6-u) du \times I_{(0,6]}(t) + \\
 & \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du \times I_{(6,12]}(t) + \\
 & \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du \times I_{(12,18]}(t) + \\
 & \beta(t) \int_0^6 \gamma_4(u) X_i(24-u) du \times I_{(18,24]}(t) + \varepsilon_i(t)
 \end{aligned} \tag{3.31}$$

em que:

$$\begin{aligned} \lim_{t \rightarrow 6^+} \beta(t) \int_0^6 \gamma_1(u) X_i(6-u) du &= \lim_{t \rightarrow 6^-} \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du \\ \lim_{t \rightarrow 12^+} \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du &= \lim_{t \rightarrow 12^-} \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du \\ \lim_{t \rightarrow 18^+} \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du &= \lim_{t \rightarrow 18^-} \beta(t) \int_0^6 \gamma_4(u) X_i(18-u) du \end{aligned}$$

e $\varepsilon_i(t)$ desempenha o mesmo papel que o ruído branco nos modelos tradicionais de regressão.

A variável Y assumirá valores de tal maneira que o seu formato seja parecido com o da curva de carga. A variável X assume valores que fazem o seu formato ficar próximo do formato da carga defasada, da seguinte forma:

$$X_i(t) = f(t) + \zeta_i(t) \quad (3.32)$$

em que $\zeta_i(\cdot)$ é um ruído branco com desvio padrão 0.07 e $f(t)$ é uma função determinística [sua forma funcional está descrita no apêndice 1]. Foram simuladas duas amostras de X , uma com 100 e outra com 1000 observações, conforme descrito em (3.32). Elas foram avaliadas para todos os inteiros no intervalo $(0, 24]$.

As funções $\gamma_1(u)$, $\gamma_2(u)$, $\gamma_3(u)$ e $\gamma_4(u)$ também foram escolhidas de forma que a variável dependente tivesse um formato parecido com o da carga de energia elétrica. Em (3.33), apresentamos as funções que são proporcionais [o símbolo \propto indica proporcionalidade] às γ_j' s. As formas funcionais completas que descrevem as $\gamma_j(u)$'s podem ser encontradas no apêndice 1:

$$\begin{aligned}
\gamma_1(u) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-4)}{4}\right)\right) & u \in (1, 6] \\
\gamma_2(u) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-9)}{5}\right)\right) & u \in (6, 12] \\
\gamma_3(u) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-15)}{4.5}\right)\right) & u \in (12, 17] \\
\gamma_4(u) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-22)}{5}\right)\right) & u \in (17, 24]
\end{aligned} \tag{3.33}$$

O parâmetro $\beta(t)$ foi obtido como função das $X_i(t)$'s, conforme³ (3.34):

$$\beta(t) = \sum_{i=1}^n \frac{F_k(X_i(t))}{n}. \tag{3.34}$$

A forma funcional que descreve F_k , $k = 1, \dots, 4$, pode ser encontrada no apêndice 1.

Na figura (3.1), podemos observar algumas funções X's e Y's simuladas.

Na figura (3.2), temos as HIF's estimadas para cada um dos períodos através do estimador de posto reduzido funcional. A HIF correspondente ao segundo período é melhor estimada para a amostra de 1000 observações em comparação à amostra de 100 observações. Para as simulações seguintes tomaremos uma amostra com 500 observações.

Na figura (3.3) temos o $\beta(t)$ estimado para o modelo (3.31).

3.4.2 Modelo com duas variáveis explicativas

Uma das variáveis explicativas, denominada X , será exatamente a mesma simulada em (3.32). A segunda variável explicativa, denominada $Temp$, será dada por:

$$Temp_i(t) = 0.3 \times \sin\left(\pi \times 2 \times \frac{(t-1)}{44}\right) + 0.2 + \epsilon_i(t) \text{ para } t \in (1, 24)$$

em que $\epsilon_i(t)$ é ruído branco com desvio padrão 0.07 e $i = 1, \dots, 100$. O termo de erro $\epsilon_i(\cdot)$ é ortogonal a $\zeta_i(\cdot)$ para todo i . Em que $\zeta_i(\cdot)$ é o ruído branco associado à X_i definida em

³Os X_i 's utilizados para gerar o β não são os mesmos utilizados para gerar os Y_i 's. Eles são muito parecidos com os X_i 's que geraram os Y_i 's, já que eles vêm da mesma distribuição.

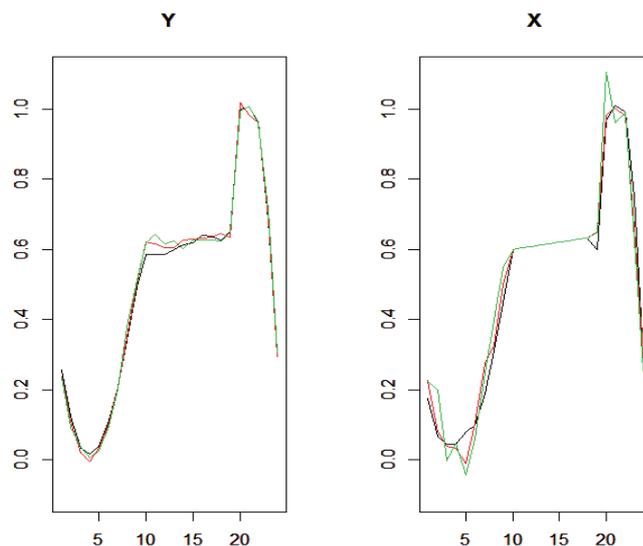


Figura 3.1: Alguns X 's e Y 's simulados.

(3.32). Inicialmente, não existem HIF's associadas a variável $Temp$. O modelo é escrito como:

$$Y_i(t) = \beta(t) \sum_{k=1}^4 \int_0^6 \gamma_k(u) X_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t) + \delta(t)Temp(t) + \varepsilon_i(t)$$

em que $i = 1, \dots, 500$, T_k e T_{k-1} assumem os valores 6, 12, 18, 24 e 0, 6, 12, 18 respectivamente. $\delta(\cdot)$ é o parâmetro funcional associado a $Temp$ e foi tomado como constante e igual a 1 para todo t . Na figura (3.4) podemos observar as HIF's estimadas associadas à variável explicativa X .

A seguir iremos associar a variável $Temp$ a determinadas HIF's. Assim como no caso da variável explicativa X , consideramos para a $Temp$ também quatro períodos: madrugada, manhã, tarde e noite. Assim temos a função $\rho_1(\cdot)$ indicando a HIF da madrugada, $\rho_2(\cdot)$

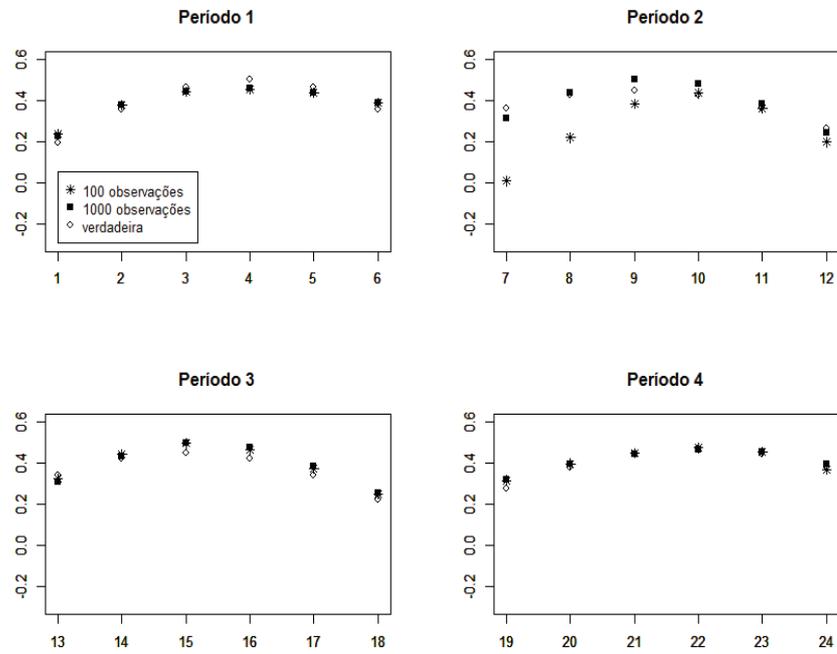


Figura 3·2: *HIF's* estimadas para cada um dos períodos.

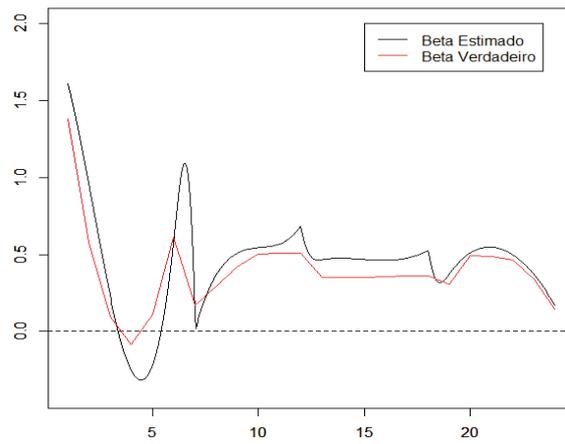


Figura 3·3: Beta estimado no modelo funcional não simultâneo.

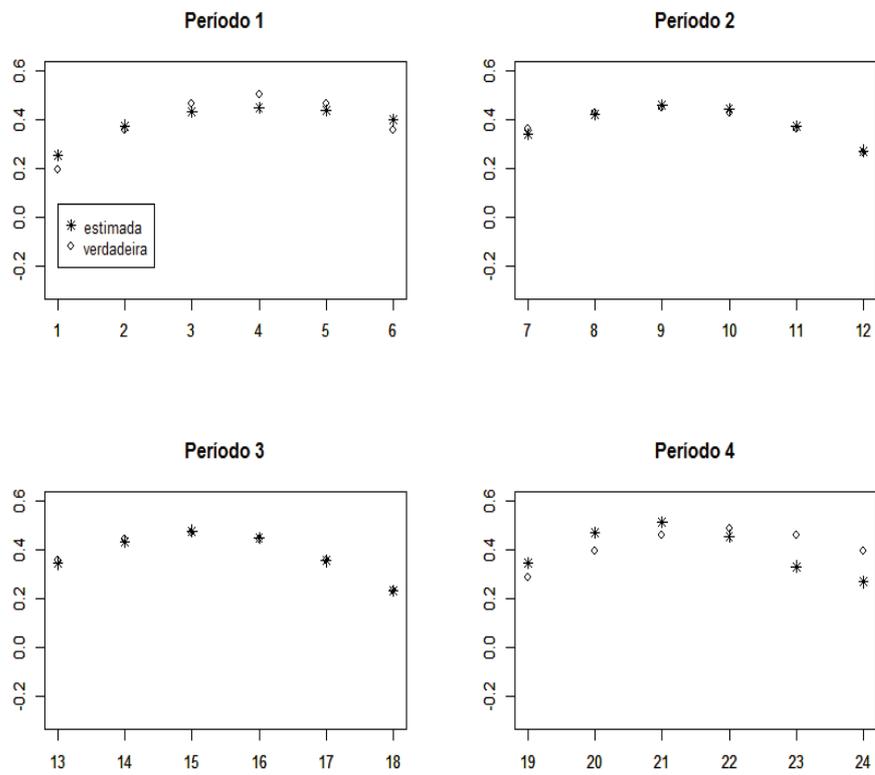


Figura 3-4: HIF's estimadas para cada um dos períodos em um modelo com duas variáveis explicativas. Apenas a variável X entra no modelo de maneira não simultânea.

indicando a HIF da manhã, $\rho_3(\cdot)$ indicando a HIF da tarde e $\rho_4(\cdot)$ indicando a HIF da noite. Elas são proporcionais às funções abaixo:

$$\begin{aligned}\rho_1(s) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-3)}{5}\right)\right) \quad s \in (1, 6] \\ \rho_2(s) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-9.5)}{4.8}\right)\right) \quad s \in (6, 12] \\ \rho_3(s) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-15.9)}{5}\right)\right) \quad s \in (12, 18] \\ \rho_4(s) &\propto \sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-33)}{8}\right)\right) \quad s \in (18, 24]\end{aligned}\tag{3.35}$$

O modelo com duas variáveis é dado por:

$$\begin{aligned}Y_i(t) &= \beta(t) \sum_{k=1}^4 \int_0^6 \gamma_k(u) X_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t) + \\ &\quad \delta(t) \sum_{j=1}^4 \int_0^6 \rho_j(u) Temp_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t) + \varepsilon_i(t)\end{aligned}\tag{3.36}$$

em que $i = 1, \dots, 500$, T_k e T_{k-1} assumem os valores 6, 12, 18, 24 e 0, 6, 12, 18, respectivamente.

Na figura (3.5), temos as HIF's estimadas para X no modelo (3.36). As variáveis X e $Temp$ são não correlacionadas entre si.

Nas figuras (3.6) e (3.7), podemos observar as HIF's estimadas para X e para $Temp$, respectivamente. Na curva indicada como variáveis não correlacionadas, os ruídos brancos associados a X , ζ_i 's, e os ruídos brancos associados à $Temp$, ϵ_i 's, são ortogonais para todo i e j , com $i = 1, \dots, 100$ e $j = 1, \dots, 100$. O desvio padrão do ruído branco é 0.07. Na curva indicada como variáveis dependentes os ruídos brancos tais que $i = j$ não são ortogonais, o que gera uma relação de dependência entre as variáveis explicativas X e $Temp$. A piora na estimação, principalmente no período 3, é resultado da dependência entre as variáveis explicativas.

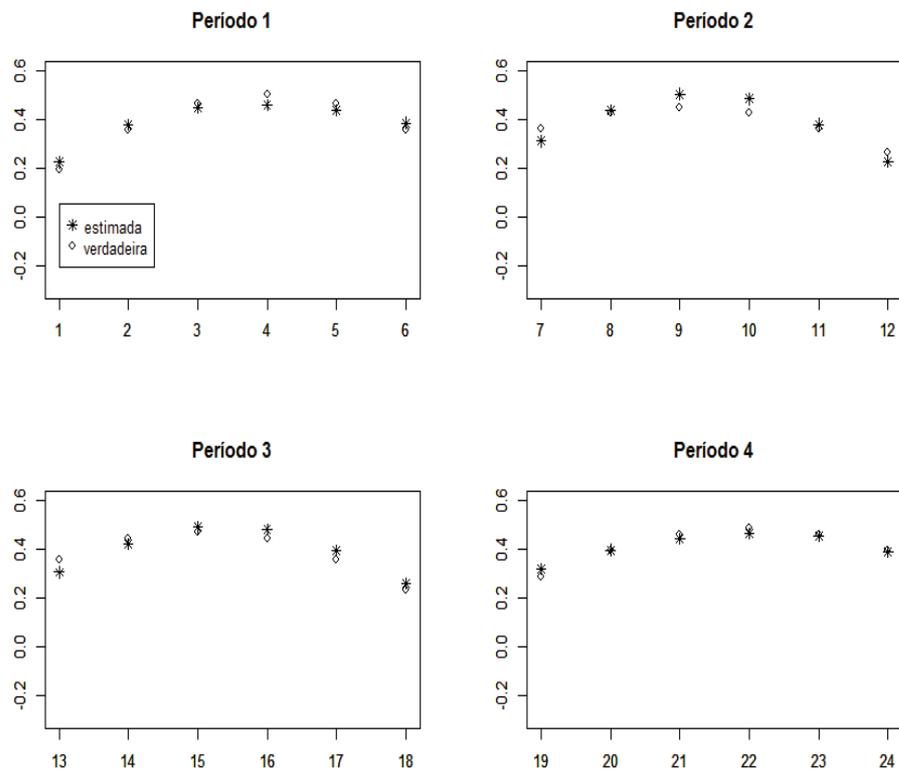


Figura 3-5: HIF's associadas à variável explicativa X , em um modelo com duas variáveis explicativas, X e $Temp$. A variável X está associada a quatro HIF's, $\gamma_1(\cdot), \dots, \gamma_4(\cdot)$ e a variável $Temp$ está associada a quatro outras HIF's $\rho_1(\cdot), \dots, \rho_4(\cdot)$. As variáveis X e $Temp$ são não correlacionadas.

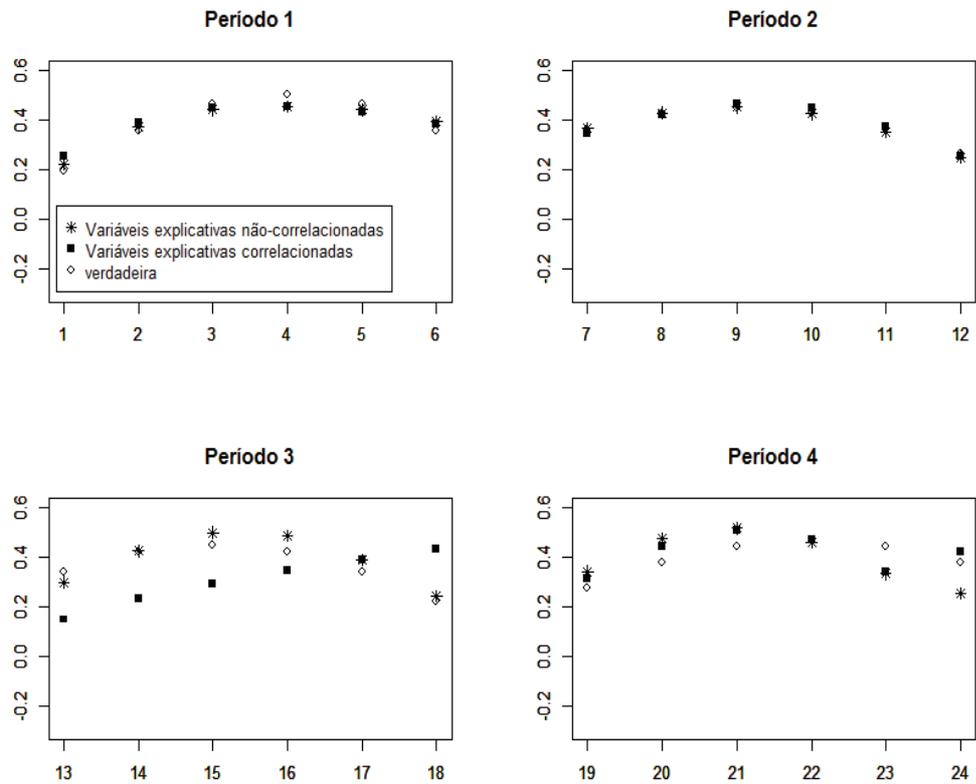


Figura 3.6: HIF's estimadas associadas a X , em um modelo com duas variáveis explicativas, que apresentam correlação entre elas. A variável X está associada a quatro HIF's, $\gamma_1(\cdot), \dots, \gamma_4(\cdot)$ e a variável Temp está associada a quatro outras HIF's $\rho_1(\cdot), \dots, \rho_4(\cdot)$.

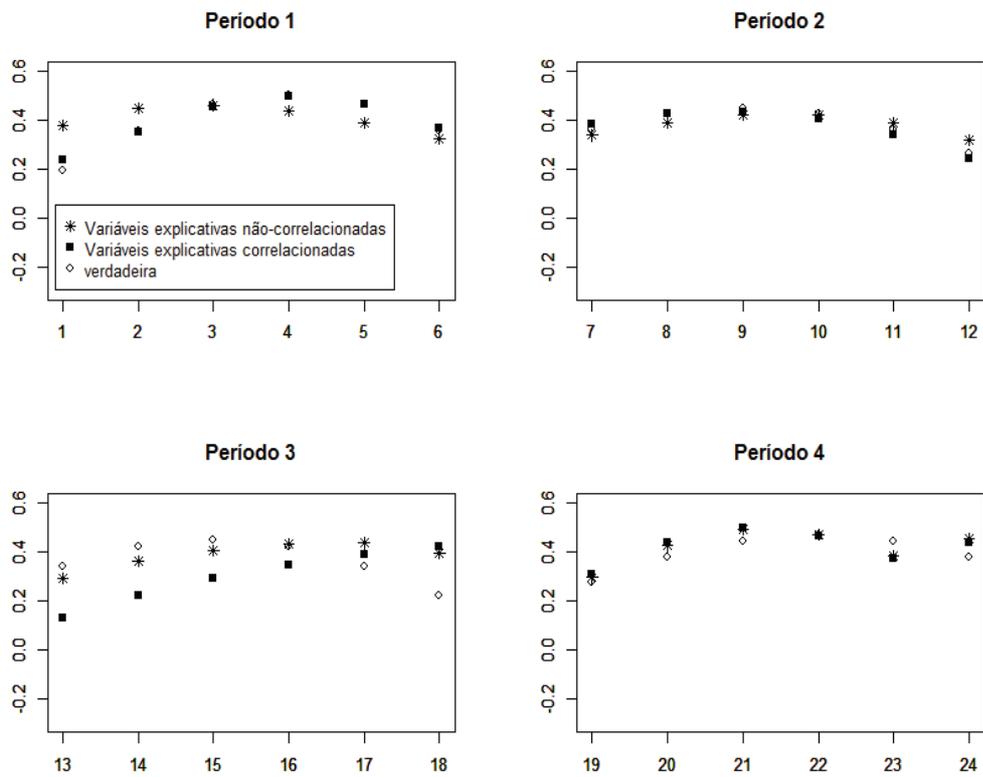


Figura 3.7: HIF's estimadas associadas a Temp, em um modelo com duas variáveis explicativas, que apresentam correlação entre elas. A variável X está associada a quatro HIF's, $\gamma_1(\cdot), \dots, \gamma_4(\cdot)$ e a variável Temp está associada a quatro outras HIF's $\rho_1(\cdot), \dots, \rho_4(\cdot)$.

Tendo em vista que a existência de dependência entre as variáveis explicativas afeta a estimativa das HIF's é necessário eliminar esta dependência antes de realizar a estimação.

Isto será feito da seguinte forma:

- 1a - Regressão funcional de Y_i contra $Temp_i$, de onde obtemos o resíduo $U_{iYTemp1}$
- 1b - Regressão funcional de X_i contra $Temp_i$, de onde obtemos o resíduo $U_{iXTemp1}$
- 1c - Utilizamos os resíduos $U_{iYTemp1}$ e $U_{iXTemp1}$ para obter a primeira estimativa da HIF associada a X , doravante denominada HIFX
- 1d - Regressão funcional de Y_i contra X_i , de onde obtemos o resíduo U_{iYX1}
- 1e - Regressão funcional de $Temp_i$ contra X_i , de onde obtemos o resíduo $U_{iTempX1}$
- 1f - Utilizamos U_{iYX1} e $U_{iTempX1}$ para obter a primeira estimativa da HIF associada a $Temp$, doravante denominada HIFTemp

Obtidas as primeira estimativas das HIFX's, $\gamma_{11}, \dots, \gamma_{41}$ e HIFTemp's, $\rho_{11}, \dots, \rho_{41}$, coletamos as seguintes variáveis:

$$Z_{iX1}(t) = \sum_{k=1}^4 \int_0^6 \gamma_{k1}(u) X_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$$

$$Z_{iTemp1}(t) = \sum_{k=1}^4 \int_0^6 \rho_{k1}(u) Temp_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$$

- 2a - Regressão funcional de Y_i contra Z_{iTemp1} , de onde obtemos o resíduo $U_{iYZTemp2}$
- 2b - Regressão funcional de X_i contra $Temp_i$, de onde obtemos o resíduo $U_{iXTemp2}$
- 2c - Utilizamos $U_{iYZTemp2}$ e $U_{iXTemp2}$ para obter a segunda estimativa da HIF associada a X .
- 2d - Regressão funcional de Y_i contra Z_{iX1} , de onde obtemos o resíduo U_{iYZX2}
- 2e - Regressão funcional de $Temp_i$ contra X_i , de onde obtemos o resíduo $U_{iTempX2}$
- 2f - Utilizamos os resíduos U_{iYZX2} e $U_{iTempX2}$ para obter a segunda estimativa da HIF associada a $Temp$.

