

Comandos (em R) para reproduzir as análises de exemplos do livro *Análise de Dados Categorizados* de Paulino e Singer (2006)

FREDERICO ZANQUETA POLETO
fred@poleto.com
20/04/2007

Conteúdo

1	Introdução	1
2	Alguns comandos básicos do R	1
3	Instalação da biblioteca de rotinas Catdata	5
4	Utilização das rotinas para a análise de dados completamente categorizados	5
	readCatdata(): leitura dos dados	5
	linML(): ajuste de modelos lineares por máxima verossimilhança	6
	loglinML(): ajuste de modelos log-lineares por máxima verossimilhança	9
	funlinWLS(): ajuste de modelos funcionais lineares por mínimos quadrados generalizados . . .	12
	waldTest(): teste de Wald	17
5	Utilização das rotinas para a análise de dados categorizados com omissão	18
	readCatdata(): leitura dos dados	18
	satMarML(): ajuste de modelos saturados por máxima verossimilhança sob os mecanismos de omissão MAR e MCAR	20
	satMcarWLS(): ajuste de modelos saturados por mínimos quadrados generalizados sob o mecanismo de omissão MCAR	25
	linML(): ajuste de modelos lineares por máxima verossimilhança	27
	loglinML(): ajuste de modelos log-lineares por máxima verossimilhança	27
	funlinWLS(): ajuste de modelos funcionais lineares por mínimos quadrados generalizados . . .	33
	waldTest(): teste de Wald	43
	Hierarquia de uso das rotinas	43
	Análises possíveis de serem realizadas com a biblioteca Catdata	44
6	Comandos para reproduzir as análises dos exemplos	44
	8.1 (p.228) / 3.1 (p.47): <i>Problema da intenção de voto</i>	45
	8.2 (p.233) / 3.2 (p.49) / 1.2 (p.4): <i>Problema do risco de cárie dentária</i>	45
	8.3 (p.236) / 3.3 (p.50) / 1.9 (p.12): <i>Problema do tamanho da ninhada</i>	45
	9.1 (p.263): <i>Problema da anemia</i>	46

9.2 (p.267): <i>Problema da acuidade visual</i>	46
9.3 (p.269) / 1.6 (p.11): <i>Problema dos defeitos de fibras têxteis</i>	48
9.4 (p.274) / 1.11 (p.13): <i>Problema da fobia em alcoólatras</i>	49
9.5 (p.278) / Exercício 8.12 (p.244): <i>Problema da obesidade juvenil</i>	50
9.6 (p.281) / 9.3 (p.269) / 1.6 (p.11): <i>Problema dos defeitos de fibras têxteis</i>	51
9.7 (p.285): <i>Problema da toxicodependência</i>	52
9.10 (p.304) / 1.4 (p.5): <i>Problema dos grafiteiros</i>	54
9.11 (p.307) / 1.5 (p.5): <i>Problema do uso do fio dental</i>	54
9.12 (pp.316, 317, 319, 324, 326): <i>Problema da aterosclerose coronariana</i>	55
10.1 (p.348) / 6.1 (p.149): <i>Problema da intoxicação de besouros</i>	57
10.2 (p.349) / 6.3 (p.156) / 1.2 (p.4): <i>Problema do risco de cárie dentária</i>	57
10.3 (p.349) / 6.4 (p.157) / 1.5 (p.5): <i>Problema do uso do fio dental</i>	58
10.4 (p.351) / 6.5 (p.160): <i>Problema da complicação pulmonar</i>	58
10.5 (p.353) / 6.6 (p.163) / 1.3 (p.4): <i>Problema do peso de recém-nascidos</i>	58
10.6 (p.357) / 6.7 (p.167) / 1.3 (p.4): <i>Problema do peso de recém-nascidos</i>	59
10.8 (pp.360, 367): <i>Problema da suscetibilidade a malária cerebral</i>	60
11.1 (p.376) / 7.1 (p.202) / 1.7 (p.11): <i>Problema do grupo sanguíneo ABO</i>	60
11.2 (p.382) / 8.1 (p.228) / 3.1 (p.47): <i>Problema da intenção de voto</i>	60
11.3 (p.383) / 8.2 (p.233) / 3.2 (p.49) / 1.2 (p.4): <i>Problema do risco de cárie dentária</i>	61
11.4 (p.384) / 8.3 (p.236) / 3.3 (p.50) / 1.9 (p.12): <i>Problema do tamanho da ninhada</i>	61
11.5 (p.388) / 9.1 (p.263): <i>Problema da anemia</i>	62
11.6 (p.388) / 9.2 (p.267): <i>Problema da acuidade visual</i>	62
11.7 (p.390) / 9.3 (p.269) / 1.6 (p.11): <i>Problema dos defeitos de fibras têxteis</i>	64
11.8 (p.392) / 10.2 (p.349) / 6.3 (p.156) / 1.2 (p.4): <i>Problema do risco de cárie dentária</i>	65
11.9 (p.393) / 10.3 (p.349) / 6.4 (p.157) / 1.5 (p.5): <i>Problema do uso do fio dental</i>	65
11.10 (p.395) / 10.4 (p.351) / 6.5 (p.160): <i>Problema da complicação pulmonar</i>	66
11.11 (p.395) / 10.5 (p.353) / 6.6 (p.163) / 1.3 (p.4): <i>Problema do peso de recém-nascidos</i>	66
11.12 (p.399) / 6.8 (p.169) / 1.2 (p.4): <i>Problema do risco de cárie dentária</i>	67
11.13 (p.399): <i>Problema da poluição por petróleo</i>	67
12.1 (pp.419, 427, 442): <i>Problema da infecção urinária</i>	68
12.2 (pp.421, 431, 444): <i>Problema da sensibilidade dentinária</i>	70
12.3 (pp.422, 432): <i>Problema da maturação do colo do útero</i>	70
13.2 (p.466)	72
13.3 (p.469)	72
13.4 (p.472) / 13.1 (p.454)	73