Obtemos a segunda rodada de estimativas das HIF's associadas à X e à $Temp$, $\gamma_{12}, \dots, \gamma_{42}, \rho_{12}, \dots, \rho_{42}$, coletamos, então, as seguintes variáveis:

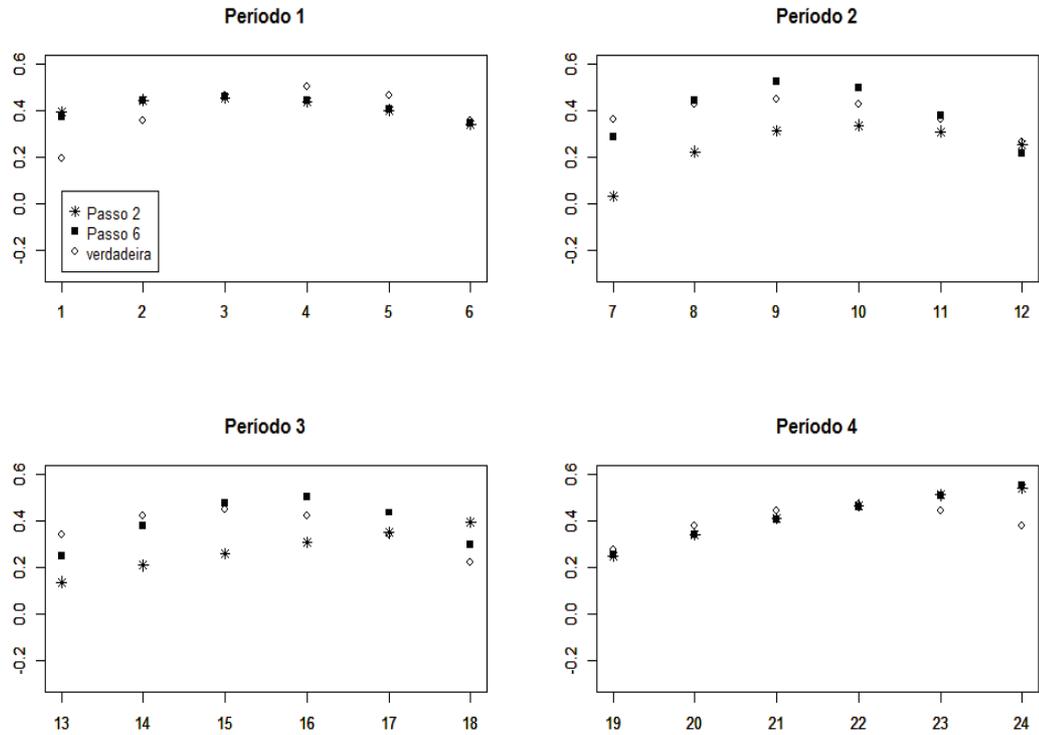


Figura 3-8: HIF's estimadas para a temperatura após o tratamento para a dependência entre as variáveis explicativas.

$$Z_{iX2}(t) = \sum_{k=1}^4 \int_0^6 \gamma_{k2}(u) X_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$$

$$Z_{iTemp2}(t) = \sum_{k=1}^4 \int_0^6 \rho_{k2}(u) Temp_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$$

e repetimos os mesmos passos enumerados em 2a até 2f até obter a convergência. Na figura (3-8), podemos observar o resultado da estimação quando a dependência entre as variáveis explicativas é tratada. As estimativas provenientes do modelo com as modificações acima apresentam uma performance melhor que as primeiras.

3.5 Modelo autorregressivo com variável exógena não simultânea

Nos modelos simulados anteriormente as duas variáveis explicativas são exógenas. Nos modelos utilizados para a carga de energia elétrica, a carga defasada em 24 horas é um regressor importante. Sendo assim, conduziremos um exercício de simulação de um modelo autorregressivo. Além disso, vimos, no capítulo 2, um modelo baseado nas evoluções anuais da carga. Em nossos exercícios de simulação, emularemos um modelo autorregressivo para a evolução de um processo ao longo de oito períodos consecutivos. Por simplicidade, iremos denominar um período de ano. Cada ano é composto de 100 dias, que por sua vez são compostos por 24 horas. Neste modelo as variáveis explicativas serão uma constante, a função dependente defasada [em 24 horas] e uma variável exógena que entra no modelo de maneira não simultânea [i.e. HIF's serão utilizadas].

Na simulação, um dia é uma função com domínio $(0, 24]$ e um período/ano é composto por 100 dias justapostos de maneira contínua. Sendo assim, um ano é uma função contínua com domínio $(0, 2400]$. Simulamos 100 modelos com a seguinte estrutura:

$$Y_{mj}(t) = \alpha(t) + \beta_0(t)\tilde{Y}_{\tilde{m}j} + \beta(t) \sum_{k=1}^{400} \int_0^6 \gamma_k(u) X_{mj}(T_k - u) du \times I_{(T_{k-1}, T_k]}(t) + \varepsilon_{mj}(t) \quad (3.37)$$

em que,

$$\begin{aligned} \tilde{Y}_{\tilde{m}} &= Y_m(t - 24) && \text{if } t \in (24, 2400] \\ \tilde{Y}_{\tilde{m}} &= Y_{m-1}(2400 + t - 24) && \text{if } t \in (0, 24] \end{aligned}$$

em que $m = 1, \dots, 8$, $j = 1, \dots, 100$ e T_k assume os valores 6, 12, 18, 24, \dots , 2400.

Para obter o valor da variável explicativa $\sum_{k=1}^4 \int_0^6 \gamma_k(u) X_{mj}(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$ é preciso definir os valores de T_k e T_{k-1} . Suponha, por exemplo, $t = 1067$, tal valor corresponde às 11 horas do quadragésimo quinto dia⁴ do ano. Sendo assim, $T_k = 12 +$

⁴A unidade de tempo é dada pela parte inteira da divisão de 1067 por 24. E o correspondente s no domínio da unidade de tempo é dado pelo resto da divisão de 1067 por 24.

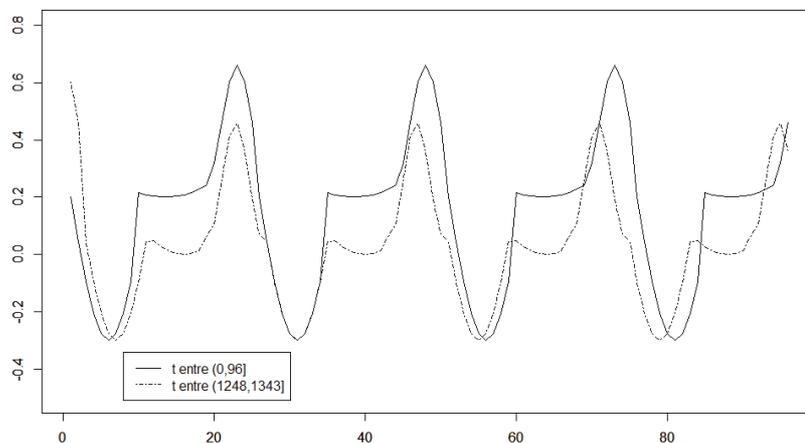


Figura 3-9: Constante avaliada em dois intervalos ao longo do ciclo.

(24×44) e $T_{k-1} = T_k - 6$.

Na figura (3-9), temos o gráfico da constante, $\alpha(t)$, para duas avaliações ao longo do ciclo⁵. O gráfico indicado pela linha tracejada é um deslocamento para baixo e esquerda do gráfico indicado pela linha cheia, emulando os níveis mais baixos de carga no inverno comparativamente ao verão.

Na figura (3-10), podemos observar a função que corresponde à variável exógena [no modelo da carga esta variável é a temperatura]. Ela foi gerada⁶ de acordo com:

$$X_i(t) = g(t) + \xi_i(t) \quad (3.38)$$

em que $\xi_i(t)$ é termo de erro com desvio padrão 0.1.

Na figura (3-11), temos a transformação, via HIF, da variável exógena. Cada um dos níveis é dado por uma integral do tipo: $\int_0^6 \gamma_k(u) X_{mj}(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$, em que k

⁵A fórmula utilizada para obter a contante abaixo pode ser encontrada no apêndice 1.

⁶A função que descreve $g(t)$ pode ser encontrada no apêndice 1.

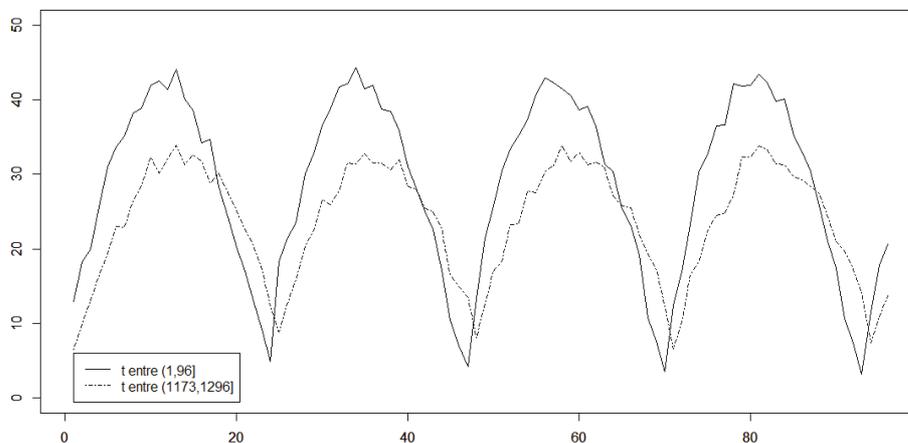


Figura 3.10: Variável exógena avaliada em dois intervalos ao longo do ciclo.

assume os valores 1, 2, 3 e 4. T_k assume os valores 6, 12, 18, 24, \dots , 2400.

Na figura (3.12), temos a função dependente obtida.

Na figura (3.13), temos as HIF's estimadas para cada um dos períodos.

3.6 Continuidade no modelo funcional não simultâneo

O tipo de dado que consideramos neste trabalho são funções que apresentam como característica importante o fato de serem contínuas. Os modelos não simultâneos aqui tratados envolvem funções em patamares [portanto descontínuas], conforme pode ser observado na figura (3.11). A fim de embasarmos uma breve discussão sobre continuidade nos modelos funcionais não simultâneos, escrevemos um modelo que por simplicidade de exposição será para a evolução diária da carga e com apenas uma variável explicativa.

$$Y_i(t) = \beta(t) \sum_{k=1}^4 \int_0^6 \gamma_k(u) X_i(T_k - u) du \times I_{(T_{k-1}, T_k]}(t) + \varepsilon_i(t) \quad (3.39)$$

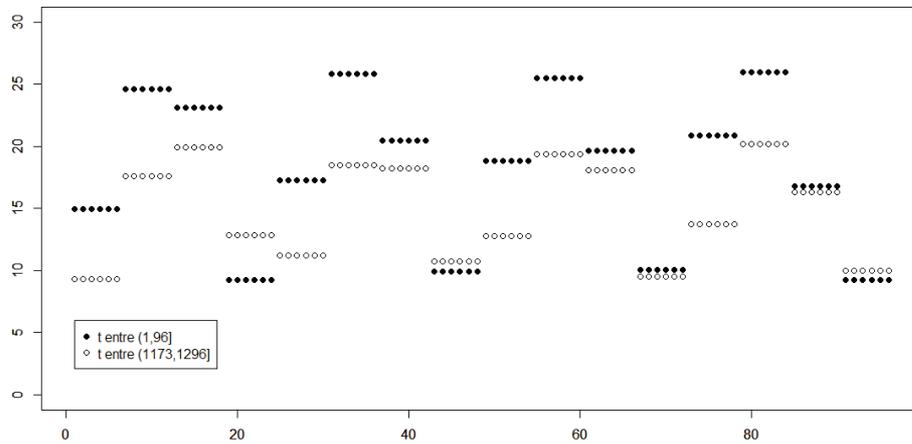


Figura 3-11: Exemplo de patamares obtidos através de integrais do tipo: $\int_0^6 \gamma_k(u) X_{mj}(T_k - u) du \times I_{(T_{k-1}, T_k]}(t)$, em que k assume os valores 1, 2, 3 e 4. T_k assume os valores 6, 12, 18, 24, \dots , 2400.

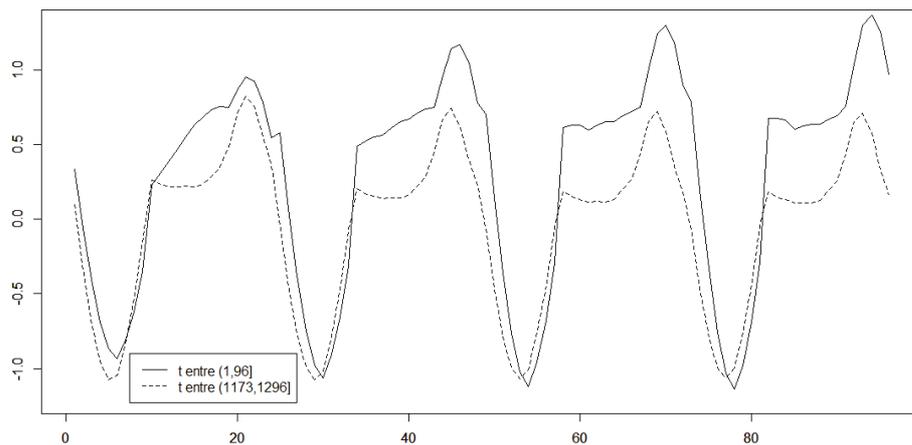


Figura 3-12: Função dependente avaliada em dois intervalos ao longo de um mesmo ciclo.

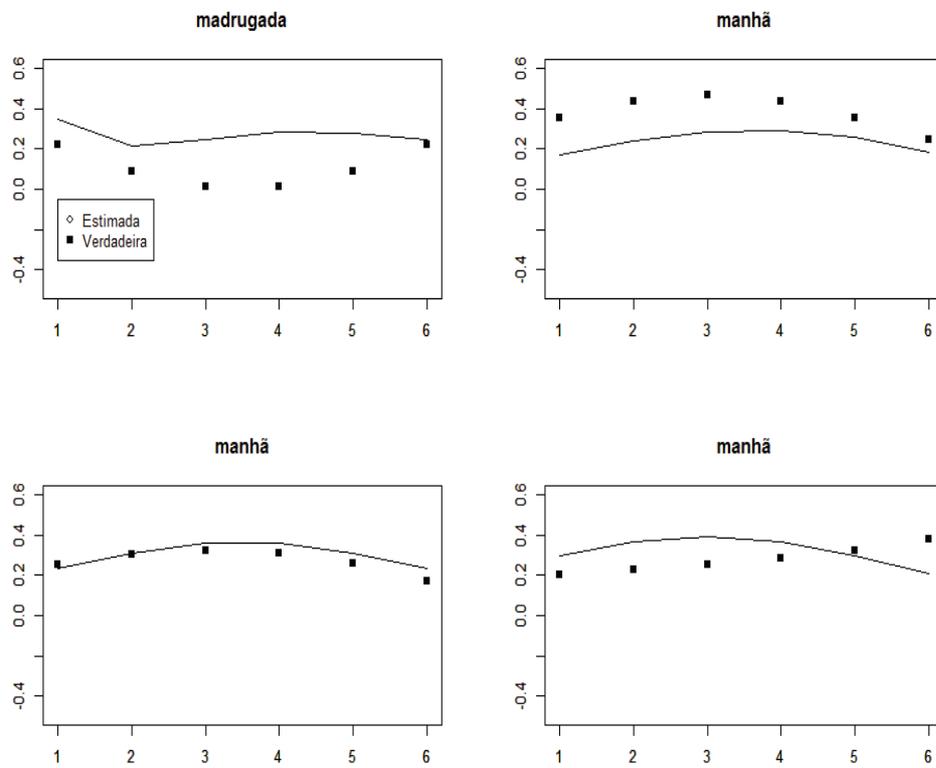


Figura 3-13: HIF's estimadas para madrugadas, manhãs, tardes e noites.

em que $i = 1, \dots, 500$, $k = 1, \dots, 4$, $t \in (0, 24]$ e T_k assume os valores 6, 12, 18, 24.

$$Y_i(6) = \lim_{t \rightarrow 6^+} \beta(t) \int_0^6 \gamma_1(u) X_i(6-u) du = \lim_{t \rightarrow 6^-} \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du$$

$$Y_i(12) = \lim_{t \rightarrow 12^+} \beta(t) \int_0^6 \gamma_2(u) X_i(12-u) du = \lim_{t \rightarrow 12^-} \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du$$

$$Y_i(18) = \lim_{t \rightarrow 18^+} \beta(t) \int_0^6 \gamma_3(u) X_i(18-u) du = \lim_{t \rightarrow 18^-} \beta(t) \int_0^6 \gamma_4(u) X_i(18-u) du$$

Faça $\int_0^6 \gamma_1(u) X_i(6-u) du = C$ e $\int_0^6 \gamma_2(u) X_i(12-u) du = D$. Por um lado temos que $Y_i(6) = \lim_{t \rightarrow 6^+} \beta(t) C$ e por outro que $Y_i(6) = \lim_{t \rightarrow 6^-} \beta(t) D$. Segue daí que:

$$\lim_{t \rightarrow 6^+} \beta(t) C = \lim_{t \rightarrow 6^-} \beta(t) D$$

o que implica:

$$\frac{\lim_{t \rightarrow 6^+} \beta_1(t)}{\lim_{t \rightarrow 6^-} \beta_2(t)} = \frac{D}{C} \quad (3.40)$$

Por (3.40), $Y_i(6)$ só seria contínua se $C = D$. Não é o caso para a carga de energia elétrica. No modelo, impomos uma de restrição de continuidade para $Y_i(\cdot)$ e para $\beta(\cdot)$. Pode ocorrer de o parâmetro $\beta(\cdot)$ variar rapidamente na proximidade do ponto 6 para compensar o salto de C para D e atender a restrição de continuidade de $Y_i(\cdot)$. O mesmo comportamento pode ocorrer nas vizinhanças dos pontos 12 e 18.

A fim de acomodar a descontinuidade subjacente ao modelo não simultâneo vamos projetar os patamares dados por: $\int_0^6 \gamma_1(u) X_i(6-u) du$, $\int_0^6 \gamma_2(u) X_i(12-u) du$, $\int_0^6 \gamma_3(u) X_i(18-u) du$ e $\int_0^6 \gamma_4(u) X_i(24-u) du$ em uma base de splines. Na figura (3-14), exibimos possíveis exemplos dos patamares supracitados e sua projeção em uma base de splines⁷.

⁷A base de splines da figura (3-14) admite primeira derivada descontínua nos pontos 6, 12 e 18.

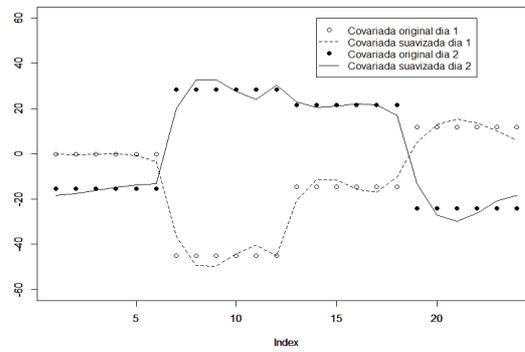


Figura 3.14: Exemplo de projeção em base de splines dos patamares dados

por: $\int_0^6 \gamma_1(u) X_i(6-u) du$, $\int_0^6 \gamma_2(u) X_i(12-u) du$, $\int_0^6 \gamma_3(u) X_i(18-u) du$ e

$$\int_0^6 \gamma_4(u) X_i(24-u) du.$$

Capítulo 4

Modelo funcional não simultâneo aplicado à carga elétrica horária

No segundo capítulo desta tese, modelamos as evoluções anuais e semanais da carga. Neste capítulo, apresentamos versões não simultâneas dos mesmos modelos. A temperatura é a covariada que se relaciona com a carga de maneira não simultânea. Por uma questão de comodidade para o leitor, rerepresentamos, nas seções seguintes, as formas funcionais dos modelos simultâneos para as evoluções semanais e anuais. Simplificações dos modelos estimados no segundo capítulo desta tese são adotadas. Para cada modelo não simultâneo, estimamos uma versão simultânea com vistas a comparações.

4.1 Modelo semanal

No modelo semanal, o domínio das funções é dado pelo intervalo $[0, 168)$. As primeiras 24 horas correspondem ao domingo, as 24 horas seguintes correspondem à segunda-feira e assim por diante. Lembramos que:

$$G_{s,t} := \begin{cases} \sigma(\{Y_s(u); u \leq t - 24\}, Y_{s-1}, Y_{s-2}, \dots, X_s, \dots) & \text{se } t \in (24, 168] \\ \sigma(\{Y_{s-1}(u); u \leq t + 168 - 24\}, Y_{s-1}, Y_{s-2}, \dots, X_s, \dots) & \text{se } t \in (0, 24] \end{cases}$$

e que no segundo capítulo havíamos assumido:

$$\begin{aligned}
E(Y_s(t)|\mathcal{G}_{s,t}) &= \alpha(t) + \beta_1(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+,V}(\vartheta) + \\
&\quad \beta_2(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+,FMA}(\vartheta) + \\
&\quad \beta_3(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+,MJJ}(\vartheta) + \\
&\quad \beta_4(t)Y_{s-(1-\lfloor \frac{t}{24} \rfloor)_+,ASO}(\vartheta) + \gamma(t)W_s(t) + D_s(t)
\end{aligned} \tag{4.1}$$

em que $s = 1, 2, \dots, N$, $\vartheta = 168 * \lfloor \frac{24}{t} \rfloor + t - 24$, onde $\lfloor \cdot \rfloor$ denota a função parte inteira e $(\cdot)_+$ denota a função parte positiva. $Y_{z,V}$ assume os valores da carga somente quando z indica uma semana na qual vale o horário de verão. Caso contrário, $Y_{z,V}$ assume valor zero. $Y_{z,FMA}$ assume os valores da carga somente quando z indica uma semana pertencente a fevereiro[e na qual não vale horário de verão], março e abril. $Y_{z,MJJ}$ segue a mesma regra para aquelas semanas pertencentes a maio, junho e julho. E por fim, $Y_{z,ASO}$ assume os valores da carga somente quando z indica uma semana pertencente a agosto, setembro e outubro[e na qual não vale horário de verão]. $W_s(t)$ é a temperatura, $\alpha_s(\cdot)$ e $D_s(t)$ desempenham a mesma função que uma constante e uma dummy de feriados nos modelos univariados de regressão.

A fim de estimar um modelo funcional não simultâneo para as curvas semanais de carga, adotaremos duas simplificações: 1 - serão excluídos da amostra os sábados, domingos e as segundas-feiras; 2 - O modelo será estimado separadamente para as semanas classificadas como $Y_{z,V}$, $Y_{z,FMA}$, $Y_{z,MJJ}$ e $Y_{z,ASO}$. Dessa forma, para o conjunto das semanas em que vale horário de verão, por exemplo, o modelo que aparece em (4.1) pode ser reescrito da seguinte forma:

$$\begin{aligned}
E(Y_s(t)|\mathcal{G}_{s,t}) &= \alpha(t) + \beta_1(t)Y_s(t - 24) + \\
&\quad \beta_2(t) \sum_{k=1}^{16} \int_0^6 \gamma_k(u)w_s(T_k - u)du \times I_{(T_{k-1}, T_k]}(t) + \varepsilon_s(t)
\end{aligned} \tag{4.2}$$

em que $s = 1, 2, \dots, N$ [a simplificação 2 implica que N corresponde ao número de semanas observadas e nas quais vale horário de verão], $t \in (0, 96]$, $w_s(\cdot)$ é a temperatura e as funções

$\alpha_s(\cdot)$ e $D_s(\cdot)$ desempenham a mesma função que uma constante e uma dummy de feriados nos modelos univariados de regressão. No modelo semanal, T_k assume os valores 6, 12, 18, 24, 30, \dots , 96 e $T_0 = 0$. A hora $t = 63$ corresponde às 15 horas da quinta-feira e (4.2) para essa hora é dada por:

$$E(Y_s(63)) = \alpha(63) + \beta_1(63)Y_s(39) + \beta_2(63) \int_0^6 \gamma_3(u)w_s(66-u)du + \varepsilon_s(63)$$

No modelo semanal não simultâneo, os coeficientes da carga defasada e da temperatura serão penalizados pelo acelerador harmônico $L_1g = (\frac{2\pi}{24})^2 D^2g + g$ que capta os movimentos recorrentes a cada 24 horas apresentados pela carga.