Agradecimentos **87**

Referências **88**

1 Introdução

O objetivo deste trabalho é ilustrar como se pode reproduzir a maior parte das análises apresentadas nos exemplos dos Capítulos 8 ao 13 de Paulino e Singer (2006) utilizando as rotinas do pacote `Catdata` desenvolvidas para o ambiente estatístico R (R Development Core Team, 2006). As rotinas foram implementadas durante o projeto de mestrado de Poletto (2006) cujo objetivo é analisar dados categorizados com omissão por meio das metodologias de máxima verossimilhança (MV) ou mínimos quadrados generalizados (MQG). Esse tipo de dados ocorre quando as respostas em algumas variáveis para uma parte das unidades experimentais não são observadas. A teoria apresentada em Poletto (2006) e Poletto, Singer e Paulino (2007a) estende os resultados do Capítulo 13 de Paulino e Singer (2006) para permitir a inclusão de variáveis explicativas. Como dados completos podem ser encarados como um caso particular de dados com omissão, as rotinas também permitem a sua análise.

Para a concretização das análises, primeiramente é necessário obter o ambiente R em

<http://cran.r-project.org>

e realizar a sua instalação. Informações de como instalar o R, bem como diversos documentos explicativos da linguagem estão disponíveis no sítio. Alguns comandos básicos que serão utilizados são apresentados na Seção 2 deste documento. Portanto, não é necessário que se conheça previamente a linguagem. Todavia, uma certa familiaridade com os comandos do R ou S-Plus podem ser úteis. As explicações são simples e direcionadas aos objetivos deste trabalho. Mais detalhes sobre o R, além do seu sítio, podem ser encontrados em Thompson (2007), Venables e Ripley (2000, 2002), Chambers e Hastie (1992), Chambers (1998), Ribeiro Jr. (sem data) e Paula (2000).

As notações utilizadas no decorrer deste texto têm significado igual ao utilizado por Paulino e Singer (2006)¹. Por conveniência, não se menciona, mas todas as referências a exemplos e fórmulas são direcionadas a Paulino e Singer (2006), a não ser que se explicita o contrário. Na Seção 3, ensina-se como carregar a biblioteca de rotinas no ambiente R. Na Seção 4, apresenta-se alguns detalhes sobre cada uma das rotinas para a análise de dados completos. Na Seção 5, descreve-se as diferenças nas entradas de dados e nos resultados obtidos pelas rotinas ao realizar a análise de dados incompletos. Mais detalhes e uma introdução ao pacote `Catdata` com outros exemplos podem ser encontrados em Poletto, Singer e Paulino (2007b). A Seção 6 contém os comandos para reproduzir os resultados dos exemplos do livro.

Pode-se encontrar a última versão deste documento, bem como outros materiais relacionados, em

<http://www.poletto.com/missing.html>.