Na figura (4.1), temos as HIF's estimadas para as madrugadas, manhãs, tardes e noites para os quatro grupos de modelos. Algumas características são interessantes. Nas madrugadas de FMA [fevereiro, março e abril] e de ASO [agosto, setembro e outubro] os pesos das horas iniciais são mais altos que das horas finais. O contrário ocorre para as madrugadas de verão e de MJJ [maio, junho e julho]. As horas finais das manhãs de verão apresentam os maiores pesos. As horas centrais das tardes de verão apresentam os maiores pesos. Esse comportamento se inverte em MJJ [as horas do início e final das tardes apresentam pesos maiores comparativamente às horas centrais da tarde]. As horas iniciais das noites de verão e MJJ são mais importantes, para explicar a carga, quando comparadas com as horas iniciais de FMA e ASO.

Nas figuras (4.2), (4.3), (4.4) e (4.5), podemos observar o boxplot das RMSE's associadas a cada hora do dia provenientes dos modelos estimados separadamente de acordo com as classificações Verão, FMA, MJJ e ASO [para cada grupo foi estimado um modelo conforme (4.2)]. A cada hora estão associadas 4 RMSE's correspondentes às terças, quartas, quintas e sextas-feiras. As RMSE's associadas às horas 3, 4, 9 e 10 apresentam medianas mais altas que as demais. Essas horas tem em comum o fato de antecederem a mudança entre períodos do dia [madrugada para manhã e manhã para tarde]. Tal característica

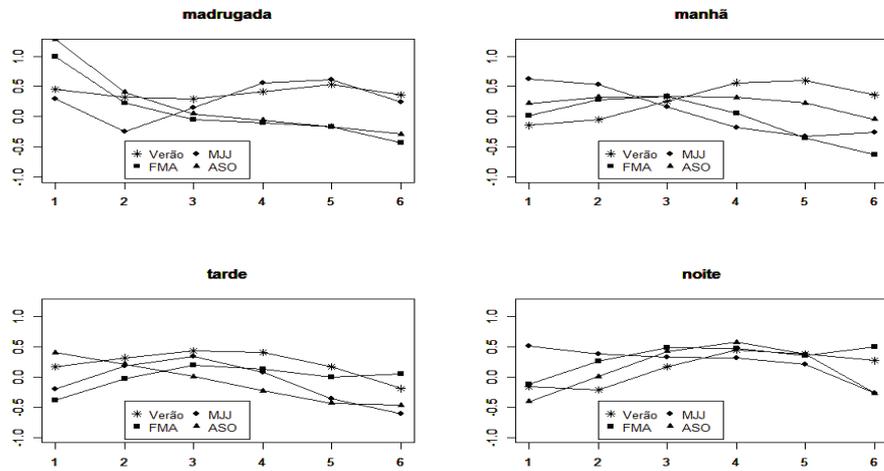


Figura 4.1: HIF's estimadas no modelo semanal para cada período do dia e do ano.

pode ser entendida também uma mudança de concavidade das curvas de carga.

O modelo simultâneo associado a (4.2) pode ser escrito da seguinte forma:

$$E(Y_s(t)|\mathcal{G}_{s,t}) = \alpha(t) + \beta_1(t)Y_s(t-24) + \beta_2(t)w_s(t) + \varepsilon_s(t) \quad (4.3)$$

em que $s = 1, 2, \dots, N$, $t \in (0, 96]$, $w_s(\cdot)$ é a temperatura e as funções $\alpha_s(\cdot)$ e $D_s(\cdot)$ desempenham a mesma função que uma constante e uma dummy de feriados nos modelos univariados de regressão.

Nas figuras (4.2), (4.3), (4.4) e (4.5), podemos observar o *boxplot* das RMSE's associadas a cada hora do dia provenientes dos modelos (4.3) estimados separadamente de acordo com as classificações verão, FMA, MJJ e ASO. A cada hora estão associadas 4 RMSE's correspondentes a terças, quartas, quintas e sextas-feiras.

As figuras (4.2), (4.3), (4.4) e (4.5) indicam que, em termos de RMSE, a média ponderada da temperatura, em que os pesos são dados pela HIF, é tão eficaz quanto a temperatura como covariada para a carga. O modelo não simultâneo permite que sejam

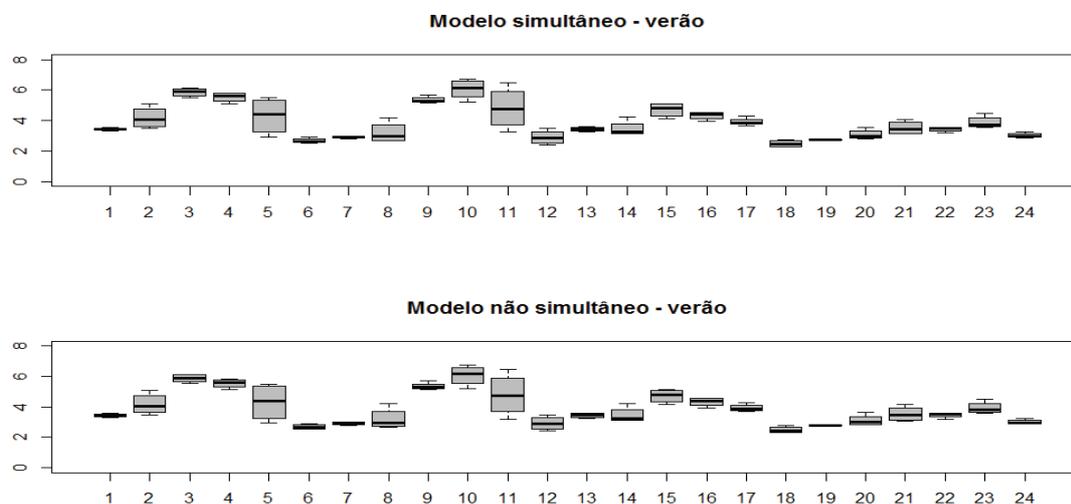


Figura 4-2: RMSE's provenientes dos modelos simultâneo e não simultâneo semanal

captados padrões de temperatura que ocorram em instantes próximos ao instante sendo explicado e não somente a temperatura do próprio [instante]. Além disso, uma potencial vantagem do modelo não simultâneo, no caso em que o modelo for utilizado para previsão, é a não necessidade da previsão da temperatura para cada instante de tempo. Basta a previsão da média ponderada para madrugadas, manhãs, tarde e noites.

4.2 Modelo anual

No modelo anual simultâneo, o domínio das funções é $[0, 8760)$. Para isso lembramos que:

$$\mathcal{G}_{m,t} := \begin{cases} \sigma(\{y_m(u); u \leq t - 24\}, y_{m-1}, y_{m-2}, \dots, w_m, \dots) & \text{se } t \in (24, 8760] \\ \sigma(\{y_{m-1}(u); u \leq t + 8760 - 24\}, y_{m-1}, y_{m-2}, \dots, w_m, \dots) & \text{se } t \in (0, 24] \end{cases}$$

e que :

$$E(y_m(t)|\mathcal{G}_{m,t}) = \alpha(t) + \beta_1(t)\tilde{y}_{\tilde{m}} + \gamma(t)w_m(t) + D_m(t). \quad (4.4)$$

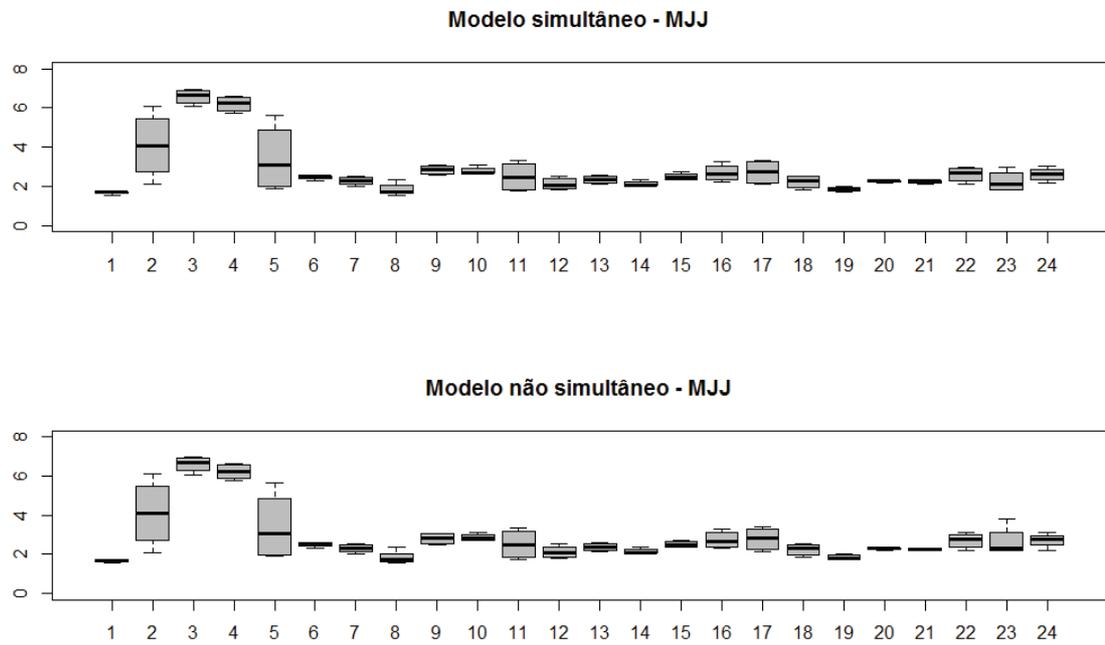


Figura 4-3: RMSE's provenientes dos modelos simultâneo e não simultâneo semanal

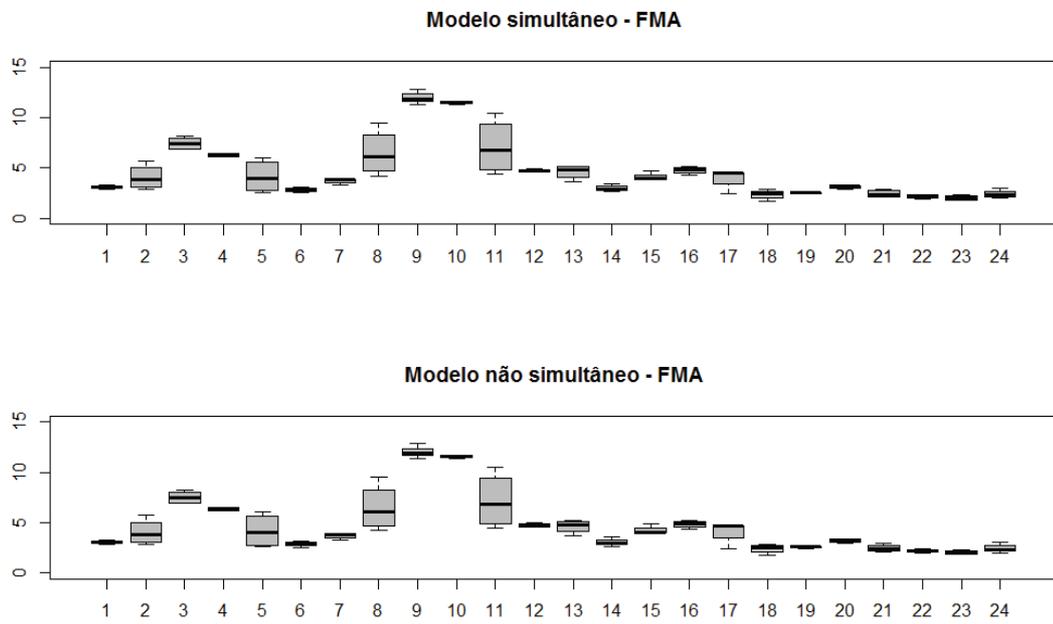


Figura 4-4: RMSE's provenientes dos modelos simultâneo e não simultâneo semanal

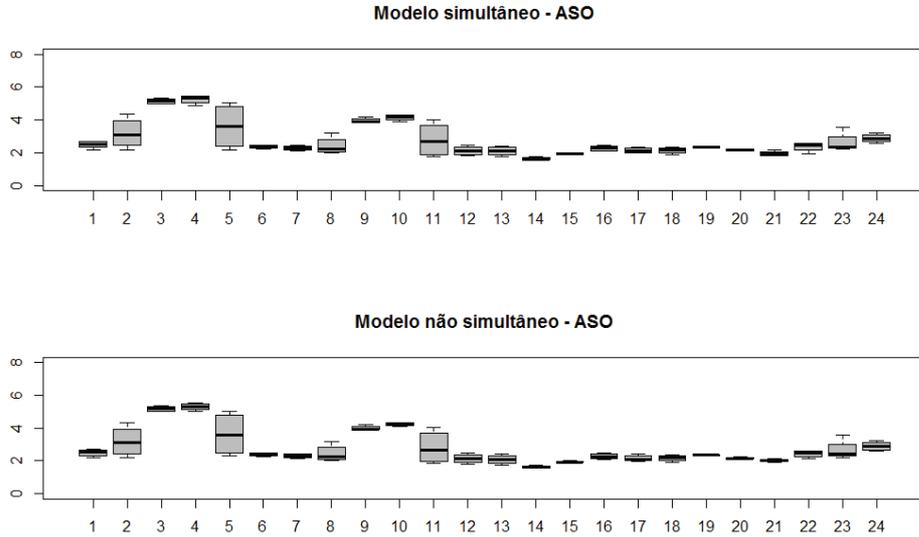


Figura 4-5: RMSE's provenientes dos modelos simultâneo e não simultâneo semanal.

A função $\tilde{y}_{\tilde{m}}$ é dada por:

$$\begin{aligned}\tilde{y}_{\tilde{m}} &= y_m(t - 24) && \text{se } t \in (24, 8760] \\ \tilde{y}_{\tilde{m}} &= y_{m-1}(4232 + t - 24) && \text{se } t \in (0, 24]\end{aligned}$$

e $m = 1, \dots, N$, $t \in [0, 8760]$; 8760 é o número de horas observadas ao longo de um ano. A variável $\tilde{y}_{\tilde{m}}$ representa a carga corrente $[y_m(t)]$ defasada em 24 horas. A função $w_m(\cdot)$ é a temperatura. As funções $\alpha(\cdot)$ e $D_m(\cdot)$ desempenham a mesma função que a constante e uma dummy para feriados nos modelos tradicionais de regressão.

O modelo será aplicado às 51 semanas completas e interiores do ano, optamos também por excluir da amostra os sábados, domingos e segundas-feiras [as segundas-feiras continuam sendo covaridas das terças feiras, elas foram excluídas somente como variáveis dependentes]. Estes dias apresentam um formato diferente para a curva de carga e é provável que também exibam HIF's diferentes. Tais implicam que $t \in (0, 4896]$.

As HIF's são estimadas para madrugadas, manhãs, tardes e noites das terças, quartas,

quintas e sextas-feiras. Elas variam de acordo com os meses do ano. Isto é, a HIF estimada para as madrugadas de janeiro podem diferir daquelas estimadas para as madrugadas de fevereiro. Sendo assim, são estimadas um total de de 48 HIF's.

Apenas a temperatura entra no modelo de maneira não simultânea¹, conforme abaixo:

$$Y_m(t) = \alpha(t) + D_m(t) + \beta_0(t)\tilde{Y}_{\tilde{m}} + \beta_1(t) \sum_{k=1}^{816} \int_0^6 \gamma_{kj}(u)w_m(T_k - u)du \times I_{(T_{k-1}, T_k]}(t) + \varepsilon_m(t) \quad (4.6)$$

em que:

$$\begin{aligned} \tilde{y}_{\tilde{m}} &= y_m(t - 24) & \text{se } t \in (24, 4896] \\ \tilde{y}_{\tilde{m}} &= y_{m-1}(4896 + t - 24) & \text{se } t \in (0, 24] \end{aligned} ,$$

$m = 1, \dots, 8$, $j = 1, \dots, 12$, $t \in [0, 4896)$, $\tilde{y}_{\tilde{m}}$ representa a carga corrente $[y_m(t)]$ defasada em 24 horas. A função $w_m(\cdot)$ é a temperatura. As funções $\alpha_m(\cdot)$ e $D_m(\cdot)$ desempenham o mesmo papel que a constante e uma dummy para feriados nos modelos tradicionais de regressão. T_k assume os valores 6, 12, 18, \dots 2400 e $T_{k-1} = T_k - 6$ com $T_0 = 0$.

Para fins didáticos faça $t = 1067$ no modelo que aparece em (4.6). Tal valor de t corresponde às 11 horas do quadragésimo quinto dia² considerado no ano. O valor de T_k neste caso é dado por $T_k = 12 + (24 \times 44) = 1068$ e o de $T_{k-1} = 1062$. O quadragésimo quinto dia do ano é um dia do mês de fevereiro. Assim para $t = 1067$ a expressão (4.6) é :

¹Para pontuar a diferença entre o nosso modelo e o de Senturk e Muller considere a seguinte expressão:

$$E[Y(t)|X(u) \ u \in (0, T)] = \beta_0(t) + \beta_1(t) \int_0^{\Delta} \gamma(u)X(t - u)du \quad (4.5)$$

para $t \in [\Delta, T]$. Para um t^* fixado, tomamos uma espécie de média ponderada dos valores de $X(t)$ para $t \in (t^*, t^* - \Delta)$ como regressor para $Y(t^*)$. Independentemente do t^* a HIF é sempre a mesma, além disso estamos sempre olhando para valores t menores que t^* o que não é necessariamente o nosso caso.

²O dia do ano é dado pela parte inteira da divisão de 1067 por 24. E a correspondente hora do dia é dada pelo resto da divisão de 1067 por 24.

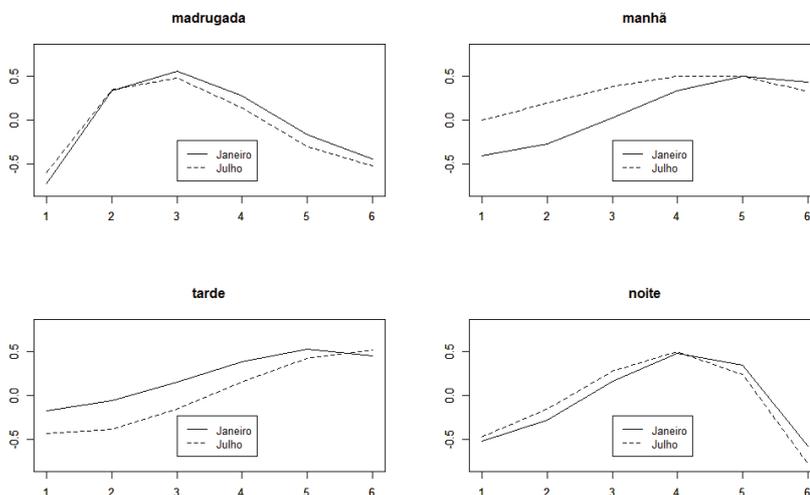


Figura 4-6: HIF's estimadas para a temperatura no modelo anual.

$$Y_m(1067) = \alpha(1067) + \beta_0(1067)\tilde{Y}_m + \beta(1067) \int_0^6 \gamma_{22}(u)w_m(1068-u)du + \varepsilon_m(1067) \quad (4.7)$$

Na figura (4-6), podemos observar as HIF's da temperatura estimadas para os meses de janeiro e julho. Podemos observar que assim como os parâmetros funcionais do modelo simultâneo, as HIF's sofrem pequenas alterações de um mês de verão para um mês de inverno.

No modelo (4.6) são necessárias ainda algumas considerações acerca de sazonalidade. A periodicidade anual (sazonalidade) é devida basicamente a fatores climáticos. Por exemplo, terças-feiras de inverno exibem formas típicas distintas das terças-feiras de verão [cf. capítulo 2]. Em função disso, estabelecemos que o parâmetro da temperatura seja penalizado pelo acelerador harmônico $L_1g = \left(\frac{24 \times 4 \times 51}{2\pi}\right)^2 g + g$ que capte os movimentos induzidos na carga por menores e maiores temperatura no inverno e no verão,

respectivamente.

A carga defasada e a carga apresentam movimentos recorrentes a cada 24 horas, por exemplo, afóra nos dias mais quentes de verão, o horário de pico ocorre no início das noites e do horário de menor consumo durante as madrugadas. Sendo assim, é razoável que o parâmetro associado à carga defasada, $\beta(t)$, apresente um comportamento periódico, com período 24. Possíveis fugas deste comportamento, pelo parâmetro $\beta(t)$, serão penalizadas pelo acelerador harmônico $L_1g = (\frac{2\pi}{24})^2 D^2g + g$.

Nas figuras (4.7), (4.8), (4.9), e (4.10), temos a distribuição das RMSE's do modelo funcional simultâneo e não simultâneo. As RMSE's foram classificadas de acordo com verao, FMA, MJJ e ASO. Em termos de RMSE, a média ponderada da temperatura, em que os pesos são dados pela HIF, é tão eficaz quanto a temperatura como covariada para a carga. O modelo não simultâneo apresenta como principal vantagem o fato de ser mais parcimonioso.

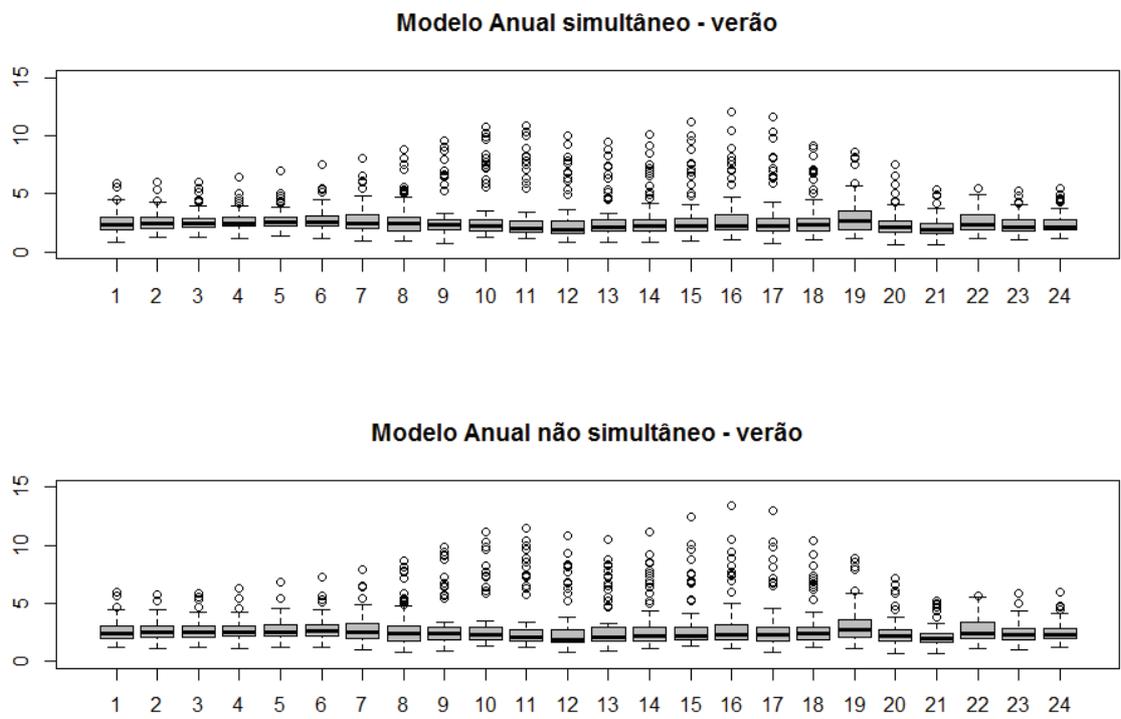


Figura 4-7: RMSE's provenientes dos modelos simultâneo e não simultâneo anual. Os valores estão em %.

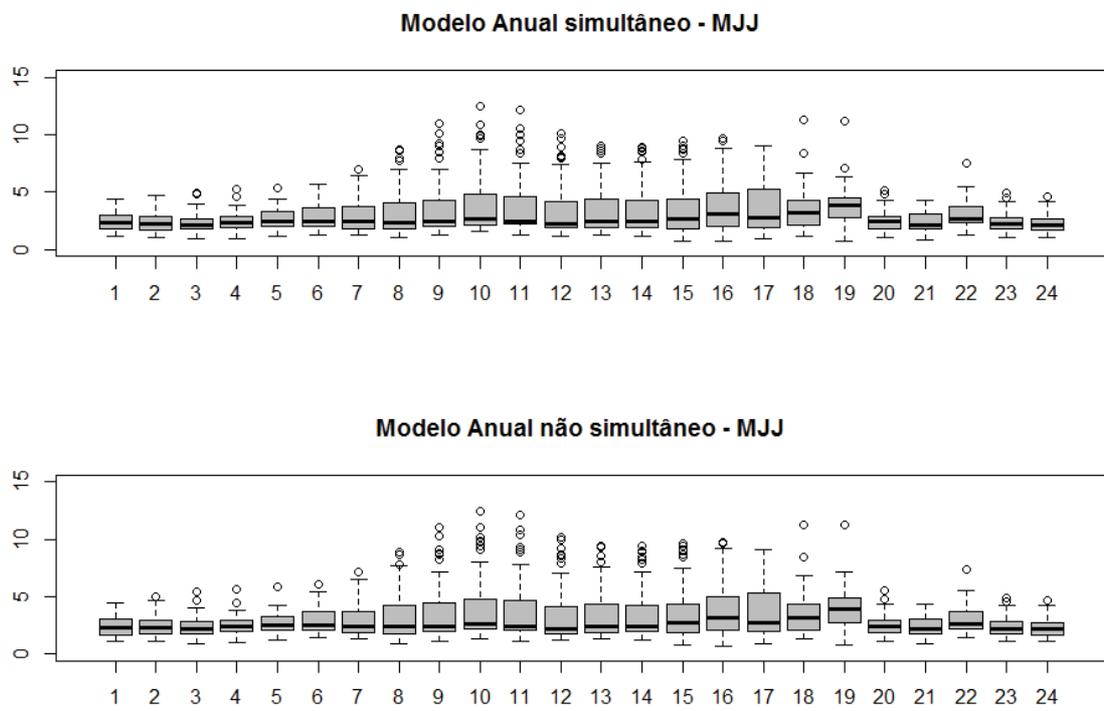


Figura 4-8: RMSE's provenientes dos modelos simultâneo e não simultâneo anual. Os valores estão em %.

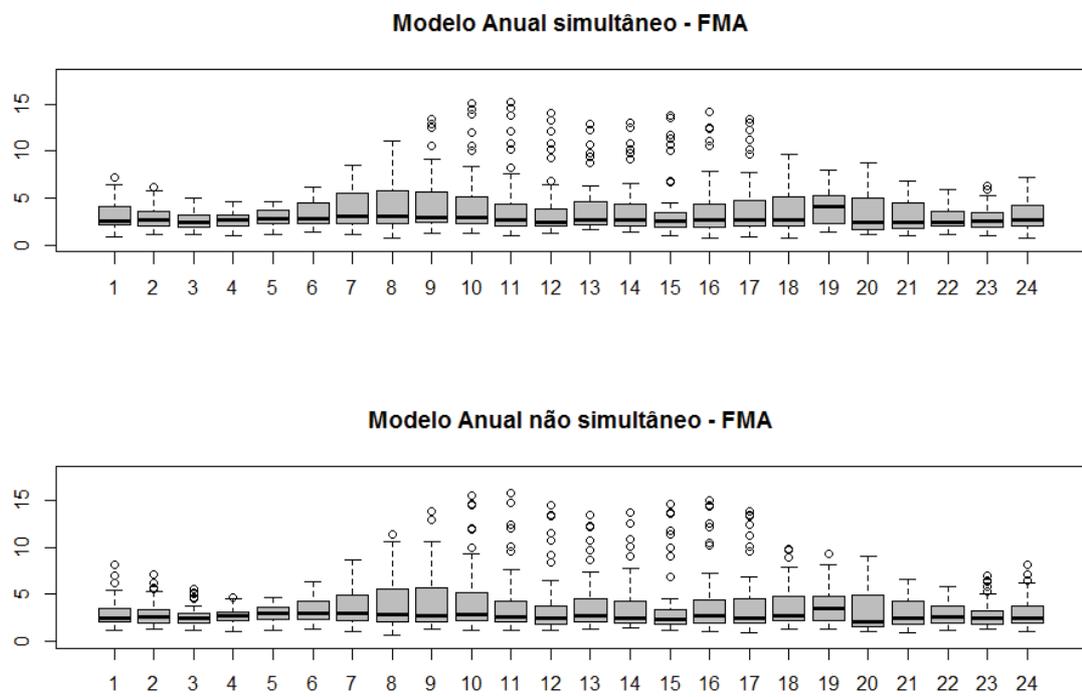


Figura 4-9: RMSE's provenientes dos modelos simultâneo e não simultâneo anual. Os valores estão em %.

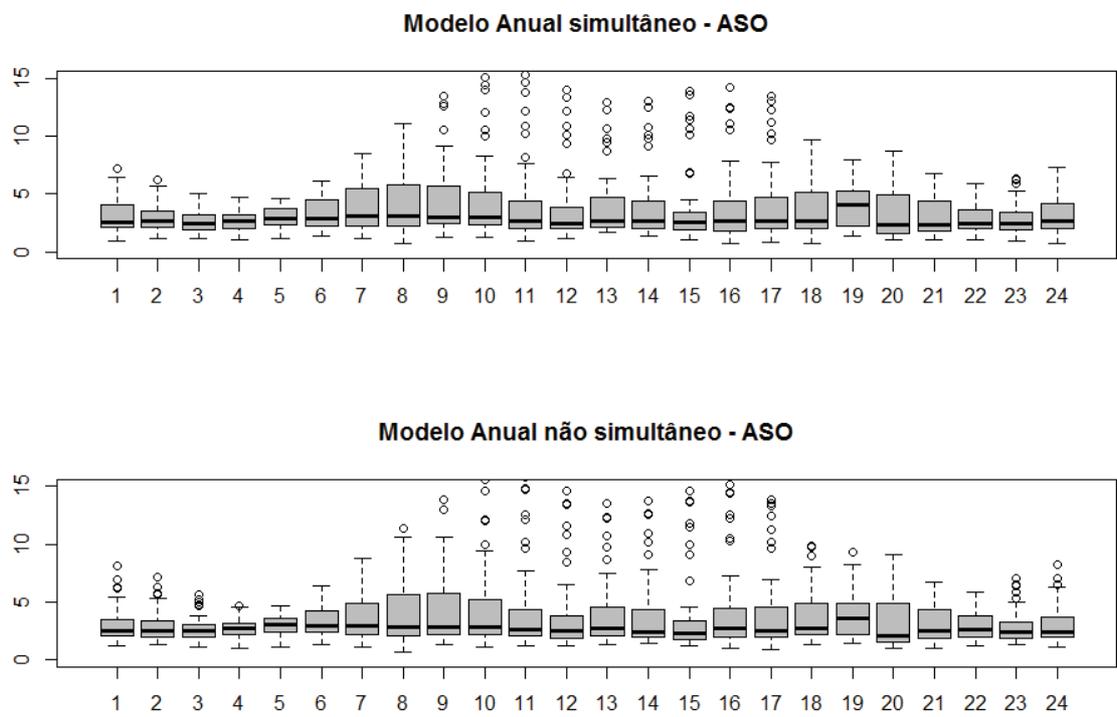


Figura 4-10: RMSE's provenientes dos modelos simultâneo e não simultâneo anual. Os valores estão em %.

Conclusão

Na tese desenvolvemos modelos para lidar com processos estocásticos que exibam três características básicas: (1) inércia, (2) ciclos e (3) dados funcionais. Por apresentar tais características e ser importante do ponto de vista econômico, a carga horária de energia elétrica foi escolhida como exemplo de aplicação de nossa metodologia.

Inicialmente foram desenvolvidos três modelos de regressão funcional do tipo simultâneo. O primeiro deles, denominado modelo diário, é do tipo periódico funcional e está baseado nas evoluções diárias da carga. Nesse modelo, as variações sazonais não induzidas por temperatura são captadas através da inclusão de determinadas senóides e cossenóides. Este modelo apresentou como potencial desvantagem a possível descontinuidade das previsões para os finais e início de cada dia da semana.

O segundo modelo, denominado semanal, está baseado nas evoluções semanais da carga. Os parâmetros desse modelo variam com as estações do ano de forma a captar a sazonalidade não induzida pela temperatura. O modelo semanal deixou claro que os parâmetros mudam de acordo com a estação do ano e que essa mudança ocorre de maneira suave. Essa suavidade nos levou a especificação de um modelo baseado nas evoluções anuais da carga. O terceiro modelo, anual, apresenta como principal vantagem o fato dos parâmetros se adaptarem ao longo do ano para captarem as mudanças induzidas pela sazonalidade. Ele também apresentou melhores RMSE's que os dois modelos anteriores. A principal desvantagem do modelo anual está associada à necessidade de registro dos dados via mudança da escala do tempo para lidar com o fato de um ano não ser constituído de um número inteiro de semanas. Por se tratarem de séries temporais esse tipo de registro pode gerar algum tipo de viés.

Os três modelos anteriores são do tipo simultâneo. Baseados nos modelos desenvolvidos por Ramsay & Silverman(2010) e Senturk & Muller (2010), propusemos que a variável dependente avaliada no tempo t passasse a se relacionar com uma média ponderada da variável explicativa em um intervalo de tempo apropriado, contendo t . Subjacente a tal modelo, obtivemos um estimador que pode ser considerado uma versão funcional do

estimador de posto reduzido em modelos lineares. Esses novos modelos foram denominados modelos funcionais não simultâneos.

O modelo funcional não simultâneo foi aplicado às evoluções semanais e anuais da carga. Uma potencial vantagem do modelo não simultâneo, no caso em que o modelo for utilizado para previsão, é a não necessidade da previsão da temperatura para cada instante de tempo. Basta a previsão da média ponderada para madrugadas, manhãs, tarde e noites.

Um desenvolvimento futuro importante diz respeito ao registro dos dados. Nos dados da carga identificamos o tipo de problema tratado em Ramsay & Silverman (2010) relacionado a registro. Essa é uma futura linha de pesquisa.

Bibliografia

- Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*, 2^a ed. Wiley, New York.
- Anderson, T. W. (2001). Reduced rank regression in cointegrated models. *Journal of Econometrics*, 106, 203-216.
- Bates, D. M. and Watts, D. G. (1988). *Nonlinear regression analysis and its applications*. John Wiley & Sons, New York.
- Box, G. and Jenkins, G. (1970). *Time series Analysis: Forecasting and control*, San Francisco: Holden-Day.
- Bosq, D. (2000). *Linear processes in function spaces*. Springer-Verlag, New York.
- Bunn, D. W. and Farmer, E. D. (1985). *Comparative Models for Electrical Load Forecasting*. John Wiley & Sons.
- Cai, Z., Das, M., Xiong, H. and Wu, X. (2006). Functional coefficient instrumental variables models. *Journal of Econometrics*, 133, 207-241.
- Caldeira, J. F. e Torrent, H. (2011). Previsão de Curvas de juros Zero-Cupom: Estimação Não-paramétrica de Dados Funcionais. <http://bibliotecadigital.fgv.br/ocs/index.php/sbe/EBE11/paper/view/2946/1340>.
Última visita em 12/12/2011.
- Cancelo, J. R., Espasa, A. e Grafe, R.(2008). Forecasting the electricity load from one day to one week ahead for Spanish system operator. *International Journal of forecasting*, 24, 588-602.
- Cottet, R. and Smith, M. (2003). Bayesian Modeling and Forecasting of Intraday Electricity Load. *Journal of the American Statistical Association*, 98, 839-849.
- Davidson, R. and Mackinnon, J. (2004). *Econometric Theory and Methods*. New York.
- Dordonnat, V., koopman, S.J., Ooms, M., Dessertaine, A. e Collet, J. (2008). A hourly Periodic State Space Model for Modelling French National Electricity Load. *International journal of forecasting*, 24, 566-587.
- Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and its Applications*.

Chapman & Hall. London.

Franses, Philip H. and Paap (2004), Richard. Periodic Time Series models. Advanced Texts in econometrics. Oxford university Press, New York.

Engle, R. F., Granger, C. W. J., Rice, J. e Weiss, A. (1986). Semiparametric estimates of the relation between weather and electricity sales. Journal of the American Statistical Association, 81, 310-320.

Györfi, L., Kohler, M., Krzyzak, A. e Walk, H. (2002). A Distribution-Free Theory of Nonparametric Regression. Springer series in statistics. Springer-verlag, New York.

Härdle, W. (1990). Applied Nonparametric Regression. Cambridge University Press, cambridge.

Seber, G. A. F and Wild, C. J. (2003). Nonlinear Regression. John Wiley & Sons, New York

Gnanadesikan, R. (1997). Methods for Statistical Data Analysis of Multivariate Observations. Wiley Series in Probability and Statistics. New York.

Good, P. (2005). Permutation, Parametric and Bootstrap Tests of Hypotheses. Springer. New York.

Green, P.J. and Silverman, B. W., Nonparametric Regression and Generalized Linear Models: A roughness penalty approach. Monographs on Statistics and Applied Probability (1994). Chapman & Hall.

Harvey, A. and Koopman, S. J. (1993). Forecasting hourly electricity demand using time-varying splines. Journal of the American Statistical Association, 88, 566–587.

Hillmer, S.C. e Tiao, G.C. (1982). An ARIMA-Model-Based Approach to Seasonal Adjustment. Journal of the American Statistical Association, 77, 63-70.

Hyde, O. e Hodnett, P. F. (1997). An adaptable automated procedure for short-term electricity load forecasting. IEEE Transactions on Power Systems. 12, 84-94.

Izenman, A. J. (1975). Reduced-Rank Regression for the Multivariate Linear Model. Journal of Multivariate Analysis, 5, 248-264.

Lawton, W.H., Sylvestre, E. A. and Maggio, M.S. Self Modeling Nonlinear Regression.

Technometrics, Vol. 14, n° 3.

Leurgans, S. E. R., Moyeed, A. and Silverman, B. W. (1993). Canonical Correlation Analysis when the Data are Curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55, 725-740.

Litterman, R e Scheinkman, J., "Common Factors Affecting Bond Returns". *The Journal of Fixed Income*, 1991, 54-61.

Loader, C.(1999). *Local Regression and likelihood*. Springer, New York.

Ramanathan, R., Engle, R., Granger, C. W. J., Varid-Araghi, F. e Brace, C. (1997). Shorte-run forecasts of electricity loads and peaks. *International journal of forecasting*, 13, 161-174.

Ramsay, J. O. e Dalzell, C. J. (1991). Some Tools for Functional Data Analysis. *Journal of the Royal Statistical Society. SeriesB (Methodological)*, 53, 539-572.

Ramsay, J.O. e Silverman, B.W. (1997). *Functional Data Analysis*. Springer Series in Statistics. Springer-Verlag, New York.

Ramsay, J.O. and Silverman, B.W. (2010). *Functional Data Analysis*. Springer Series in Statistics. Springer-Verlag, New York.

Ramsay, J.O., Hooker, Giles and Graves, Spence (2009). *Functional Data Analysis with R and MatLab*. New York.

Reinsch, C. H. (1967). Smoothing by Spline Functions. *Numerische Mathematik*, 10, 177-183.

with an Application to Brazilian Data. *International journal of forecasting*, vol. 24.

Rice, J. A. and Silverman, B.W. (1991). Estimating the Mean and Covariance Structure Nonparametrically When the Data are Curves. *Journal of the Royal Statistical Society. Series B*, 53, 233-243.

Seber, G. A. F. e Wild, C. J. (2003). *Nonlinear Regression*. John Wiley and Sons, New Jersey.

Sentruk, D., Muller, H. G. (2010). Functional Varying Coefficient Models for Longitudinal Data. *Journal of the American Statistical Association*, 105, 1256-1264.

Silveira, Getulio Borges. Bessada, Octavio. Banco Central do Brasil. Trabalhos para discussão n°73. Análise de Componentes Principais de Dados Funcionais: Uma aplicação às Estruturas a Termo das Taxas de Juros. Maio, 2003. <http://www.bcb.gov.br/pec/wps/port/wps73.pdf>. Última visita em 30/09/2011.

Smith, M. (2000). Modeling and short-term forecasting of New South Wales electricity system load. *Journal of Business and Economic Statistics*, 18, 465-478.

Soares, L. J. e Medeiros, M. (2008). Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data. *International Journal of Forecasting*, 24, 630-644.

Soares, L. J. e Souza, L. R. (2006). Forecasting electricity demand using generalized long memory. *International Journal of Forecasting*, 22, 17-28.

Tso, M. K. -S. (1981) Reduced-Rank Regression and Canonical Analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43, 183-189

Valor, E., Meneu, V., e Caselles, V. (2001). Daily air temperature and electricity load in Spain. *Journal of Applied Meteorology*, vol 40, 1413-1421.

Wand, M. P. and Jones, M.C. (1995). *Kernel Smoothing*. Chapman & Hall

Yao, F., Müller, H.-G. and Wang, J.-L. (2005). Functional Linear Regression Analysis for Longitudinal Data. *The Annals of Statistics*, 33, 2873-2903.

Apêndice 1

Formulas seção 4.4.1

Forma funcional de $f(t)$ que aparece em (3.32).

$$\begin{aligned}
 f(t) &= 0.5 \times \sin\left(-\frac{\pi}{2} + \frac{\pi}{2} \times \frac{(t-4)}{5}\right) + 0.5 \text{ para } t \in (0, 9] & (4.8) \\
 f(t) &= \sqrt{0.005 \times (t-10) + 0.36} \text{ para } t \in (9, 18] \\
 f(t) &= 0.5 \times \sin\left(\frac{\pi}{2} + \frac{\pi}{2} \times \frac{(t-20)}{1.2}\right) + 0.5 \text{ para } t \in (18, 20] \\
 f(t) &= \sin\left(1.9 \times \frac{(t-18)}{4}\right) \text{ para } t \in (21, 24]
 \end{aligned}$$

Formas funcionais de $\gamma_1(t)$, $\gamma_2(t)$, $\gamma_3(t)$ e $\gamma_4(t)$ que aparecem em (3.33).

$$\begin{aligned}
 \gamma_1(u) &= \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-4)}{4}\right)\right)}{\sqrt{\int_1^8 \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-4)}{4}\right)\right)\right]^2 du}} \quad u \in (1, 6] \\
 \gamma_2(u) &= \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-9)}{5}\right)\right)}{\sqrt{\int_5^{14} \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-9)}{5}\right)\right)\right]^2 du}} \quad u \in (6, 12] \\
 \gamma_3(u) &= \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-15)}{4.5}\right)\right)}{\sqrt{\int_{11}^{20} \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-15)}{4.5}\right)\right)\right]^2 du}} \quad u \in (12, 17] \\
 \gamma_4(u) &= \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-22)}{5}\right)\right)}{\sqrt{\int_{17}^{24} \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(u-22)}{5}\right)\right)\right]^2 du}} \quad u \in (17, 24]
 \end{aligned}$$

Forma funcional do $\beta(t)$ que aparece em (3.34).

$$\beta(t) = \sum_{i=1}^n \frac{X_i(t)}{\int_0^6 \gamma_1(u) X_i(6-u) du} \Big/ n \text{ para } t \in (1, 6]$$

$$\beta(t) = \sum_{i=1}^n \frac{X_i(t)}{\int_0^6 \gamma_2(u) X_i(12-u) du} \Big/ n \text{ para } t \in (6 : 12]$$

$$\beta(t) = \sum_{i=1}^n \frac{X_i(t)}{\int_0^6 \gamma_3(u) X_i(18-u) du} \Big/ n \text{ para } t \in (12, 18]$$

$$\beta(t) = \sum_{i=1}^n \frac{X_i(t)}{\int_0^6 \gamma_3(u) X_i(24-u) du} \Big/ n \text{ para } t \in (18, 24]$$

Fórmulas seção 4.4.2

Forma funcional para os ρ 's que aparecem em (3.35):

$$\rho_1(s) = \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-3)}{5}\right)\right)}{\sqrt{\int_1^6 \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-3)}{5}\right)\right)\right]^2 ds}} \quad s \in (1, 6]$$

$$\rho_2(s) = \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-9.5)}{4.8}\right)\right)}{\sqrt{\int_6^{12} \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-9.5)}{4.8}\right)\right)\right]^2 ds}} \quad s \in (6, 12]$$

$$\rho_3(s) = \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-15.9)}{5}\right)\right)}{\sqrt{\int_{12}^{18} \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-15.9)}{5}\right)\right)\right]^2 ds}} \quad s \in (12, 18]$$

$$\rho_4(s) = \frac{\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-33)}{8}\right)\right)}{\sqrt{\int_{18}^{24} \left[\sin\left(\frac{\pi}{2} \times \left(1 + \frac{(t-33)}{8}\right)\right)\right]^2 ds}} \quad s \in (18, 24]$$

Fórmulas seção 4.5

Forma funcional da função $\alpha(t)$ que aparece em (3.37):

$$\left(\begin{array}{l} 0.5 \sin\left(\frac{-\pi}{2} + \frac{\pi}{2} \frac{(s-5)}{5}\right) I_{(0,9]}(s) + \\ \left(\sin\left(\frac{-\pi}{2} + \frac{\pi}{2} \frac{(s-11.5)}{30}\right) + 1\right) I_{(9,18]}(s) + \\ 0.2 \sin\left(\frac{\pi}{2} \frac{(s-11.5)}{30}\right) + 1.3 \end{array} \right) I_{(18,24]}(s) \right) + c$$

em que $c = 0.2$ para os 25 primeiros dias que compõem o ciclo e $c = 0.1$ para os dias 26 a 50.

$$\left(\begin{array}{l} 0.5 \left(\sin\left(\frac{-\pi}{2} + \frac{\pi}{2} \frac{(s-4)}{5}\right) + 0.2\right) I_{(0,9]}(s) + \\ \left(\sin\left(\frac{-\pi}{2} + \frac{\pi}{2} \frac{(s-14)}{30}\right) + 1\right) I_{(9,16]}(s) + \\ 0.2 \sin\left(\frac{\pi}{2} \frac{(s-19)}{30}\right) + 1.3 \end{array} \right) I_{(16,24]}(s) \right) + c$$

em que $c = 0$ para os dias 51 a 75 e $c = 0.1$ para os dias 76 a 100 que compõem o ciclo.

Forma funcional da função $g(t)$ que aparece em (3.38):

$$g(t) = 30 \left(\sin\left(\pi \frac{t}{21}\right) + 0.25 \right) + 5, 25$$

para os 25 primeiros dias que compõem o ano.

$$g(t) = \frac{55}{2} \left(\sin\left(\pi \frac{t}{21}\right) + 0.25 \right) + 4, 25$$

para os 25 dias 26 a 50 que compõem o ano.

$$g(t) = 25 \left(\sin\left(\pi \frac{t}{21}\right) + 0.25 \right) + 2, 25$$

para os dias 51 a 75 que compõem o ano.

$$g(t) = \frac{57}{2} \left(\sin\left(\pi \frac{t}{21}\right) + 0.25 \right) + 3, 25$$

para os dias 76 a 100 que compõem o ano.