2 Alguns comandos básicos do R

O comando `help(rotina)` ou `?rotina` é utilizado para se obter uma ajuda rápida sobre determinada rotina. A documentação do R pode ser aberta no navegador padrão através do comando `help.start()`. Para procurar em quais rotinas aparece determinado assunto, deve-se digitar `help.search("assunto")`.

¹É importante salientar que as notações deste trabalho e de Paulino e Singer (2006) diferem um pouco das utilizadas por Poletto (2006) e Poletto, Singer e Paulino (2007a, 2007b).

O comando `args(rotina)` retorna os argumentos que a rotina pode utilizar. Comumente alguns deles são opcionais. Ao utilizar o comando `objeto<-rotina(arg1=a,arg2=b)`, aplica-se a `rotina` com o argumento `arg1` assumindo o valor `a`, o argumento `arg2` assumindo o valor `b` e armazena-se o resultado em `objeto`. O nome do objeto deve iniciar com uma letra e pode conter letras, números e ponto final em suas outras posições. Neste texto utiliza-se a convenção de armazenar os objetos do exemplo `x.y` em `exy.cont`, em que `cont` é uma abreviação indicativa do conteúdo do objeto.

Um objeto pode conter apenas um escalar, uma matriz, arranjos (*arrays*) ou até uma lista de objetos. Rotinas que realizam análises estatísticas, *e.g.*, modelos lineares generalizados, mistos ou de análise de sobrevivência, em geral, armazenam uma lista de objetos como resultado. Nestes casos, ao digitar `objeto` ou `print(objeto)` os resultados são apresentados de forma reduzida e o comando `summary(objeto)`, geralmente, apresenta uma quantidade maior de informações. O comando `names(objeto)` retorna a lista dos objetos armazenados. Supondo-se que estejam armazenados `obj1` e `obj2`, para visualizar `obj2` basta digitar `objeto$obj2`.

O R ignora tudo o que estiver à direita do sinal `#`, interpretando o conteúdo como um comentário. Considerando que `>` é a indicação que o R está pronto para receber os comandos, ilustra-se a utilização de algumas rotinas com os respectivos comentários contendo as explicações:

```
> x<-2 #armazenando um escalar
> x
[1] 2
> x<-c(1,9,-15,8) #constrói um vetor
> x
[1] 1 9 -15 8
> x<-c(-3,x,6) #amplia o vetor anterior
> x
[1] -3 1 9 -15 8 6
> c(-3,x,6) #se não armazenar num objeto, o resultado já é apresentado
[1] -3 -3 1 9 -15 8 6 6
> 1:6 #seqüência de números
[1] 1 2 3 4 5 6
> 6:1 #ao contrário
[1] 6 5 4 3 2 1
> seq(0,1,0.1) #a seqüência começa em 0, vai até 1, com espaçamentos de 0.1
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> seq(0.1,1,0.2) #não necessariamente termina em 1
[1] 0.1 0.3 0.5 0.7 0.9
> rep(2,5) #repete o 1º argumento o número de vezes indicado no 2º argumento
[1] 2 2 2 2 2
> rep(c(1,3),4)
[1] 1 3 1 3 1 3 1 3
> c(rep(1,3),rep(0,4))
[1] 1 1 1 0 0 0 0
> rep(c(0,2,5),c(1,3,4)) #pode-se utilizar um vetor em cada argumento
[1] 0 2 2 2 5 5 5 5
> x[4] #extrai o 4º elemento de x
[1] -15
> x[2:4] #extrai do 2º ao 4º elemento de x
[1] 1 9 -15
> x[c(1,3:4)] #extrai os elementos 1, 3 e 4 de x
[1] -3 9 -15
```

```

> x[-3] #reproduz x sem o 3º elemento
[1] -3  1 -15  8  6
> x[-c(1,4)] #reproduz x sem o 1º e o 4º elementos
[1] 1 9 8 6