Apêndice 2

Programação do modelo diário

```

#-----
#Dias utilizados na modelagem - Domingo
#-----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")
#-----
# Criando uma Dummy de feriados
#-----
D1= ehFeriadoMBR(Dias)
D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
#-----
#Semanas completas nos dias observados
#-----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
#-----
#Pegando os dados de Temperatura apenas para semanas completas
#-----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,
DiasSemCom),3:26]))),168)
#-----
#Pegando os dados de Temperatura apenas para semanas completas

```

```

#-----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
#-----
#Transformando os dados em relação a média da
#semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
CargaSemCom))[-1]),168),,168)))
#-----
#Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
#-----
# Restringido os dados transformados para semanas completas
# a partir de 2003. Para resolver os problemas induzidos pelo
#racionamento de energia
#-----
Temperatura= matrix(c(TempComp),24)[
,is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24)[
,is.element(DiasSemComTransf,Dias)]

```

```

LagCarga= matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias-1)]
#-----
#Pegando a carga do domingo
#-----
CargaDomDfer= Carga[,weekdays(Dias)==
"domingo"]
#-----
#Pegando a Temperatura do domingo
#-----
TempDomDfer= Temperatura[,weekdays(Dias)==
"domingo"]
#-----
#covariada do domingo
#-----
CovDomDfer= LagCarga[,weekdays(Dias-1)==
"sábado"]
#-----
# Pegando as segundas de feriados e
# organizando de forma matricial
#-----
DFerDom=t(matrix(rep(DFeriado,24),
length(DFeriado))[,weekdays
(Dias)=="domingo"])
#-----
#pegando a carga dos feriados
#-----
MCargaDFer=CargaDomDfer*DFerDom

```

```
# -----  
# Criando a constante  
# -----  
Const= rep(1,418)  
# -----  
# Nós da base de splines  
# -----  
Breaks=c(1,10,11,12,17,18,19,20,24)  
# -----  
# Ordem do Spline  
# -----  
norder=4  
# -----  
# Criando uma base de splines  
# -----  
cargabasis = create.bspline.basis(c(1, 24),  
norder+length(Breaks)-2,  
norder,breaks=Breaks)  
cargafdPar = fdPar(cargabasis)  
# -----  
# Domínio do spline  
# -----  
Argvals=c(1:24)  
# -----  
# Carga como combinação da base de splines  
# -----  
cargaSmooth=smooth.basis(Argvals,CargaDomDfer,  
cargafdPar)
```

```

cargaDferfd = cargaSmooth$fd
# -----
# Covariada como combinação linear da base de splines
# -----
LagcargaSmooth=smooth.basis(Argvals,
CovDomDfer,cargafdPar)
LagcargaDferfd= LagcargaSmooth$fd
# -----
#Dummy como combinação linear da
#base de splines
# -----
DFerSmooth=smooth.basis(Argvals,MCargaDFer
,cargafdPar)
DFerfd = DFerSmooth$fd
# -----
# criando uma base de fourier para a temperatura
# -----
Tempbasis = create.fourier.basis(c(1, 24),7)
TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)
# -----
# escrevendo a temperatura como combinação da base de fourier
# -----
TempSmooth=smooth.basis(Argvals,
TempDomDfer,cargafdPar)
TempDferfd = TempSmooth$fd
# -----
# Senoides e cossenoides

```

```

#-----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin1= sin( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Cos2= cos( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin2= sin( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
#-----
#Número de variáveis explicativas
#-----
NVE= 4
#-----
# Nós da base de parâmetros ordem do spline
#-----
Breaks=c(1,10,11,12,17,18,19,19,19,20,24)
#-----
# Ordem do Spline
#-----
norder=4
#-----
# Criando a base de parâmetros
#-----
betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)
#-----
#lista de parâmetros para serem estimados
#-----
betaList = vector("list",NVE)

```

```

names(betaList)= c("Const","LagCarga","Dummy","Temp")#,"Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betaafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= rep(1,ncol(CargaDomDfer))
CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFferfd
CovListNew[[4]]=TempDferfd
CovListNew[[5]]= Cos1
CovListNew[[6]]= Sin1
CovListNew[[7]]= Cos2
CovListNew[[8]]= Sin2
#-----
# Rodando o modelo
#-----

```

```

ModeloDomDFer= fRegress(cargaDferfd,CovListNew , betaList)
#-----
#Dias utilizados na modelagem - Segunda-Feira
#-----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")
#-----
# Criando uma Dummy de feriados
#-----
D1= ehFeriadoMBR(Dias)
D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
#-----
#Semanas completas nos dias observados
#-----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
#-----
#Pegando os dados de Temperatura apenas para semanas completas
#-----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
#-----
#Pegando os dados da carga apenas para semanas completas
#-----

```

```

CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
#-----
#Transformando os dados em relação a media da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
CargaSemCom))[-1]),168),,168)))
#-----
#Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
#-----
# Restringido os dados transformados para semanas completas a partir de 2003.
# Para resolver os problemas induzidos pelo racionamento de energia
#-----
Temperatura= matrix(c(TempComp),24)[
, is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24)[
, is.element(DiasSemComTransf,Dias)]
LagCarga= matrix(c(CargaComp),24)[
, is.element(DiasSemComTransf,Dias-7)]
#-----

```

```

#Pegando a carga da segunda-feira
#-----
CargaSegDfer= Carga[,weekdays(Dias)==
"segunda-feira"]
#-----
#Pegando a Temperatura da segunda
#-----
TempSegDfer= Temperatura[,weekdays(Dias)==
"segunda-feira"]
#-----
#A covariada da segunda é a média dos dias úteis da semana imediatamente anterior
#-----
CovSegDfer= apply(array( c(LagCarga[,
weekdays(Dias-7)!="sábado"&
weekdays(Dias-7)!="domingo"]),
c(24,5,419)),3,rowMeans)
#-----
# Pegando as segundas de feriados e organizando de forma matricial
#-----
DFerSeg=t(matrix(rep(DFeriado,24),
length(DFeriado))[,weekdays
(Dias)=="segunda-feira"]
#-----
#Pegando a carga dos feriados
#-----
MCargaDFer=CargaSegDfer*DFerSeg
#-----
#Criando a constante

```

```
# -----  
Const= rep(1,418)  
# -----  
# Nós da base de splines  
# -----  
Breaks=c(1,10,11,12,17,18,19,20,24)  
# -----  
# Ordem do Spline  
# -----  
norder=4  
# -----  
# Criando uma base de splines  
# -----  
cargabasis = create.bspline.basis(c(1, 24),  
norder+length(Breaks)-2,  
norder,breaks=Breaks)  
# -----  
# Transforando em parâmetro  
# -----  
cargafdPar = fdPar(cargabasis)  
# -----  
# Domínio do spline  
# -----  
Argvals=c(1:24)  
# -----  
# Carga como combinação da base de splines  
# -----  
cargaSmooth=smooth.basis(Argvals,CargaSegDfer[,-1],
```

```

cargafdPar)
cargaDferfd = cargaSmooth$fd
#-----
# Covariada como combinação linear da base de splines
#-----
LagcargaSmooth=smooth.basis(Argvals,
CovSegDfer[,-419],cargafdPar)
LagcargaDferfd= LagcargaSmooth$fd
#-----
# Dummy como combinação linear da base de splines
#-----
DFerSmooth=smooth.basis(Argvals,MCargaDFer
[-1],cargafdPar)
DFerfd = DFerSmooth$fd
#-----
# Criando uma base de fourier para a temperatura
#-----
Tempbasis = create.fourier.basis(c(1, 24),7)
TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)
#-----
# Escrevendo a temperatura como combinação da base de fourier
#-----
TempSmooth=smooth.basis(Argvals,
TempSegDfer[-1],cargafdPar)
TempDferfd = TempSmooth$fd
#-----
# Senoides e cossenoides

```

```

#-----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"] [-1]

Sin1= sin( (2*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"] [-1]

Cos2= cos( (4*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"] [-1]

Sin2= sin( (4*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"] [-1]

#-----
#Número de variáveis explicativas
#-----
NVE= 4
#-----
# Nós da base de parâmetros ordem do spline
#-----
Breaks=c(1,10,11,14,17,18,19,19,19,20,24)
#-----
# Ordem do Spline
#-----
norder=4
#-----
# Criando a base de parâmetros
#-----
betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)

```

```

#-----
#lista de parâmetros para serem estimados
#-----
betaList = vector("list",NVE)
names(betaList)= c("Const","LagCarga","Dummy","Temp","Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= rep(1,ncol(LagcargaDferfd$coef))
CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFferfd
CovListNew[[4]]=TempDferfd
CovListNew[[5]]= Cos1
CovListNew[[6]]= Sin1
CovListNew[[7]]= Cos2

```

```

CovListNew[[8]]= Sin2
# -----
# Rodando o modelo
# -----
ModeloSegDFer= fRegress(cargaDferfd,CovListNew , betaList)
# -----
# Dias utilizados na modelagem - terça-feira
# -----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")
# -----
# Criando uma Dummy de feriados
# -----
D1= ehFeriadoMBR(Dias)
D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
# -----
# Semanas completas nos dias observados
# -----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
# -----
# Pegando os dados de Temperatura apenas para semanas completas
# -----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,

```

```

DiasSemCom),3:26]))),168)
TempComp=TempSemCom[,-1]
#-----
# Pegando os dados da carga apenas para semanas completas
#-----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26]))),168)
#-----
# Transformando os dados em relação a média da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
(CargaSemCom))[-1]),168),,168)))
#-----
# Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
#-----
# Restringido os dados transformados para semanas completas a partir de 2003.
# Para resolver os problemas induzidos pelo racionamento de energia
#-----
Temperatura= matrix(c(TempComp),24)[
,is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24)[

```

```

, is.element(DiasSemComTransf, Dias)]
LagCarga= matrix(c(CargaComp), 24) [
, is.element(DiasSemComTransf, Dias-1)]
# -----
# Pegando a carga da terça-feira
# -----
CargaTerDfer= Carga[, weekdays(Dias)==
"terça-feira"]
# -----
# Pegando a Temperatura da terça-feira
# -----
TempTerDfer= Temperatura[, weekdays(Dias)==
"terça-feira"]
# -----
# Covariada da terça-feira
# -----
CovTerDfer= LagCarga[, weekdays(Dias-1)==
"segunda-feira"]
# -----
# Pegando as terças de feriados e organizando de forma matricial
# -----
DFerTer=t(matrix(rep(DFeriado, 24),
length(DFeriado)))[, weekdays
(Dias)=="terça-feira"]
# -----
# Pegando a carga dos feriados
# -----
MCargaDFer=CargaTerDfer*DFerTer

```

```
# -----  
# Nós da base de splines  
# -----  
Breaks=c(1,10,11,12,17,18,19,20,24)  
# -----  
# Ordem do Spline  
# -----  
norder=4  
# -----  
# Criando uma base de splines  
# -----  
cargabasis = create.bspline.basis(c(1, 24),  
norder+length(Breaks)-2,  
norder,breaks=Breaks)  
cargafdPar = fdPar(cargabasis)  
# -----  
# Domínio do spline  
# -----  
Argvals=c(1:24)  
# -----  
# Carga como combinação da base de splines  
# -----  
cargaSmooth=smooth.basis(Argvals,CargaTerDfer,  
cargafdPar)  
cargaDferfd = cargaSmooth$fd  
# -----  
# Covariada como combinação linear da base de splines  
# -----
```

```

LagcargaSmooth=smooth.basis(Argvals,
CovTerDfer,cargafdPar)
LagcargaDferfd= LagcargaSmooth$fd
#-----
# Dummy como combinação linear da base de splines
#-----
DFerSmooth=smooth.basis(Argvals,MCargaDFer
,cargafdPar)
DFerfd = DFerSmooth$fd
#-----
# Criando uma base de fourier para a temperatura
#-----
Tempbasis = create.fourier.basis(c(1, 24),7)
TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)
#-----
# Escrevendo a temperatura como combinação da base de fourier
#-----
TempSmooth=smooth.basis(Argvals,
TempTerDfer,cargafdPar)
TempDferfd = TempSmooth$fd
#-----
# Senoides e cossenoides
#-----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"]
Sin1= sin( (2*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"]
Cos2= cos( (4*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"]
Sin2= sin( (4*pi/52)* (1:length(Dias))/7 ) [weekdays(Dias)=="domingo"]

```

```

#-----
#Número de variáveis explicativas
#-----
NVE= 4
#-----
# Nós da base de parâmetros ordem do spline
#-----
Breaks=c(1,10,11,12,17,18,19,19,19,20,24)
#-----
# Ordem do Spline
#-----
norder=4
#-----
# Criando a base de parâmetros
#-----
betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)
#-----
#lista de parâmetros para serem estimados
#-----
betaList = vector("list",NVE)
names(betaList)= c("Const","LagCarga","Dummy","Temp","Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))

```

```

betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = betafdPar
betaList[[NVE]] = betafdPar
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= rep(1,ncol(CargaDomDfer))
CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFerfd
CovListNew[[4]]=TempDferfd
CovListNew[[5]]= Cos1
CovListNew[[6]]= Sin1
CovListNew[[7]]= Cos2
CovListNew[[8]]= Sin2
#-----
# Rodando o modelo
#-----
ModeloTerDFer= fRegress(cargaDferfd,CovListNew , betaList)
#-----
# Dias utilizados na modelagem - Quarta-feira
#-----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")

```

```

# -----
# Criando uma Dummy de feriados
# -----
D1= ehFeriadoMBR(Dias)
D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
# -----
# Semanas completas nos dias observados
# -----
DiasSemCom= seq(   from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
# -----
# Pegando os dados de Temperatura apenas para semanas completas
# -----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
# -----
# Pegando os dados de Carga apenas para semanas completas
# -----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
# -----
# Transformando os dados em relação à media da semana anterior e tirando log.
# -----

```

```

CargaComp = log(CargaSemCom[, -1] / t(matrix(
rep(rev(rev(colMeans
(CargaSemCom))[-1]), 168), , 168)))
# -----
# Dias que entram nos dados transformados
# -----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5, by="days")
# -----
# Restringido os dados transformados para semanas completas a partir de 2003.
# Para resolver os problemas induzidos pelo racionamento de energia.
# -----
Temperatura= matrix(c(TempComp), 24) [
, is.element(DiasSemComTransf, Dias)]
Carga=matrix(c(CargaComp), 24) [
, is.element(DiasSemComTransf, Dias)]
LagCarga= matrix(c(CargaComp), 24) [
, is.element(DiasSemComTransf, Dias-1)]
# -----
# Pegando a carga da quarta-feira
# -----
CargaQuaDfer= Carga[, weekdays(Dias)==
"quarta-feira"]
# -----
# Pegando a Temperatura da quarta-feira
# -----

```

```

TempQuaDfer= Temperatura[,weekdays(Dias)==
"quarta-feira"]
#-----
# Covariada da quarta-feira
#-----
CovQuaDfer= LagCarga[,weekdays(Dias-1)==
"terça-feira"]
#-----
# Pegando as quarta-feira de feriados e organizando de forma matricial
#-----
DFerQua=t(matrix(rep(DFeriado,24),
length(DFeriado))[,weekdays
(Dias)== "quarta-feira"])
#-----
# Pegando a carga dos feriados
#-----
MCargaDFer=CargaQuaDfer*DFerQua
#-----
# Criando a constante
#-----
Const= rep(1,418)
#-----
# Nós da base de splines ordem do Spline criando uma base de splines
#-----
Breaks=c(1,10,11,12,17,18,19,20,24)
norder=4
cargabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,

```

```

norder,breaks=Breaks)
cargafdPar = fdPar(cargabasis)
#-----
#Domínio do spline
#-----
Argvals=c(1:24)
#-----
# Carga como combinação da base de splines
#-----
cargaSmooth=smooth.basis(Argvals,CargaQuaDfer,
cargafdPar)
cargaDferfd = cargaSmooth$fd
#-----
# Covariada como combinação linear da base de splines
#-----
LagcargaSmooth=smooth.basis(Argvals,
CovQuaDfer,cargafdPar)
LagcargaDferfd= LagcargaSmooth$fd
#-----
# Dummy como combinação linear da base de splines
#-----
DFerSmooth=smooth.basis(Argvals,MCargaDFer
,cargafdPar)
DFerfd = DFerSmooth$fd
#-----
# Criando uma base de fourier para a temperatura
#-----
Tempbasis = create.fourier.basis(c(1, 24),7)

```

```

TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)
#-----
# Escrevendo a temperatura como combinação da base de fourier
#-----
TempSmooth=smooth.basis(Argvals,
TempQuaDfer,cargafdPar)
TempDferfd = TempSmooth$fd
#-----
# Senoides e cossenoides
#-----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin1= sin( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Cos2= cos( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin2= sin( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
#-----
#Número de variáveis explicativas
#-----
NVE= 4
#-----
# Nós da base de parâmetros ordem do spline
#-----
Breaks=c(1,10,11,12,17,18,19,19,19,20,24)
#-----
# Ordem do Spline
#-----
norder=4
#-----

```

```

# Criando a base de parâmetros
#-----
betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)
#-----
#lista de parâmetros para serem estimados
#-----
betaList = vector("list",NVE)
names(betaList)= c("Const","LagCarga","Dummy","Temp","Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]] = rep(1,ncol(CargaDomDfer))

```

```

CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFerfd
CovListNew[[4]]=TempDferfd
CovListNew[[5]]= Cos1
CovListNew[[6]]= Sin1
CovListNew[[7]]= Cos2
CovListNew[[8]]= Sin2

#-----
# Rodando o modelo
#-----
ModeloQuaDFer= fRegress(cargaDferfd,CovListNew , betaList)
#-----
# Dias utilizados na modelagem - Quinta-feira
#-----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")
#-----
# Criando uma Dummy de feriados
#-----
D1= ehFeriadoMBR(Dias)
D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
#-----
# Semanas completas nos dias observados
#-----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,

```

```

by="days")
#-----
# Pegando os dados de Temperatura apenas para semanas completas
#-----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
#-----
# Pegando os dados de carga apenas para semanas completas
#-----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
#-----
# Transformando os dados em relação à média da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
(CargaSemCom))[-1]),168),,168)))
#-----
# Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(    from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
#-----

```

```

# Restringido os dados transformados para semanas completas a partir de 2003.
# Para livrar dos problemas induzidos pelo racionamento de energia
# -----
Temperatura= matrix(c(TempComp),24) [
, is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias)]
LagCarga= matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias-1)]
# -----
# Pegando a carga da quinta-feira
# -----
CargaQuiDfer= Carga[,weekdays(Dias)==
"quinta-feira"]
# -----
# Pegando a Temperatura da quinta-feira
# -----
TempQuiDfer= Temperatura[,weekdays(Dias)==
"quinta-feira"]
# -----
# Covariada da quinta-feira
# -----
CovQuiDfer= LagCarga[,weekdays(Dias-1)==
"quarta-feira"]
# -----
# Pegando as quintas-feiras de feriados e organizando de forma matricial
# -----
DFerQui=t(matrix(rep(DFeriado,24),

```

```

length(DFeriado)))[,weekdays
(Dias)=="quinta-feira"]
#-----
# Pegando a carga dos feriados
#-----
MCargaDFer=CargaQuiDfer*DFerQui
#-----
# Nós da base de splines
#-----
Breaks=c(1,10,11,12,17,18,19,19,19,20,24)
#-----
# Ordem do Spline
#-----
norder=4
#-----
# Criando uma base de splines
#-----
cargabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
cargafdPar = fdPar(cargabasis)
#-----
# Domínio do spline
#-----
Argvals=c(1:24)
#-----
# Carga como combinação da base de splines
#-----

```

```

cargaSmooth=smooth.basis(Argvals,CargaQuiDfer,
cargafdPar)
cargaDferfd = cargaSmooth$fd
#-----
# Covariada como combinação linear da base de splines
#-----
LagcargaSmooth=smooth.basis(Argvals,
CovQuiDfer,cargafdPar)
LagcargaDferfd= LagcargaSmooth$fd
#-----
# Dummy como combinação linear da base de splines
#-----
DFerSmooth=smooth.basis(Argvals,MCargaDFer
,cargafdPar)
DFerfd = DFerSmooth$fd
#-----
# Criando uma base de fourier para a temperatura
#-----
Tempbasis = create.fourier.basis(c(1, 24),3)
TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)
#-----
# Escrevendo a temperatura como combinação da base de fourier
#-----
TempSmooth=smooth.basis(Argvals,
TempQuiDfer,cargafdPar)
TempDferfd = TempSmooth$fd
#-----

```

```

# Senoides e cossenoides
# -----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin1= sin( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
# -----
#Número de variáveis explicativas
# -----
NVE= 4
# -----
# Nós da base de parâmetros ordem do spline
# -----
Breaks=c(1,10,11,12,17,18,19,19,19,20,24)
# -----
# Ordem do Spline
# -----
norder=4
# -----
# Criando a base de parâmetros
# -----
betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)
# -----
#lista de parâmetros para serem estimados
# -----
betaList = vector("list",NVE)

```

```

names(betaList)= c("Const","LagCarga","Dummy","Temp","Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = betafdPar
betaList[[NVE]] = betafdPar
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= rep(1,ncol(CargaDomDfer))
CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFerfd
CovListNew[[4]]=TempDferfd
#CovListNew[[5]]= Cos1
#CovListNew[[6]]= Sin1
#CovListNew[[7]]= Cos2
#CovListNew[[8]]= Sin2
#-----
# Rodando o modelo
#-----
ModeloQuiDFer= fRegress(cargaDferfd,CovListNew , betaList)
#-----

```

```

# Dias utilizados na modelagem - Sexta-feira
# -----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")
# -----
# Criando uma Dummy de feriados
# -----
D1= ehFeriadoMBR(Dias)
D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
# -----
# Semanas completas nos dias observados
# -----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
# -----
# Pegando os dados de Temperatura apenas para semanas completas
# -----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
# -----
# Pegando os dados de carga para semanas completas
# -----
CargaSemCom= matrix(c(t(dadoscarga[is.element(

```

```

DiasDadosCarga,
DiasSemCom),3:26]))),168)
#-----
# Transformando os dados em relação a media da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
(CargaSemCom))[-1]),168),,168)))
#-----
# Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
#-----
# Restringido os dados transformados para semanas completas a partir de 2003.
# Para resolver os problemas induzidos pelo racionamento de energia
#-----
Temperatura= matrix(c(TempComp),24) [
,is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24) [
,is.element(DiasSemComTransf,Dias)]
LagCarga= matrix(c(CargaComp),24) [
,is.element(DiasSemComTransf,Dias-1)]
#-----
# Pegando a carga da sexta-feira
#-----

```

```

CargaSexDfer= Carga[,weekdays(Dias)==
"sexta-feira"]
#-----
# Pegando a Temperatura da sexta-feira
#-----
TempSexDfer= Temperatura[,weekdays(Dias)==
"sexta-feira"]
#-----
# Covariada da sexta-feira
#-----
CovSexDfer= LagCarga[,weekdays(Dias-1)==
"quinta-feira"]
#-----
# Pegando as segundas de feriados e organizando de forma matricial
#-----
DFerSex=t(matrix(rep(DFeriado,24),
length(DFeriado))[,weekdays
(Dias)=="sexta-feira"])
#-----
# Pegando a carga dos feriados
#-----
MCargaDFer=CargaSexDfer*DFerSex
#-----
# Nós da base de splines
#-----
Breaks=c(1,10,11,12,17,18,19,20,24)
#-----
# Ordem do Spline

```

```
# -----  
norder=4  
# -----  
# Criando uma base de splines  
# -----  
cargabasis = create.bspline.basis(c(1, 24),  
norder+length(Breaks)-2,  
norder,breaks=Breaks)  
cargafdPar = fdPar(cargabasis)  
# -----  
# Domínio do spline  
# -----  
Argvals=c(1:24)  
# -----  
# Carga como combinação da base de splines  
# -----  
cargaSmooth=smooth.basis(Argvals,CargaSexDfer,  
cargafdPar)  
cargaDferfd = cargaSmooth$fd  
# -----  
# Covariada como combinação linear da base de splines  
# -----  
LagcargaSmooth=smooth.basis(Argvals,  
CovSexDfer,cargafdPar)  
LagcargaDferfd= LagcargaSmooth$fd  
# -----  
# Dummy como combinação linear da base de splines  
# -----
```

```

DFerSmooth=smooth.basis(Argvals,MCargaDFer
,cargafdPar)
DFerfd = DFerSmooth$fd
#-----
# Criando uma base de fourier para a temperatura
#-----
Tempbasis = create.fourier.basis(c(1, 24),7)
TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)
#-----
# Escrevendo a temperatura como combinação da base de fourier
#-----
TempSmooth=smooth.basis(Argvals,
TempSexDfer,cargafdPar)
TempDferfd = TempSmooth$fd
#-----
# Senoides e cossenoides
#-----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin1= sin( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Cos2= cos( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin2= sin( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
#-----
#Número de variáveis explicativas
#-----
NVE= 4
#-----
# Nós da base de parâmetros ordem do spline

```

```

# -----
Breaks=c(1,10,11,12,17,18,19,19,19,20,24)
# -----
# Ordem do Spline
# -----
norder=4
# -----
# Criando a base de parâmetros
# -----
betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)
# -----
#lista de parâmetros para serem estimados
# -----
betaList = vector("list",NVE)
names(betaList)= c("Const","LagCarga","Dummy","Temp")#,"Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = fdPar(
create.fourier.basis(c(1, 24),5))

```

```

betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= rep(1,ncol(CargaDomDfer))
CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFerfd
CovListNew[[4]]=TempDferfd
CovListNew[[5]]= Cos1
CovListNew[[6]]= Sin1
CovListNew[[7]]= Cos2
CovListNew[[8]]= Sin2
#-----
# Rodando o modelo
#-----
ModeloSexDFer= fRegress(cargaDferfd,CovListNew , betaList)
#-----
# Dias utilizados na modelagem - Sábado
#-----
Dias = seq( from= as.Date("05/01/2003", "%d/%m/%Y"),
to= as.Date("15/01/2011", "%d/%m/%Y"),
by="days")
#-----
# Criando uma Dummy de feriados
#-----
D1= ehFeriadoMBR(Dias)

```

```

D2= ehFeriadoBR(Dias)
DFeriado=D1+D2
#-----
# Semanas completas nos dias observados
#-----
DiasSemCom= seq(   from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
#-----
# Pegando os dados de Temperatura apenas para semanas completas
#-----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
#-----
# Pegando os dados da Carga apenas para semanas completas
#-----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
#-----
# Transformando os dados em relação a media da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
(CargaSemCom))[-1]),168),,168)))
#-----

```

```

# Dias que entram nos dados transformados
# -----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
# -----
# Restringido os dados transformados para semanas completas a partir de 2003.
# Para resolver os problemas induzidos pelo racionamento de energia
# -----
Temperatura= matrix(c(TempComp),24) [
, is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias)]
LagCarga= matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias-1)]
# -----
# Pegando a carga do sábado
# -----
CargaSabDfer= Carga[,weekdays(Dias)==
"sábado"]
# -----
# Pegando a Temperatura do sábado
# -----
TempSabDfer= Temperatura[,weekdays(Dias)==
"sábado"]
# -----
# Covariada do sábado

```

```
#-----  
CovSabDfer= LagCarga[,weekdays(Dias-1)==  
"sexta-feira"]  
#-----  
# Pegando os sábados de feriados e organizando de forma matricial  
#-----  
DFerSab=t(matrix(rep(DFeriado,24),  
length(DFeriado))[,weekdays  
(Dias)=="sábado"]  
#-----  
# Pegando a carga dos feriados  
#-----  
MCargaDFer=CargaSabDfer*DFerSab  
#-----  
# Nós da base de splines  
#-----  
Breaks=c(1,10,11,12,17,18,19,20,24)  
#-----  
# Ordem do Spline  
#-----  
norder=4  
#-----  
# Criando uma base de splines  
#-----  
cargabasis = create.bspline.basis(c(1, 24),  
norder+length(Breaks)-2,  
norder,breaks=Breaks)  
cargafdPar = fdPar(cargabasis)
```

```

# -----
# Domínio do spline
# -----
Argvals=c(1:24)
# -----
# Carga como combinação da base de splines
# -----
cargaSmooth=smooth.basis(Argvals,CargaSabDfer,
cargafdPar)
cargaDferfd = cargaSmooth$fd
# -----
# Covariada como combinação linear da base de splines
# -----
LagcargaSmooth=smooth.basis(Argvals,
CovSabDfer,cargafdPar)
LagcargaDferfd= LagcargaSmooth$fd
# -----
# Dummy como combinação linear da base de splines
# -----
DFerSmooth=smooth.basis(Argvals,MCargaDFer
,cargafdPar)
DFerfd = DFerSmooth$fd
# -----
# Criando uma base de fourier para a temperatura
# -----
Tempbasis = create.fourier.basis(c(1, 24),7)
TempfdPar = fdPar(Tempbasis)
Argvals=c(1:24)

```

```

# -----
# Escrevendo a temperatura como combinação da base de fourier
# -----
TempSmooth=smooth.basis(Argvals,
TempSabDfer,cargafdPar)
TempDferfd = TempSmooth$fd
# -----
# Senoides e cossenoides
# -----
Cos1= cos( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin1= sin( (2*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Cos2= cos( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
Sin2= sin( (4*pi/52)* (1:length(Dias))/7 )[weekdays(Dias)=="domingo"]
# -----
#Número de variáveis explicativas
# -----
NVE= 4
# -----
# Nós da base de parâmetros ordem do spline
# -----
Breaks=c(1,10,11,12,17,18,19,19,20,24)
# -----
# Ordem do Spline
# -----
norder=4
# -----
# Criando a base de parâmetros
# -----

```

```

betabasis = create.bspline.basis(c(1, 24),
norder+length(Breaks)-2,
norder,breaks=Breaks)
betafdPar = fdPar(betabasis)
#-----
#lista de parâmetros para serem estimados
#-----
betaList = vector("list",NVE)
names(betaList)= c("Const","LagCarga","Dummy","Temp","Cos","Sin","Cos2","Sin2")

for (j in 1:(NVE)) betaList[[j]] = betafdPar
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-3]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-2]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE-1]] = fdPar(
create.fourier.basis(c(1, 24),5))
betaList[[NVE]] = fdPar(
create.fourier.basis(c(1, 24),5))
#-----
#Lista de covariadas
#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= rep(1,ncol(CargaDomDfer))
CovListNew[[2]]= LagcargaDferfd
CovListNew[[3]]=DFferfd

```

```

CovListNew[[4]]=TempDferfd
CovListNew[[5]]= Cos1
CovListNew[[6]]= Sin1
CovListNew[[7]]= Cos2
CovListNew[[8]]= Sin2
#-----
# Rodando o modelo
#-----
ModeloSabDFer= fRegress(cargaDferfd,CovListNew , betaList)

Programação do modelo semanal
#-----
# Dias utilizados na modelagem
#-----
dias = seq( from= as.Date("05/01/2003",
"%d/%m/%Y"),to= as.Date("15/01/2011",
"%d/%m/%Y"),by="days")
#-----
# Semanas completas nos dias observados
#-----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,
by="days")
#-----
# Pegando os dados de Temperatura apenas para semanas completas
#-----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,

```

```

DiasSemCom),3:26]))),168)
#-----
# Pegando os dados de Temperatura apenas para semanas completas
#-----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26]))),168)
TempComp=TempSemCom[,-1]
#-----
# Transformando os dados em relação a media da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans
(CargaSemCom))[-1]),168),,168)))
#-----
# Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(   from= as.Date(
"06/01/2001", "%m/%d/%Y")+2+7,
to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
#-----
# Restringido os dados transformados para semanas completas a partir de 2003.
#-----
Temperatura= matrix(c(TempComp),24) [
, is.element(DiasSemComTransf,Dias)]
Carga=matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias)]

```

```

LagCarga= matrix(c(CargaComp),24) [
, is.element(DiasSemComTransf,Dias-1)]
#-----
# Dias observados e nos quais não vale o horário de verão
#-----
HorarioInverno= rep(c(1,0),9) +
as.Date( c( "2003-02-15", "2003-09-28",
"2004-02-14", "2004-11-07",
"2005-02-19", "2005-10-16",
"2006-02-18", "2006-11-05",
"2007-02-24", "2007-10-14",
"2008-02-16", "2008-10-19",
"2009-02-14", "2009-10-18",
"2010-02-20", "2010-10-17",
"2011-02-19", "2011-10-16") )
#-----
# Função que verifica amostra em que vale horário de verão
#-----
EhVerao= apply( outer( seq( from= as.Date("2003-01-05"),
to= as.Date("2011-01-15"),
by="days"),
HorarioInverno,
function(x,y) x<y ),
1, sum )
EhVerao= (EhVerao%%2 == 0)
EhVerao= t(matrix( rep(EhVerao, 24), , 24) )
#-----
# Bases

```

```

#-----
# Nós da base de splines para um dia
#-----
PosIni= c( 7, 11, 13, 13, 13, 18, 18, 18,
19, 19, 19, 20, 20, 20, 23)
#-----
# Expandindo os nós para uma semana
#-----
PosFin= PosIni
for(k in 1:6) PosFin= c(PosFin, PosIni+ 24*k)
Whoras=(1:168)
#-----
# Criando uma base de splines que será utilizada para suavizar a carga
#-----
Wloadbasis = create.bspline.basis(rangeval=c(0,24*7),
norder=4,breaks=Whoras[PosFin] )
loadBasis = create.bspline.basis(
rangeval=c(0,24),norder=6,
breaks=horas[c( 1, 4, 5, 6, 10, 11, 12, 17,
rep(19,3), 21, 24)] )
#-----
# Melhores parâmetros de suavização
# Os dados serão classificados em 4 classes: verão, FMA, MJJ e ASO
#-----
# criando sequência de 01/05
#-----
PrMaios= seq(from=as.Date("2003-05-01"),
to=as.Date("2010-05-01"),by="year") #

```

```

#-----
# Se o primeiro de maio ocorre no meio da semana alguns dias da semana serão FMA
e
# outros MJJ. Para classificar a semana de forma homogênea coletamos os dias da
semana
# na vizinhança do 1º de maio
#-----
PrMaiosIN= seq(from= as.Date(PrMaios[1])-
(as.POSIXlt(as.Date(PrMaios[1]))$yday),
to=PrMaios[1],by="days")
PrMaiosFN=PrMaiosIN
N= length(PrMaiosIN)
#-----
# Dias que ocorrem na mesma semana em que ocorre o 1º o maio
#-----
for( i in 2:N){
X =seq(from= as.Date(PrMaios[i])-
(as.POSIXlt(as.Date(PrMaios[i]))$yday),
to=PrMaios[i],by="days")
PrMaiosFN=c(PrMaiosFN,X) }
#-----
# Criando sequência de 01/08
#-----
PrAgo= seq(from=as.Date("2003-08-01"),
to=as.Date("2010-08-01"),by="year")
PrAgoFN= seq(from= as.Date(PrAgo[1])-
(as.POSIXlt(as.Date(PrAgo[1]))$yday),
to=PrAgo[1],by="days")

```

```

N= length(PrAgo)
# -----
# Dias que ocorrem na mesma semana em que ocorre o 1º de agosto
# -----
for( i in 2:N){
X =seq(from= as.Date(PrAgo[i])-
(as.POSIXlt(as.Date(PrAgo[i]))$yday),
to=PrAgo[i],by="days")
PrAgoFN=c(PrAgoFN,X) }
# -----
# Dias que estão na vizinhança do 1º de maio?
# -----
TestePrMaio= is.element(dias,PrMaiosFN)
TestePrAgo= is.element(dias,PrAgoFN)
# -----
# Hverao=True vale Horário de verão
# -----
Hverao=as.vector(t(EhVerao[1,]))
# -----
# Inicializando um vetor de classificações
# -----
vclasses=rep("verao",length(dias))
# -----
# Classificando os dias que pertencem à fevereiro e não vale horário de verão
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==1)]="FMA"
# -----
# Classificando os dias que pertencem à março

```

```

# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==2)]= "FMA"
# -----
# Classificando os dias que pertencem à abril
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==3)]= "FMA"
# -----
# Os dias que pertencem à semana em que o 1º de maio está contido serão
# classificados como MJJ independente deles pertencerem a abril
# -----
vclasses[(!Hverao)&(TestePrMaio)]= "MJJ"
# -----
# classificando os dias que pertencem à maio
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==4)]= "MJJ"
# -----
# Classificando os dias que pertencem à junho
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==5)]= "MJJ"
# -----
# Classificando os dias que percentencem à julho
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==6)]= "MJJ"
# -----
# Os dias que pertencem à semana em que o 1º de agosto está contido serão
# classificados como ASO independente deles pertencerem a julho
# -----
vclasses[(!Hverao)&(TestePrAgo)]= "ASO"

```

```

# -----
# Classificando os dias que percentencem à agosto
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==7)]="ASO"
# -----
# Classificando os dias que percentencem à setembro
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==8)]="ASO"
# -----
# Classificando os dias que percentencem à outubro e não
# vale o horário de verão
# -----
vclasses[(!Hverao)&(as.POSIXlt(dias)$mon==9)]="ASO"
# -----
# Cada linha de MatrixLoad é uma semana
# -----
MatrixLoad=t(matrix(c(Carga),168))
# -----
# As classificações devem valer para a semana
# -----
Mvclasses= matrix(vclasses,,7,byrow=T)
# -----
# Matriz que contém apenas semanas em que vale horário de verão
# -----
MatrixLoadverao= MatrixLoad[Mvclasses[,1]=="verao",]
# -----
# Sequência de possíveis parâmetros de suavização
# -----

```

```

Sverao=seq(0.01,0.3,by=0.01)
# -----
# Vetor que armazena o gcv
# -----
gcvLocverao=rep(0,length(Sverao))
# -----
# Suavizando a carga e obtendo os gcv's, o parâmetro de
# suavização é aquele que faz o gcv ser mínimo
# -----
for(i in 1:length(Sverao)){
Wloadfd= smooth.basisPar( Whoras,t(MatrixLoadverao),
Wloadbasis, lambda=Sverao[i])gcvLocverao[i]= sum(Wloadfd$gcv)}
# ----- - verao - 0,05
# -----
# FMA= fevereiro, Março, abril
# matriz que contém apenas semanas de FMA
# -----
MatrixLoadFMA=MatrixLoad[Mvclasses[,1]=="FMA",]
# -----
# Sequência de possíveis parâmetros de suavização
# -----
SFMA=seq(0.11,0.20,by=0.01)
# -----
#vetor que armazena o gcv
# -----
gcvLocFMA=rep(0,length(SFMA))
for(i in 1:length(SFMA)){
# -----

```

```

# Suavizando a carga e obtendo os gcv's, o parâmetro de
# suavização é aquele que faz o gcv ser mínimo
# -----
Wloadfd= smooth.basisPar( Whoras, t(MatrixLoadFMA),
Wloadbasis, lambda=SFMA[i])
gcvLocFMA[i]= sum(Wloadfd$gcv)}
# ----- - FMA - 0.15
# -----
# MJJ= maio, junho, julho
# matriz que contém apenas semanas de MJJ
# -----
MatrixLoadMJJ=MatrixLoad[Mvclasses[,1]=="MJJ",]
# -----
# Sequência de possíveis parâmetros de suavização
# -----
SMJJ=seq(0.11,0.20,by=0.01)
# -----
# Vetor que armazena o gcv
# -----
gcvLocMJJ=rep(0,length(SMJJ))
# -----
# Suavizando a carga e obtendo os gcv's, o parâmetro de suavização
# é aquele que faz o gcv ser mínimo
# -----
for(i in 1:length(SMJJ)){
Wloadfd= smooth.basisPar( Whoras, t(MatrixLoadMJJ),
Wloadbasis, lambda=SMJJ[i])
gcvLocMJJ[i]= sum(Wloadfd$gcv)}

```

```

# ----- - MJJ - 0.13
# -----
# ASO= Agosto, setembro, outubro
# matriz que contém apenas semanas de ASO
# -----
MatrixLoadASO=MatrixLoad[Mvclasses[,1]=="ASO",]
# -----
# Sequência de possíveis parâmetros de suavização
# -----
SASO=seq(0.11,0.20,by=0.01)
# -----
# Vetor que armazena o gcv
# -----
gcvLocASO=rep(0,length(SASO))
for(i in 1:length(SASO)){
# -----
# Suavizando a carga e obtendo os gcv's, o parâmetro de
# suavização é aquele que faz o gcv ser mínimo
# -----
Wloadfd= smooth.basisPar( Whoras, t(MatrixLoadASO),
Wloadbasis, lambda=SASO[i])
gcvLocASO[i]= sum(Wloadfd$gcv)}
# ----- - ASO - 0.13
# -----
# Fd da variavel dependente
# cada coluna da MatrixLoadT é uma coluna
# -----
MatrixLoadT=t(MatrixLoad)

```

```

#jogando fora a primeira semana
MatrixLoadT=MatrixLoadT[,-1]
CoefSV=rep(0,length(Mvclasses[,1]))
#coeficiente de suavização do verão
CoefSV[Mvclasses[,1]=="verao"]= 0.05
#coeficiente de suavização de FMA
CoefSV[Mvclasses[,1]=="FMA"]= 0.15
#coeficiente de suavização de MJJ
CoefSV[Mvclasses[,1]=="MJJ"]= 0.13
#coeficiente de suavização de ASO
CoefSV[Mvclasses[,1]=="ASO"]= 0.13
#-----
# Inicializando as suavizações e suavizando cada coluna
#-----
MloadCoef=0*MatrixLoadT
WloadfdV= smooth.basisPar( Whoras, y= MatrixLoadT[,1],
Wloadbasis, lambda=CoefSV[1])
MloadCoef=matrix(0,length(WloadfdV$fd$coef),ncol(MatrixLoadT))
MloadCoef[,1]=WloadfdV$fd$coef
#-----
# Suavizando cada coluna da matriz da variável dependente utilizando os
# os parâmetros de suavizaçãode acordo com a classificação da coluna
#-----
for( i in 2:ncol(MatrixLoadT)) {
WloadfdV= smooth.basisPar( Whoras,
y= MatrixLoadT[,i],
Wloadbasis, lambda=CoefSV[i])
MloadCoef[,i]= WloadfdV$fd$coef }

```

```

# -----
# Transformando a carga suavizada em um funcioanal
# -----
Loadfd= fd(MloadCoef,Wloadbasis)
# -----
# Construção das variáveis defasadas
# -----
# Inicializando uma matriz para construir a variável
# explicativa defasada da segunda-feira
# -----
TRANSEG = MatrixLoadT
# -----
# Obtendo a carga defasada
# -----
CovPrDom=rev(rev(MatrixLoadT[,1])[1:24])
LoadVec= c(TRANSEG)
LagLoadIN= matrix(c(CovPrDom,rev(rev(LoadVec)[-1:24]))),24*7)
# -----
# Inicilizando matriz para popular com a média da
# carga dos dias úteis anterior a cada segunda feira
# -----
Covseg= 0*MatrixLoadT
# -----
# Média da carga dos dias úteis úteis da semana anterior é a
# covariada da segunda-feira
# -----
DiasUteis= MatrixLoadT[-c(1:24,145:168),]
for( i in 1:ncol(MatrixLoadT)) {

```

```

trab= matrix( DiasUteis[,i], 24)
Covseg[25:48,i]= apply(trab, 1,
function(x) mean(x, trim=.2))}
for( i in 1:ncol(LagLoadIN)) {
LagLoadIN[25:48,i]= Covseg[25:48,i] }
#inicializando a carga defasada relativa aos dias de verão
LagLoadV= LagLoadIN
#inicializando a carga defasada relativa aos dias de FMA
LagLoadFMA=LagLoadIN
#inicializando a carga defasada relativa aos dias de MJJ
LagLoadMJJ=LagLoadIN
#inicializando a carga defasada relativa aos dias de ASO
LagLoadASO=LagLoadIN
MvclassesLag=Mvclasses[-1,]
#carga defasada relativa aos dias de verão
LagLoadV[,MvclassesLag[,1]!="verao"]=0
#carga defasada relativa aos dias de FMA
LagLoadFMA[,MvclassesLag[,1]!="FMA"]=0
#carga defasada relativa aos dias de MJJ
LagLoadMJJ[,MvclassesLag[,1]!="MJJ"]=0
#carga defasada relativa aos dias de ASO
LagLoadASO[,MvclassesLag[,1]!="ASO"]=0
#-----
#Construção das fds das variáveis defasadas
#-----
# Suavizando carga defasada relativa aos dias de verão
#-----
LagloadSmoothV= smooth.basisPar( Whoras, y= LagLoadV,

```

```

Wloadbasis, lambda=0.05)
#fd da carga defasada - verão
LagloadfdV= fd(LagloadSmoothV$fd$coef,Wloadbasis)
#-----
# Suavizando carga defasada relativa aos dias de FMA
#-----
LagloadSmoothFMA= smooth.basisPar( Whoras, y= LagLoadFMA,
Wloadbasis, lambda=0.15)
#fd da carga defasada - FMA
LagloadfdFMA= fd(LagloadSmoothFMA$fd$coef,Wloadbasis)
#-----
# Suavizando carga defasada relativa aos dias de MJJ
#-----
LagloadSmoothMJJ= smooth.basisPar( Whoras, y= LagLoadMJJ,
Wloadbasis, lambda=0.13)
#fd da carga defasada - MJJ
LagloadfdMJJ= fd(LagloadSmoothMJJ$fd$coef,Wloadbasis)
#-----
# Suavizando carga defasada relativa aos dias de ASO
#-----
LagloadSmoothASO= smooth.basisPar( Whoras, y= LagLoadASO,
Wloadbasis, lambda=0.13)
#fd da carga defasada - ASO
LagloadfdASO= fd(LagloadSmoothASO$fd$coef,Wloadbasis)
#-----
# Organizando a temperatura por semana
#-----
Temperatura= matrix(c(Temperatura),168,)[-1]

```

```

# -----
# Criando uma base para a temperatura transformando a temperatura em um fd
# -----
Tempbasis= create.fourier.basis(c(0,168), 15, 168)
Tempfd= smooth.basis(Whoras, Temperatura, Tempbasis)$fd
#criando uma constante
Const=rep(1,418)
#dias que são feriados fixos
D1= ehFeriadoBR(dias)
#dias que são feriados móveis
D2= ehFeriadoMBR(dias)
#classificando os dias como feriados ou não
DFer=D1+D2
#transformando a classificação em uma matriz de classificações
Fer= matrix(t(matrix(rep(DFer,24),,24)),168)
# capturando a carga de dias que são feriados
Feriados= Fer[,-1]*MatrixLoadT
aux= c(t(matrix(rep(DFer,24),,24)))
# capturando os dias posteriores aos feriados
FerPosFer=matrix(c(rep(0,24),rev(rev(aux)[-(1:24)]))),168)
#dias que são feriados e os pós feriados
FerePosFer= FerPosFer+Fer
# capturando a carga dos feriados e dos pós feriados
FeriadosePosFeriados=FerePosFer[,-1]*MatrixLoadT
#suavizando a carga dos feriados e pós feriados
DSmooth= smooth.basisPar( Whoras, y= FeriadosePosFeriados,
Wloadbasis, lambda=0.13)
#transformando a carga dos feriados e pós feriados em um funcional

```

```

Dfd= fd(DSmooth$fd$coef,Wloadbasis)
#número de variáveis explicativas
NVE=7
#-----
# Nós da base na qual os parâmetros serão expressos
#-----
BreaksPar= c(24,24,24,48,48,48,72,72,72,
96,96,96,120,120,120,144,144,144)
#-----
# Base utilizada para exprimir os parâmetros
#-----
betabasis = create.bspline.basis(
rangeval=c(0,24*7),
norder=4,
breaks=BreaksPar )
betafdPar = fdPar(betabasis)
betaList = vector("list",NVE)
names(betaList)=c("LagCargaV", "LagCargaFMA",
"LagCargaMJJ", "LagCargaASO", "Const", "Fer", "Temp")
for (j in 1:(NVE-1)) betaList[[j]] = betafdPar
#-----
# Base dos parâmetros da temperatura
#-----
betaTempbasis= create.fourier.basis(c(0,168), 15, 168)
betaTempfdPar = fdPar(betaTempbasis)
betaList[[NVE]]=betaTempfdPar
#-----
#Lista de variáveis explicativas

```

```

#-----
CovListNew=vector("list",NVE)
CovListNew[[1]]= LagloadfdV
CovListNew[[2]]= LagloadfdFMA
CovListNew[[3]]= LagloadfdMJJ
CovListNew[[4]]= LagloadfdASO
CovListNew[[5]]= Const
CovListNew[[6]]= Dfd
CovListNew[[7]]= Tempfd
#-----
#Rodando o modelo funcional
#-----
ModeloSem= fRegress(Loadfd,CovListNew , betaList)

```

Programação do modelo anual

```

#-----
# Dias utilizados na modelagem
#-----
DiasAnual= seq( as.Date("2003-01-05"),
as.Date("2010-12-25"),by="days")
#-----
#Semanas completas nos dias observados
#-----
DiasSemCom= seq( from= as.Date("06/01/2001", "%m/%d/%Y")+2,
to= as.Date("01/20/2011", "%m/%d/%Y")-5,by="days")

```

```

#-----
#Pegando os dados de Temperatura apenas para semanas completas
#-----
TempSemCom= matrix(c(t(dadostemp[is.element(
DiasDadosCarga,DiasSemCom),3:26])),168)
#-----
# Pegando os dados de Temperatura apenas para semanas completas
#-----
CargaSemCom= matrix(c(t(dadoscarga[is.element(
DiasDadosCarga,
DiasSemCom),3:26])),168)
TempComp=TempSemCom[,-1]
#-----
# Transformando os dados em relação a media da semana anterior e tirando log.
#-----
CargaComp = log(CargaSemCom[,-1] / t(matrix(
rep(rev(rev(colMeans(CargaSemCom))[-1]),168),,168)))
#-----
# Dias que entram nos dados transformados
#-----
DiasSemComTransf= seq(from= as.Date("06/01/2001",
"%m/%d/%Y")+2+7,to= as.Date("01/20/2011",
"%m/%d/%Y")-5,by="days")
Temperatura= matrix(c(TempComp),24)[,is.element(DiasSemComTransf,
DiasAnual)]
Carga=matrix(c(CargaComp),24)[
,is.element(DiasSemComTransf,DiasAnual)]
LagCarga= matrix(c(CargaComp),24)[

```

```

, is.element(DiasSemComTransf, DiasAnual-1)]
# -----
# Dados organizados por semana
# -----
CargaSem=matrix(c(Carga), 168)
TempSem=matrix(c(Temperatura), 168)
LagCargaSem=matrix(c(LagCarga), 168)
# -----
# Selecionando as semanas ímpares
# -----
CargaSemImpar= CargaSem[, seq(1, 416, by=2)]
TempSemImpar= TempSem[, seq(1, 416, by=2)]
LagCargaSemImpar= LagCargaSem[, seq(1, 416, by=2)]
# -----
# Carga organizada de maneira anual
# -----
LoadAnualImpar= matrix(c(CargaSemImpar), , 8)
TempAnualImpar= matrix(c(TempSemImpar), , 8)
LagCargaAnualImpar= matrix(c(LagCargaSemImpar), , 8)
# -----
# Carga das semanas ímpares organizada de maneira anual
# -----
# -----
# Criando base de splines para a carga anual
# -----
PosIni= c( 7, 11, 13, 13, 13, 18, 19, 19, 19, 20, 23)
PosIniSem=PosIni
for(i in 1:6) PosIniSem=c(PosIniSem, PosIni+24*i )

```

```

Anualhoras= (1:(52*168))-0.5
Dhoras = (1:24)-0.5
PosFinSemAnual= PosIniSem
for(k in 1:51) PosFinSemAnual= c(PosFinSemAnual, PosIniSem + 168*k)
Anualloadbasis = create.bspline.basis(
rangeval=c(0,24*7*52),
norder=4,
breaks=c(PosFinSemAnual-0.05) )
# -----
# Marcando os nós para uma base de splines
# para as semans ímpares do ano
# -----
CargaDay=matrix(c(Carga),24)
ANO1= seq( as.Date("2003-01-05"),
as.Date("2004-01-03"),
by="days")
CargaDayANO1= CargaDay[,is.element(DiasAnual,ANO1)]
CargaDayANO1Impar= matrix(matrix(CargaDayANO1,168)[,seq(1,52,by=2)],24)
Pico= rep(NA,ncol(CargaDayANO1Impar))
for( i in 1:ncol(CargaDayANO1Impar)){ maximo= max(CargaDayANO1Impar[,i])
matriz= cbind(1:24,CargaDayANO1Impar[,i]==maximo)
Pico[i]=matriz[CargaDayANO1Impar[,i]==maximo,1] }
# -----
#O pico varia ao longo do ano e por isso criamos 5 conjuntos de nós
# -----
breaks1= c( 7, 11, 13,18,19,20, 23)
breaks2= c( 7, 11, 13,18,19,19,19, 23)
breaks3= c( 7, 11, 13,18,19,20,20,20, 23)

```

```

breaks4= c( 7, 11, 13,18,19,21,21,21, 23)
breaks5= c( 7, 11, 13,18,20,22,22,22 ,23)
#-----
# Verificando o horário de ocorrência do pico em cada um dos dias da primeiro
# ano observado e contruindo os nós do ano
#-----
BreaksIni= breaks3
Breaksfim= BreaksIni
for( i in (1:ncol(CargaDayANO1Impar[,-182]))) {
if(Pico[i]==15) Breaksfim=c(Breaksfim, 24*i+breaks1)
if(Pico[i]==16) Breaksfim=c(Breaksfim, 24*i+breaks1)
if(Pico[i]==19) Breaksfim=c(Breaksfim, 24*i+breaks2)
if(Pico[i]==20) Breaksfim=c(Breaksfim, 24*i+breaks3)
if(Pico[i]==21) Breaksfim=c(Breaksfim, 24*i+breaks4)
if(Pico[i]==22) Breaksfim=c(Breaksfim, 24*i+breaks5) }
#-----
# Criando um base de splines para as semans ímpares do ano
#-----
PosIni= c( 7, 11, 13, 18,19,20,20,20, 23)
PosIniSem=PosIni
for(i in 1:6) PosIniSem=c(PosIniSem,PosIni+24*i )
AnualhorasImpar= (1:(26*168))-0.5
PosFinSemAnualImpar= PosIniSem
for(k in 1:25) PosFinSemAnualImpar= c(PosFinSemAnualImpar, PosIniSem +
168*k)
AnualloadbasisImpar = create.bspline.basis(rangeval=c(0,24*7*26),
norder=4,breaks=Breaksfim-0.5 )
#-----

```

```

# Aceleradores harmônicos para a constante
# -----
ConstanteBasis= create.constant.basis(c(1,(24*7*26)))
BetaAnualList= vector("list",4)
BetaAnualList[[1]]= fd(0,ConstanteBasis)
BetaAnualList[[2]]= fd((((182/(2*pi))^2)+ ((7/(2*pi))^2)),ConstanteBasis)
BetaAnualList[[3]]= fd(0,ConstanteBasis)
BetaAnualList[[4]]= fd((((182/(2*pi))^2)* ((7/(2*pi))^2)),ConstanteBasis)
Harmacellfd= Lfd(4,BetaAnualList)
BetaAnualListD= vector("list",1)
BetaAnualListD[[1]]= fd(1,ConstanteBasis)
HarmacellfdD= Lfd(1,BetaAnualListD)
# -----
#A covariada da segunda feira é a média dos dias
#úteis da semana imediatamente anterior
# -----
CovSegComp= matrix(c(CargaComp),24)[, is.element(DiasSemCom[-(1:7)],DiasAnual-7)]
CovSegAMedia= CovSegComp[,weekdays(DiasAnual-7)!="sábado"&
weekdays(DiasAnual-7)!="domingo"]
CovSeg= apply(array(c(CovSegAMedia),c(24,5,419)),3,
function(x) rowMeans(x))
CovSegImpar= CovSeg[,seq(1,416,by=2)]
DiasAnualImpar= matrix(c(t(matrix(as.character(rep
(DiasAnual,24)),length(DiasAnual))))),168)[,seq(1,416,by=2)]
LagDiasAnualImpar=c(as.Date(matrix(c(DiasAnualImpar),24)[1,],"%Y-%m-%d")-7)
MDiasAnualImpar= matrix(c(LagCargaAnualImpar),24)
MDiasAnualImpar[,c(weekdays(LagDiasAnualImpar)=="domingo")]=CovSegImpar
LagLoadAnualImpar= matrix(c(MDiasAnualImpar),,8)

```

```

# -----
# Temperatura observada para a semanas ímpares do ano
# -----
TempImpar= matrix(c(Temperatura),168,)[,seq(1,416,by=2)]
TempImparAnual= matrix(c(TempImpar),,8)
# -----
# Dummy de feriado
# -----
DAnual1= ehFeriadoMBR(DiasAnual)
DAnual2= ehFeriadoBR(DiasAnual)
DFeriado =DAnual1+DAnual2
MatrixHora= t(matrix(rep(DFeriado,24),,24))
MatrixDAnual= matrix(c(MatrixHora),,8,byrow = FALSE)
Feriados= matrix((matrix(c(MatrixDAnual),168)[,seq(1,416,2)]),,8)
# -----
# Escrevendo como combinação linear de splines a carga defasa,
# a temperatura, a carga e a dummy de feriados
# -----
LoadAnualImparfd = smooth.basis(AnualhorasImpar,LoadAnualImpar,
AnualloadbasisImpar)$fd
Teste= register.fd(LoadAnualImparfd[1],LagLoadAnualImparfd[1] )
LagLoadAnualImparfd = smooth.basis(AnualhorasImpar,LagLoadAnualImpar,
AnualloadbasisImpar)$fd
TempImparAnualbasis = create.fourier.basis(c(0,24*7*26),125,24*7*26)
TempImparAnualfd= smooth.basis(AnualhorasImpar,TempImparAnual,TempImparAnualbasis)$fd
Feriadosfd = smooth.basis(AnualhorasImpar,Feriados*LoadAnualImpar,
AnualloadbasisImpar)$fd
# -----

```

```

# Nós dos parâmetros que serão estimados
#-----
breaks1= c( 7, 11, 13,18,19,20, 23)
breaks2= c( 7, 11, 13,18,19,19,19, 23)
breaks3= c( 7, 11, 13,18,19,20,20,20, 23)
breaks4= c( 7, 11, 13,18,19,21,21,21, 23)
breaks5= c( 7, 11, 13,18,20,22,22,22 ,23)
BreaksIni= breaks3
Breaksfim= BreaksIni
for( i in (1:ncol(CargaDayANO1Impar[,-182]))) {
  if(Pico[i]==15) Breaksfim=c(Breaksfim, 24*i+breaks1)
  if(Pico[i]==16) Breaksfim=c(Breaksfim, 24*i+breaks1)
  if(Pico[i]==19) Breaksfim=c(Breaksfim, 24*i+breaks2)
  if(Pico[i]==20) Breaksfim=c(Breaksfim, 24*i+breaks3)
  if(Pico[i]==21) Breaksfim=c(Breaksfim, 24*i+breaks4)
  if(Pico[i]==22) Breaksfim=c(Breaksfim, 24*i+breaks5) }
#-----
# Criando uma lista de parâmetros para serem estimados
#-----
betaAnualList = vector("list",4)
names(betaAnualList)=c("Const","Lag","temp")
AnualBetabasis = create.bspline.basis(
rangeval=c(0,24*26*7),norder=3,breaks=Breaksfim-0.5)
betaAnualfdPar = fdPar(AnualBetabasis)
betaAnualList[[1]] = betaAnualfdPar
AnualBetabasis = create.bspline.basis(rangeval=c(0,24*26*7),norder=6,breaks=Breaksfim)
betaAnualfdPar = fdPar(AnualBetabasis,HarmacellLfd,lambda=0.00001)
betaAnualList[[2]] = betaAnualfdPar

```

```

betaAnualList[[3]] = betaAnualfdPar
#-----
#Base se fourier para os parametros da temperatura
#-----
AnualBetabasisTemp = create.fourier.basis(
rangeval=c(0,24*26*7),21)
#-----
# Populando a lista com a base de fourier para os parametros da temperatura
#-----
betaAnualfdParTemp = fdPar(AnualBetabasisTemp)
betaAnualList[[4]] = betaAnualfdParTemp
#-----
#Lista de variáveis explicativas
#-----
CovListNew=vector("list",4)
CovListNew[[1]]= rep(1,8)
CovListNew[[2]]= LagLoadAnualImparfd
CovListNew[[3]]= Feriadosfd
CovListNew[[4]]= TempImparAnualfd
#-----
# Rodando o modelo anual
#-----
ModeloAnual1= fRegress(LoadAnualImparfd,CovListNew , betaAnualList)

```

Modelos não simultâneos

Modelo semanal

```

#-----
# Dias que compõem a base de dados
#-----
DiasAnual= seq( as.Date("2003-01-05"),
as.Date("2010-12-25"),
by="days")
#-----
# Dias considerados na modelagem
#-----
DiasAnualAjt= c( rev(rev(DiasAnual[getYear(DiasAnual)==2003])[-(1:4)]),

rev(rev(DiasAnual[getYear(DiasAnual)==2004])[-(1:3)] )[-(1:6)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2005])[-1])[-(1:7)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2006])[-(1:8)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2007])[-(1:6)] )[-(1:2)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2008])[-(1:5)] )[-(1:4)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2009])[-(1:3)] )[-(1:5)]),
DiasAnual[getYear(DiasAnual)==2010] )[-(1:2)] )
#-----
# Carga avaliada nos dias que entram no modelo
#-----
CargaAnual= LoadTrab[ is.element(DiasDados,DiasAnualAjt),]
#-----
# Tirando o log da carga
#-----

```

```

LogCargaAnual=log(CargaAnual)
# -----
# Carga organizada por semanas
# -----
CargaSem=matrix(c(t(LogCargaAnual)),168)
# -----
# Excluindo os sábados domingos e segundas
# -----
NewCarga= CargaSem[-c(1:48,145:168),]
# -----
# Carga organizada de maneira anual
# -----
LoadAnual= matrix(c(NewCarga),,8)
# -----
# Dummy de feriados
# -----
DAnual1= ehFeriadoMBR(DiasAnualAjt)
DAnual2= ehFeriadoBR(DiasAnualAjt)
DFeriado =DAnual1+DAnual2
MatrixHora= t(matrix(rep(DFeriado,24),,24))
MatrixDAnual= matrix(c(MatrixHora),,8,byrow = FALSE)
Feriados= matrix((matrix(c(MatrixDAnual),168)[-c(1:48,145:168),]),,8)
# -----
# Bases
# -----
PosIni= c(1,7,10, 12,12,12, 17,
19,19,19, 20, 20, 20, 23)
PosSemAjt= PosIni

```

```

for(k in 1:3) PosSemAjt= c(PosSemAjt, PosIni + 24*k)
SemVERloadbasis = create.bspline.basis(
rangeval=c(0,4*24),
norder=4,
breaks=PosSemAjt )
#-----
# aceleradores harmônicos
#-----
ConstanteBasis= create.constant.basis(c(1,(51*4*24)))
BetaAnualList= vector("list",3)
BetaAnualList[[1]]= fd(1,ConstanteBasis)
BetaAnualList[[2]]= fd(0,ConstanteBasis)
BetaAnualList[[3]]= fd((24/(2*pi))^2,ConstanteBasis)
HarmaccelLfd= Lfd(3,BetaAnualList)
ConstanteBasis= create.constant.basis(c(1,(51*4*24)))
BetaAnualList= vector("list",3)
BetaAnualList[[1]]= fd(1,ConstanteBasis)
BetaAnualList[[2]]= fd(0,ConstanteBasis)
BetaAnualList[[3]]= fd((96/(2*pi))^2,ConstanteBasis)
HarmaccelTempLfd= Lfd(3,BetaAnualList)
#-----
# Carga defasada em 24 horas
#-----
DiasCargaDefasadaAnual= DiasAnualAjt-1
AnualLoadL=log( LoadTrab[ is.element(DiasDados,DiasCargaDefasadaAnual),])
LagLoadAnualTrab = matrix(c(t(AnualLoadL)),168)[-c(1:48,145:168),]
LagLoadAnual=matrix(c(LagLoadAnualTrab),,8)
#-----

```

```

# Dias observados e nos quais não vale o horário de verão
# -----
HorarioInverno= rep(c(1,0),9) +
as.Date( c( "2003-02-15", "2003-09-28",
"2004-02-14", "2004-11-07",
"2005-02-19", "2005-10-16",
"2006-02-18", "2006-11-05",
"2007-02-24", "2007-10-14",
"2008-02-16", "2008-10-19",
"2009-02-14", "2009-10-18",
"2010-02-20", "2010-10-17",
"2011-02-19", "2011-10-16") )
# -----
# Função que verifica amostra em que vale horário de verão
# -----
EhVerao2= apply( outer( DiasAnualAjt,
HorarioInverno,
function(x,y) x<y ),
1, sum )
EhVerao2= (EhVerao2%%2 == 0)
EhVerao2= t(matrix( rep(EhVerao2, 24), , 24) )
# -----
# criando sequência de 01/05
# -----
PrMaios= seq(from=as.Date("2003-05-01"),
to=as.Date("2010-05-01"),by="year") #
# -----

```

Se o primeiro de maio ocorre no meio da semana alguns dias da semana serão FMA
e

#outros MJJ. Para classificar a semana de forma homogênea coletamos os dias da
semana

na vizinhança do 1º de maio

#-----

```
PrMaiosIN= seq(from= as.Date(PrMaios[1])-
(as.POSIXlt(as.Date(PrMaios[1]))$wday),
to=PrMaios[1],by="days")
```

PrMaiosFN=PrMaiosIN

N= length(PrMaiosIN)

#-----

Dias que ocorrem na mesma semana em que ocorre o 1º o maio

#-----

```
for( i in 2:N){
```

```
X =seq(from= as.Date(PrMaios[i])-
(as.POSIXlt(as.Date(PrMaios[i]))$wday),
to=PrMaios[i],by="days")
```

```
PrMaiosFN=c(PrMaiosFN,X) }
```

#-----

criando sequência de 01/08

#-----

```
PrAgo= seq(from=as.Date("2003-08-01"),
to=as.Date("2010-08-01"),by="year")
```

```

PrAgoFN= seq(from= as.Date(PrAgo[1])-
(as.POSIXlt(as.Date(PrAgo[1]))$wday),
to=PrAgo[1],by="days")

N= length(PrAgo)
# -----
# dias que ocorrem na mesma semana em que ocorre o 1º de agosto
# -----
for( i in 2:N){
X =seq(from= as.Date(PrAgo[i])-
(as.POSIXlt(as.Date(PrAgo[i]))$wday),
to=PrAgo[i],by="days")
PrAgoFN=c(PrAgoFN,X) }
# -----
# Dias que estão na vizinhança do 1º de maio?
# -----
TestePrMaio= is.element(DiasAnualAjt,PrMaiosFN)

TestePrAgo= is.element(DiasAnualAjt,PrAgoFN)
# -----
# Hverao=True vale Horário de verão
# -----
Hverao=as.vector(t(EhVerao2[1,]))
# -----
# inicializando um vetor de classificações
# -----
vclasses=rep("verao",length(DiasAnualAjt))

```

```

# -----
# classificando os dias que pertencem à fevereiro e não vale horário de verão
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==1)]="FMA"
# -----
# classificando os dias que pertencem à março
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==2)]="FMA"
# -----
# classificando os dias que pertencem à abril
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==3)]="FMA"
# -----
# os dias que pertencem à semana em que o o 1º de maio está contido serão
# classificados como MJJ independente deles pertencerem a abril
# -----
vclasses[(!Hverao)&(TestePrMaio)]="MJJ"
# -----
# classificando os dias que pertencem à maio
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==4)]="MJJ"
# -----
# classificando os dias que pertencem à junho
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==5)]="MJJ"
# -----
# classificando os dias que percentencem à julho
# -----

```

```

vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==6)]="MJJ"
#-----
# os dias que pertencem à semana em que o 1º de agosto está contido serão
# classificados como ASO independente deles pertencerem a julho
#-----
vclasses[(!Hverao)&(TestePrAgo)]="ASO"
#-----
# classificando os dias que percentencem à agosto
#-----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==7)]="ASO"
#-----
# classificando os dias que percentencem à setembro
#-----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==8)]="ASO"
#-----
# classificando os dias que percentencem à outubro e não
# vale o horário de verão
#-----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==9)]="ASO"
Season= matrix(vclasses,7,)
#-----
# Separando de acordo com verao, FMA, MJJ, ASO. Classificar e depois rodar
#-----
Temperatura=dadostemp[is.element(DiasDados,DiasAnualAjt),3:26]
TemperaturaAnual= matrix(c(matrix(c(t(Temperatura)),168,)
[-c(1:48,145:168),]),,8)
TemperaturaSemTrab= matrix(c(TemperaturaAnual),96,)[,Season[1,]=="FMA"]

```

```

FeriadosSem= matrix(c(Feriados),96,)[,Season[1,]=="FMA"]
TemperaturaSem=TemperaturaSemTrab[ ,colSums(FeriadosSem)==0]
TempSembasis= create.fourier.basis(c(0,24*4),41)
TempVeraofd= smooth.basis(1:(24*4),
TemperaturaSem,TempSembasis)$fd
LoadSemVeraoTrab= matrix(c(LoadAnual),96,)[,Season[1,]=="FMA"]
LoadSemVerao=LoadSemVeraoTrab[ ,colSums(FeriadosSem)==0]
LagLoadSemVeraoTrab= matrix(c(LagLoadAnual),96,)[,Season[1,]=="FMA"]
LagLoadSemVerao=LagLoadSemVeraoTrab[ ,colSums(FeriadosSem)==0]
LoadVeraofd = smooth.basisPar(1:96,LoadSemVerao,
SemVERloadbasis, lambda=0.05)$fd
LagLoadVeraofd = smooth.basisPar(1:96,LagLoadSemVerao,
SemVERloadbasis, lambda=0.05)$fd
# -----
# Projetando a carga na carga defasada
# -----
betaSemList=vector("list",1)
betaSemList[[1]]= fdPar( SemINVloadbasis )

CovListNew=vector("list",1)
CovListNew[[1]]= LagLoadVeraofd
ModeloSemGeral= fRegress(LoadVeraofd,CovListNew , betaSemList)
# -----
# Projetando a temperatura na carga defasada
# -----
CovListNew=vector("list",1)
CovListNew[[1]]= LagLoadVeraofd
ModeloTempSem= fRegress(TempVeraofd , CovListNew , betaSemList)

```

```

# -----
#Obtendo a carga e a temperatura com controle o efeito da carga defasada
# -----
LoadAdj= eval.fd(1:(24*4) ,LoadVeraofd-ModeloSemGeral$yhatfd$fd )
TempAdj= eval.fd(1:(24*4) ,TempVeraofd-ModeloTempSem$yhatfd$fd )
# -----
# Estimando as HIF's
# -----
Int= vector("list",16)
Int[[1]]=1:8
Int[[2]]=5:14
Int[[3]]=11:20
Int[[4]]=17:24

Nbases=3

HIFListV=vector("list",4)

# -----
# for nos quatro intervalos do dia
# -----
for ( l in 1:4){
# -----
# Bases para cada um dos intervalos
# -----
RRRBasis = create.bspline.basis(c(Int[[1]][1],Int[[1]][length(Int[[1]])]),
norder=4,nbasis=length(Int[[1]])-Nbases)

```

```

RRRBasisFDPAR=fdPar(RRRBasis)$fd
#—————
# Variável explicativa em forma matricial
#—————
TempAdjDay= matrix(c(TempAdj),24)[Int[[1]], ]

Xifd= smooth.basisPar(Int[[1]],
TempAdjDay,RRRBasisFDPAR)$fd

#—————
# variável dependente em forma matricial
#—————
LoadAdjDay= matrix(c(LoadAdj),24)[Int[[1]], ]

Yifd= smooth.basis(Int[[1]],LoadAdjDay,
RRRBasisFDPAR)$fd

PsiXC= t(Xifd$coef)
#—————
# obtendo a matrix de variância da variável explicativa
#—————
CovPsiX= array(0,c(ncol(PsiXC),
ncol(PsiXC),n))

```

```

for(k in 1:n){
CovPsiX[, ,k]=outer(PsiXC[k,],PsiXC[k,])}
ECovPsiX=matrix(NA,ncol(PsiXC),
ncol(PsiXC))
for(i in 1:ncol(PsiXC)){
for(j in 1:ncol(PsiXC)){
ECovPsiX[i,j]=mean(CovPsiX[i,j,1:n]) }}
EGXX=ECovPsiX
#—————
# obtendo a matrix de variância da variável dependente
#—————
PsiYC= t(Yifd$coef)

CovPsiY= array(0,c(ncol(PsiYC),
ncol(PsiYC),n))
for(k in 1:n){
CovPsiY[, ,k]=outer(PsiYC[k,],PsiYC[k,])}

ECovPsiY=matrix(NA,ncol(PsiYC),
ncol(PsiYC))
for(i in 1:ncol(PsiYC)){
for(j in 1:ncol(PsiYC)){
ECovPsiY[i,j]=mean(CovPsiY[i,j,1:n]) }}
EGYY=ECovPsiY
#—————
# obtendo a matrix de variância cruzada
#—————
CovPsiXY= array(0,c(ncol(PsiYC),

```

```

ncol(PsiYC),n))
for(k in 1:n){
CovPsiXY[, ,k]=outer(PsiXC[k,],PsiYC[k,])}
ECovPsiXY=matrix(NA,ncol(PsiYC),
ncol(PsiYC))
for(i in 1:ncol(PsiYC)){
for(j in 1:ncol(PsiYC)){
ECovPsiXY[i,j]=mean(CovPsiXY[i,j,1:n]) }}
EGXY=ECovPsiXY
EGYX= t(EGXY)
#—————
# tirando a raiz da matriz de covariância da variável dependente
#—————
GamafdAMEio= eigen(EGYY)$vectors[,1:4]%%
diag(eigen(EGYY)$values[1:4]^(0.5))%%
t(eigen(EGYY)$vectors[,1:4])
#—————
# Elevando a menos meio a matriz de covariância da variável dependente
#—————
GamafdAMenosMeio= eigen(EGYY)$vectors[,1:4]%%
diag(eigen(EGYY)$values[1:4]^(-0.5))%%
t(eigen(EGYY)$vectors[,1:4])
#—————
# obtendo a inversa da matriz de covariância da variável explicativa
#—————
EGXXAMenos1=
eigen(EGXX)$vectors[,1:4]%%
diag(eigen(EGXX)$values[1:4]^(-1))%%

```

```

t(eigen(EGXX)$vectors[,1:4])
KeyFunc= GamafdAMEio %*% EGYX %*% EGXXAMenos1%*%
EGXY%*%GamafdAMEio

BfdCoefs= t(eigen(KeyFunc)$vectors[,1])%*%
GamafdAMEio%*% EGYX%*%EGXXAMenos1

AfdVBfdCoefs =Re(c(BfdCoefs))
HistFunc= fd(AfdVBfdCoefs,RRRBasis)

Padronizacao= sqrt(0.1*sum(eval.fd(seq(
Int[[1]][1],Int[[1]]
[length(Int[[1]])],
by=0.1),HistFunc)^2))
HistDis=AfdVBfdCoefs /Padronizacao
Argvals=Int[[1]]
HistDisFunc= fd(HistDis,RRRBasis)
HIFListV[[1]]= HistDisFunc

}

HIFList[[4]]=-1*HIFList[[4]]

```

```

#-----
#obtendo a integral da temperatura multiplicada pela HIF
#-----
IntNew= vector("list",4)
IntNew[[1]]=1:6
IntNew[[2]]=7:12
IntNew[[3]]=13:18
IntNew[[4]]=19:24

Trab1= eval.fd(IntNew[[1]],HIFList[[1]])
for(l in 2:4){

Trab1= c(Trab1,eval.fd(IntNew[[l]],HIFList[[l]])) }
HIFEst= Trab1

TrabTeste=matrix(rep(HIFEst,ncol(TemperaturaSem)*4),6)

IntHIFM= matrix(c(t(matrix(rep(colSums(TrabTeste*matrix(c(TemperaturaSem),
6)),6),ncol(TrabTeste))))),96)

IntHIFfd=numeric(0)
BreaksHIFIN= c(c(0,6,6,6,12,12,12,18,18,18,24),
c(0,6,6,6,12,12,12,18,18,18,24)+24,
c(0,6,6,6,12,12,12,18,18,18,24)+(2*24),
c(0,6,6,6,12,12,12,18,18,18,24)+(3*24))
IntHIFfd=
smooth.basis(1:(4*24),IntHIFM,

```

```

create.bspline.basis(
rangeval=c(0,4*24),
norder=4,breaks= BreaksHIFIN )$fd
#-----
#Base dos parâmetros da carga e da carga defasada
#-----
BreaksNew= c(c( 7,9 ,10,13,18,19,20, 23) ,
c( 7,9, 10,13,18,19,20, 23)+24,
c( 7,9, 10,13,18,19,20, 23)+(2*24),
c( 7,9, 10,13,18,19,20, 23)+(3*24))
#-----
# Base dos parâmetros da temperatura
#-----
BreaksHIFIN2= c(0,6,7,12,13,18,19,24)
BreaksHIF2 = BreaksHIFIN2
for(i in 1:(4-1)) {
BreaksHIF2 = c(BreaksHIF2,BreaksHIFIN2+(i*24))}
#-----
# Modelo com HIF
#-----
betaSemList=vector("list",3)
SemBetabasis = create.bspline.basis(
rangeval=c(0,24*4),norder=4,
breaks=BreaksNew)

betaSemList[[1]] = fdPar(SemBetabasis)

SemBetabasis = create.bspline.basis(

```

```

rangeval=c(0,24*4),norder=5,
breaks=BreaksNew)

betaSemList[[2]] = fdPar(SemBetabasis,HarmacelLfd,lambda=0.01) #0.02
betaSemList[[3]] = fdPar(create.bspline.basis(
rangeval=c(0,24*4),norder=5,
breaks=BreaksHIF2),HarmacelLfd,lambda=190)

CovListNew=vector("list",3)
CovListNew[[1]]= rep(1,ncol(LoadVeraofd$coef))
CovListNew[[2]]= LagLoadVeraofd
CovListNew[[3]]= IntHIFfd

ModWeekCHIF= fRegress(LoadVeraofd,CovListNew , betaSemList)

```

Modelo anual

```

#-----
# Dias que compõem a base de dados
#-----
DiasAnual= seq( as.Date("2003-01-05"),
as.Date("2010-12-25"), by="days")
#-----
# Dias considerados na modelagem
#-----
DiasAnualAjt= c( rev(rev(DiasAnual[getYear(DiasAnual)==2003])[-(1:4)]),

rev(rev(DiasAnual[getYear(DiasAnual)==2004])[-(1:3)])[- (1:6)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2005])[-1])[-(1:7)]),

```

```

rev(rev(DiasAnual[getYear(DiasAnual)==2006])[-(1:8)]),
rev(rev(DiasAnual[getYear(DiasAnual)==2007])[-(1:6)])[-(1:2)],
rev(rev(DiasAnual[getYear(DiasAnual)==2008])[-(1:5)])[-(1:4)],
rev(rev(DiasAnual[getYear(DiasAnual)==2009])[-(1:3)])[-(1:5)],
DiasAnual[getYear(DiasAnual)==2010])[-(1:2)] )
# -----
# Carga avaliada nos dias que entram no modelo
# -----
CargaAnual= LoadTrab[ is.element(DiasDados,DiasAnualAjt),]
# -----
# Tirando o log da carga
# -----
# -----
# Carga organizada por semanas
# -----
CargaSem=matrix(c(t(LogCargaAnual)),168)
# -----
# Excluindo os sábados domingos e segundas
# -----
NewCarga= CargaSem[-c(1:48,145:168),]
# -----
# Carga organizada de maneira anual
# -----
LoadAnual= matrix(c(NewCarga),,8)
# -----
# Dummy de feriados
# -----
DAnual1= ehFeriadoMBR(DiasAnualAjt)

```

```

DAnual2= ehFeriadoBR(DiasAnualAjt)
DFeriado =DAnual1+DAnual2
MatrixHora= t(matrix(rep(DFeriado,24),,24))
MatrixDAnual= matrix(c(MatrixHora),,8,byrow = FALSE)
Feriados= matrix((matrix(c(MatrixDAnual),168)[-c(1:48,145:168),]),,8)
#-----
# Bases
#-----
PosIni= c( 7,10 ,11, 13, 13, 13, 18,18,18, 19, 20, 20, 20, 23)
PosSemAjt= PosIni
for(k in 1:3) PosSemAjt= c(PosSemAjt, PosIni + 24*k)
SemVERloadbasis = create.bspline.basis(
rangeval=c(0,4*24), norder=4, breaks=PosSemAjt )

PosIni= c( 7, 11, 13, 18, 19,19, 19, 21,23)
PosSemAjt= PosIni
for(k in 1:3) PosSemAjt= c(PosSemAjt, PosIni + 24*k)

SemINVloadbasis = create.bspline.basis(
rangeval=c(0,4*24),norder=4,breaks=PosSemAjt )
#-----
# aceleradores harmônicos
#-----
ConstanteBasis= create.constant.basis(c(1,(51*4*24)))
BetaAnualList= vector("list",3)
BetaAnualList[[1]]= fd(1,ConstanteBasis)
BetaAnualList[[2]]= fd(0,ConstanteBasis)

```

```

BetaAnualList[[3]]= fd((24/(2*pi))^2,ConstanteBasis)
HarmacellLfd= Lfd(3,BetaAnualList)
ConstanteBasis= create.constant.basis(c(1,(51*4*24)))
BetaAnualList= vector("list",3)
BetaAnualList[[1]]= fd(1,ConstanteBasis)
BetaAnualList[[2]]= fd(0,ConstanteBasis)
BetaAnualList[[3]]= fd((96/(2*pi))^2,ConstanteBasis)
HarmacelTempLfd= Lfd(3,BetaAnualList)
# -----
# Carga defasada em 24 horas
# -----
DiasCargaDefasadaAnual= DiasAnualAjt-1
AnualLoadL=log( LoadTrab[ is.element(DiasDados,DiasCargaDefasadaAnual),])
LagLoadAnualTrab = matrix(c(t(AnualLoadL)),168)[-c(1:48,145:168),]
LagLoadAnual=matrix(c(LagLoadAnualTrab),,8)
# -----
# Dias observados e nos quais não vale o horário de verão
# -----
HorarioInverno= rep(c(1,0),9) +
as.Date( c( "2003-02-15", "2003-09-28",
"2004-02-14", "2004-11-07",
"2005-02-19", "2005-10-16",
"2006-02-18", "2006-11-05",
"2007-02-24", "2007-10-14",
"2008-02-16", "2008-10-19",
"2009-02-14", "2009-10-18",
"2010-02-20", "2010-10-17",
"2011-02-19", "2011-10-16" ) )

```

```

# -----
# Função que verifica amostra em que vale horário de verão
# -----
EhVerao2= apply( outer( DiasAnualAjt,

HorarioInverno, function(x,y) x<y ),1, sum )
EhVerao2= (EhVerao2%%2 == 0)
EhVerao2= t(matrix( rep(EhVerao2, 24), , 24) )
# -----
# criando sequência de 01/05
# -----
PrMaios= seq(from=as.Date("2003-05-01"),
to=as.Date("2010-05-01"),by="year") #
# -----
# Se o primeiro de maio ocorre no meio da semana alguns dias da semana serão FMA
e
#outros MJJ. Para classificar a semana de forma homogênea coletamos os dias da
semana
# na vizinhança do 1º de maio
# -----
PrMaiosIN= seq(from= as.Date(PrMaios[1])-
(as.POSIXlt(as.Date(PrMaios[1]))$wday),to=PrMaios[1],by="days")
PrMaiosFN=PrMaiosIN
N= length(PrMaiosIN)
# -----
# Dias que ocorrem na mesma semana em que ocorre o 1º o maio
# -----
for( i in 2:N){

```

```

X =seq(from= as.Date(PrMaios[i])-
(as.POSIXlt(as.Date(PrMaios[i]))$yday),
to=PrMaios[i],by="days")
PrMaiosFN=c(PrMaiosFN,X) }
# -----
# criando sequência de 01/08
# -----
PrAgo= seq(from=as.Date("2003-08-01"),
to=as.Date("2010-08-01"),by="year")
PrAgoFN= seq(from= as.Date(PrAgo[1])-
(as.POSIXlt(as.Date(PrAgo[1]))$yday),
to=PrAgo[1],by="days")

N= length(PrAgo)
# -----
# dias que ocorrem na mesma semana em que ocorre o 1º de agosto
# -----
for( i in 2:N){
X =seq(from= as.Date(PrAgo[i])-
(as.POSIXlt(as.Date(PrAgo[i]))$yday),
to=PrAgo[i],by="days")
PrAgoFN=c(PrAgoFN,X) }
# -----
# Dias que estão na vizinhança do 1º de maio?
# -----
TestePrMaio= is.element(DiasAnualAjt,PrMaiosFN)

```

```

TestePrAgo= is.element(DiasAnualAjt,PrAgoFN)
# -----
# Hverao=True vale Horário de verão
# -----
Hverao=as.vector(t(EhVerao2[1,]))
# -----
# inicializando um vetor de classificações
# -----
vclasses=rep("verao",length(DiasAnualAjt))
# -----
# classificando os dias que pertencem à fevereiro e não vale horário de verão
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==1)]="FMA"
# -----
# classificando os dias que pertencem à março
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==2)]="FMA"
# -----
# classificando os dias que pertencem à abril
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==3)]="FMA"
# -----
# os dias que pertencem à semana em que o 1º de maio está contido serão
# classificados como MJJ independente deles pertencerem a abril
# -----
vclasses[(!Hverao)&(TestePrMaio)]="MJJ"
# -----
# classificando os dias que pertencem à maio

```

```

# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==4)]="MJJ"
# -----
# classificando os dias que pertencem à junho
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==5)]="MJJ"
# -----
# classificando os dias que percentencem à julho
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==6)]="MJJ"
# -----
# os dias que pertencem à semana em que o 1º de agosto está contido serão
# classificados como ASO independente deles pertencerem a julho
# -----
vclasses[(!Hverao)&(TestePrAgo)]="ASO"
# -----
# classificando os dias que percentencem à agosto
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==7)]="ASO"
# -----
# classificando os dias que percentencem à setembro
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==8)]="ASO"
# -----
# classificando os dias que percentencem à outubro e não
# vale o horário de verão
# -----
vclasses[(!Hverao)&(as.POSIXlt(DiasAnualAjt)$mon==9)]="ASO"

```

```

Season= matrix(vclasses,7,)
#-----
# Separando de acordo com verao, FMA, MJJ, ASO.
#-----
Temperatura=dadostemp[is.element(DiasDados,DiasAnualAjt),3:26]
TemperaturaAnual= matrix(c(matrix(c(t(Temperatura)),168,)
[-c(1:48,145:168),]),,8)
TemperaturaSemTrab= matrix(c(TemperaturaAnual),96,)[,Season[1,]=="ASO"]

FeriadosSem= matrix(c(Feriados),96,)[,Season[1,]=="ASO"]
TemperaturaSem=TemperaturaSemTrab[ ,colSums(FeriadosSem)==0]
TempSembasis= create.fourier.basis(c(0,24*4),41)
TempVeraofd= smooth.basis(1:(24*4),
TemperaturaSem,TempSembasis)$fd
LoadSemVeraoTrab= matrix(c(LoadAnual),96,)[,Season[1,]=="ASO"]
LoadSemVerao=LoadSemVeraoTrab[ ,colSums(FeriadosSem)==0]
LagLoadSemVeraoTrab= matrix(c(LagLoadAnual),96,)[,Season[1,]=="ASO"]
LagLoadSemVerao=LagLoadSemVeraoTrab[ ,colSums(FeriadosSem)==0]
LoadVeraofd = smooth.basisPar(1:96,LoadSemVerao,
SemVERloadbasis, lambda=0.05)$fd
LagLoadVeraofd = smooth.basisPar(1:96,LagLoadSemVerao,
SemVERloadbasis, lambda=0.05)$fd
#-----
# Projetando a carga na carga defasada
#-----
betaSemList=vector("list",1)
betaSemList[[1]]= fdPar( SemINVloadbasis )
CovListNew=vector("list",1)

```

```

CovListNew[[1]]= LagLoadVeraofd
ModeloSemGeral= fRegress(LoadVeraofd,CovListNew , betaSemList)
#-----
# Projetando a temperatura na carga defasada
#-----
CovListNew=vector("list",1)
CovListNew[[1]]= LagLoadVeraofd
ModeloTempSem= fRegress(TempVeraofd , CovListNew , betaSemList)
#-----
#Obtendo a carga e a temperatura com controle o efeito da carga defasada
#-----
LoadAdj= eval.fd(1:(24*4) ,LoadVeraofd-ModeloSemGeral$yhatfd$fd )
TempAdj= eval.fd(1:(24*4) ,TempVeraofd-ModeloTempSem$yhatfd$fd )
#-----
# Estimando as HIF's
#-----
Int= vector("list",16)
Int[[1]]=1:8
Int[[2]]=5:14
Int[[3]]=11:20
Int[[4]]=17:24
Nbases=2
HIFListSem=vector("list",4)
#-----
# for nos quatro intervalos do dia
#-----
for ( l in 1:4){
#-----

```

```

# Bases para cada um dos intervalos
# -----
RRRBasis = create.bspline.basis(c(Int[[1]][1],Int[[1]][length(Int[[1])])),
norder=4,nbasis=length(Int[[1]])-Nbases)
RRRBasisFDPAR=fdPar(RRRBasis)$fd
# -----
# Variável explicativa em forma matricial
# -----
TempAdjDay= matrix(c(TempAdj),24)[Int[[1]], ]
Xifd= smooth.basisPar(Int[[1]],
TempAdjDay,RRRBasisFDPAR)$fd
# -----
# variável dependente em forma matricial
# -----
LoadAdjDay= matrix(c(LoadAdj),24)[Int[[1]], ]
Yifd= smooth.basis(Int[[1]],LoadAdjDay,
RRRBasisFDPAR)$fd
PsiXC= t(Xifd$coef)
# -----
# obtendo a matrix de variância da variável explicativa
# -----
CovPsiX= array(0,c(ncol(PsiXC),
ncol(PsiXC),n))
for(k in 1:n){
CovPsiX[, ,k]=outer(PsiXC[k,],PsiXC[k,])}
ECovPsiX=matrix(NA,ncol(PsiXC),
ncol(PsiXC))
for(i in 1:ncol(PsiXC)){

```

```

for(j in 1:ncol(PsiXC)){
ECovPsiX[i,j]=mean(CovPsiX[i,j,1:n]) }}
EGXX=ECovPsiX
#—————
# Obtendo a matrix de variância da variável dependente
#—————
PsiYC= t(Yifd$coef)
CovPsiY= array(0,c(ncol(PsiYC),
ncol(PsiYC),n))
for(k in 1:n){
CovPsiY[, ,k]=outer(PsiYC[k,],PsiYC[k,])}
ECovPsiY=matrix(NA,ncol(PsiYC),
ncol(PsiYC))
for(i in 1:ncol(PsiYC)){
for(j in 1:ncol(PsiYC)){
ECovPsiY[i,j]=mean(CovPsiY[i,j,1:n]) }}
EGYY=ECovPsiY
#—————
# obtendo a matrix de variância cruzada
#—————
CovPsiXY= array(0,c(ncol(PsiYC),
ncol(PsiYC),n))
for(k in 1:n){
CovPsiXY[, ,k]=outer(PsiXC[k,],PsiYC[k,])}
ECovPsiXY=matrix(NA,ncol(PsiYC),
ncol(PsiYC))
for(i in 1:ncol(PsiYC)){
for(j in 1:ncol(PsiYC)){

```

```

ECovPsiXY[i,j]=mean(CovPsiXY[i,j,1:n]) }}
EGXY=ECovPsiXY
EGYX= t(EGXY)
#-----
# Tirando a raiz da matriz de covariância da variável dependente
#-----
GamafdAMEio= eigen(EGYY)$vectors[,1:4]%%
diag(eigen(EGYY)$values[1:4])^(0.5)%%
t(eigen(EGYY)$vectors[,1:4])
#-----
# Elevando a menos meio a matriz de covariância da variável dependente
#-----
GamafdAMenosMeio= eigen(EGYY)$vectors[,1:4]%%
diag(eigen(EGYY)$values[1:4]^(-0.5))%%
t(eigen(EGYY)$vectors[,1:4])
#-----
# obtendo a inversa da matriz de covariância da variável explicativa
#-----
EGXXAMenos1= eigen(EGXX)$vectors[,1:4]%%
diag(eigen(EGXX)$values[1:4]^(-1))%%
t(eigen(EGXX)$vectors[,1:4])
KeyFunc= GamafdAMEio %% EGYX %% EGXXAMenos1%%
EGXY%%GamafdAMEio
BfdCoefs= t(eigen(KeyFunc)$vectors[,1])%%
GamafdAMEio%% EGYX%%EGXXAMenos1
AfdVBfdCoefs =Re(c(BfdCoefs))
HistFunc= fd(AfdVBfdCoefs,RRRBasis)
Padronizacao= sqrt(0.1*sum(eval.fd(seq(

```

```

Int[[1]][1],Int[[1]]
[length(Int[[1]])],
by=0.1),HistFunc)^2))
HistDis=AfdVBfdCoefs /Padronizacao
Argvals=Int[[1]]
HistDisFunc= fd(HistDis,RRRBasis)
HIFList[[1]]= HistDisFunc }
#-----
#obtendo a integral da temperatura multiplicada pela HIF
#-----
IntNew= vector("list",4)
IntNew[[1]]=1:6
IntNew[[2]]=7:12
IntNew[[3]]=13:18
IntNew[[4]]=19:24
Trab1= eval.fd(IntNew[[1]],HIFList[[1]])
for(l in 2:4){
Trab1= c(Trab1,eval.fd(IntNew[[l]],HIFList[[l]])) }
HIFEst= Trab1
TrabTeste=matrix(rep(HIFEst,ncol(TemperaturaSem)*4),6)
IntHIFM= matrix(c(t(matrix(rep(colSums(TrabTeste*matrix(c(TemperaturaSem),
6)),6),ncol(TrabTeste))))),96)
IntHIFfd=numeric(0)
BreaksHIFIN= c(c(0,6,6,6,6,12,12,12,12,18,18,18,24),
c(0,6,6,6,6,12,12,12,18,18,18,24)+24,
c(0,6,6,6,6,12,12,12,18,18,18,24)+(2*24),
c(0,6,6,6,6,12,12,12,18,18,18,24)+(3*24))
IntHIFfd=

```

```

smooth.basis(1:(4*24),IntHIFM,create.bspline.basis(
rangeval=c(0,4*24),norder=4,breaks= BreaksHIFIN ))$fd
#-----
#Base dos parâmetros da carga e da carga defasada
#-----
BreaksNew= c(c( 7,9 ,10,13,18,19,20, 23) ,
c( 7,9, 10,13,18,19,20, 23)+24,
c( 7,9, 10,13,18,19,20, 23)+(2*24),
c( 7,9, 10,13,18,19,20, 23)+(3*24))
#-----
# Base dos parâmetros da temperatura
#-----
BreaksHIFIN2= c(0,6,7,12,13,18,19,24)
BreaksHIF2 = BreaksHIFIN2
for(i in 1:(4-1)) {
BreaksHIF2 = c(BreaksHIF2,BreaksHIFIN2+(i*24))}
#-----
# Modelo com HIF
#-----
betaSemList=vector("list",3)
SemBetabasis = create.bspline.basis(
rangeval=c(0,24*4),norder=4,breaks=BreaksNew)
betaSemList[[1]] = fdPar(SemBetabasis)
SemBetabasis = create.bspline.basis(
rangeval=c(0,24*4),norder=5,
breaks=BreaksNew)
betaSemList[[2]] = fdPar(SemBetabasis,HarmacellLfd,lambda=0.01) #0.02
betaSemList[[3]] = fdPar(create.bspline.basis(

```

```

rangeval=c(0,24*4),norder=5,
breaks=BreaksHIF2),Harmacellfd,lambda=95)
CovListNew=vector("list",3)
CovListNew[[1]]= rep(1,ncol(LoadVeraofd$coef))
CovListNew[[2]]= LagLoadVeraofd
CovListNew[[3]]= IntHIFfd
ModWeekCHIF= fRegress(LoadVeraofd,CovListNew , betaSemList)
saidahat = eval.fd(c(1:(4*24)), ModWeekCHIF$yhatfdobj$fd)
entrada = eval.fd(c(1:(4*24)), ModWeekCHIF$yfdPar$fd)
residuo = entrada - saidahat
SESem=residuo^2
MSESem=rep(NA,24)
for(i in 1:(4*24)) MSESem[i] = sum( SESem[i,] )/ncol(SESem)
MSEWeekComHIF=sqrt(MSESem)*100
names(MSEWeekComHIF)=c(1:(4*24))
#-----
# Modelo sem HIF
#-----
betaSemList=vector("list",3)
SemBetabasis = create.bspline.basis(
rangeval=c(0,24*4),norder=4,breaks=BreaksNew)
betaSemList[[1]] = fdPar(SemBetabasis)
SemBetabasis = create.bspline.basis(
rangeval=c(0,24*4),norder=5,breaks=BreaksNew)
betaSemList[[2]] = fdPar(SemBetabasis,Harmacellfd,lambda=0.009) #0.02
betaSemList[[3]] = fdPar(create.bspline.basis(
rangeval=c(0,24*4),norder=5, breaks=BreaksNew),Harmacellfd,lambda=40)
CovListNew=vector("list",3)

```

```
CovListNew[[1]]= rep(1,ncol(LoadVeraofd$coef))
CovListNew[[2]]= LagLoadVeraofd
CovListNew[[3]]= TempVeraofd
ModWeekSemHIF= fRegress(LoadVeraofd,CovListNew , betaSemList)
saidahat = eval.fd(c(1:(4*24)), ModWeekSemHIF$yhatfdobj$fd)
entrada = eval.fd(c(1:(4*24)), ModWeekSemHIF$yfdPar$fd)
residuo = entrada - saidahat
SESem=residuo^2
MSESem=rep(NA,24)
for(i in 1:(4*24)) MSESem[i] = sum( SESem[i,] )/ncol(SESem)
MSEWeekSemHIF=sqrt(MSESem)*100
names(MSEWeekSemHIF)=c(1:(4*24))
```