```

Existem várias maneiras disponíveis para se construir matrizes. As mais simples são por meio dos comandos `rbind()` e `cbind()` que empilham escalares, vetores ou matrizes em, respectivamente, linhas e colunas. Outra possibilidade é utilizar o comando `matrix()` que recebe um vetor e pelo menos uma das dimensões da matriz. Veja os exemplos:

```

> x1<-rbind(1:3,c(1,3,-1))
> x1
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    3   -1
> x2<-cbind(c(1,3),c(0,4))
> x2
      [,1] [,2]
[1,]    1    0
[2,]    3    4
> cbind(x2,x1)
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    1    2    3
[2,]    3    4    1    3   -1
> matrix(1:6,nrow=3) #este comando utiliza o argumento byrow com F (False) como padrão...
      [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> matrix(1:6,ncol=3) #que faz com que o vetor seja preenchido por coluna na matriz
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> matrix(1:6,ncol=3,byrow=T) #utilizando a opção T (True), preenche-se por linha
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> x1[2,] #extraí a 2ª linha de x1
[1] 1 3 -1
> x1[,2] #extraí a 2ª coluna de x1
[1] 2 3
> x1[,2:3] #extraí uma submatriz de x1
      [,1] [,2]
[1,]    2    3
[2,]    3   -1
> x1[,-1] #outra maneira
      [,1] [,2]
[1,]    2    3
[2,]    3   -1
> x1[,c(1,3)] #outra submatriz
      [,1] [,2]
[1,]    1    3
[2,]    1   -1

```

Os sinais +, -, * e / representam as operações básicas de adição, subtração, multiplicação e divisão.

Ao utilizar estes sinais entre vetores ou matrizes, a operação é realizada elemento a elemento. O mesmo acontece com logaritmos neperianos e exponenciais, representadas pelas rotinas `log()` e `exp()`, respectivamente. A operação `A%*%B` realiza a multiplicação matricial, `t(A)` transpõe, `solve(A)` inverte e `A%x%B` ou `kron(A,B)` realiza o produto de Kronecker (à direita). Ao receber como argumento uma matriz quadrada, a rotina `diag()` extrai sua diagonal principal. Se, no entanto, receber como argumento um vetor, a rotina criará uma matriz quadrada com os elementos do vetor na diagonal principal. Por fim, se a rotina receber como argumento um número `n`, uma matriz identidade de ordem `n` é retornada.

```
> x1<-matrix(1:9,nrow=3)
> x2<-matrix(9:1,nrow=3)
> t(x1)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> x1
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> x2
      [,1] [,2] [,3]
[1,]    9    6    3
[2,]    8    5    2
[3,]    7    4    1
> x1*x2
      [,1] [,2] [,3]
[1,]    9   24   21
[2,]   16   25   16
[3,]   21   24    9
> x1%*%x2
      [,1] [,2] [,3]
[1,]   90   54   18
[2,]  114   69   24
[3,]  138   84   30
> solve(x1*x2)
      [,1]      [,2]      [,3]
[1,] -0.7361111  1.3333333 -0.6527778
[2,]  0.8888889 -1.6666667  0.8888889
[3,] -0.6527778  1.3333333 -0.7361111
> diag(x1)
[1] 1 5 9
> diag(c(1,3,4))
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    3    0
[3,]    0    0    4
> diag(2.5) #números quebrados são truncados
      [,1] [,2]
[1,]    1    0
[2,]    0    1
> diag(2) %x% x1
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    4    7    0    0    0
```

```

[2,] 2 5 8 0 0 0
[3,] 3 6 9 0 0 0
[4,] 0 0 0 1 4 7
[5,] 0 0 0 2 5 8
[6,] 0 0 0 3 6 9
> x1 %x% diag(2)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 0 4 0 7 0
[2,] 0 1 0 4 0 7
[3,] 2 0 5 0 8 0
[4,] 0 2 0 5 0 8
[5,] 3 0 6 0 9 0
[6,] 0 3 0 6 0 9

```

3 Instalação da biblioteca de rotinas Catdata

Deve-se importar as rotinas para o R. Pode-se fazer isso diretamente da internet com o comando `source("http://www.poletto.com/Catdata.r")` ou gravá-la no próprio computador e, supondo que foram armazenadas dentro do diretório `C:\Dir1\Dir2`, utilizar o comando `source("C:/Dir1/Dir2/Catdata.r")`. Note que as “barras invertidas” do caminho de localização do arquivo são substituídas pelas “barras de data”. Após realizar esta operação, é recomendável sair do R clicando-se na opção “Sim” (“Yes”) com relação à pergunta “Salvar imagem da área de trabalho?” (“Save workspace image?”). Como isto fará com que todos os objetos sejam armazenados, sugere-se que isto seja feito antes de começar a rodar outros comandos. Desta forma, das próximas vezes que o R for inicializado, as rotinas já estarão carregadas e prontas para utilização, não sendo mais necessário repetir o procedimento — a não ser que as rotinas sejam removidas com, *e.g.*, `rm(list=ls())`.

4 Utilização das rotinas para a análise de dados completamente categorizados

A leitura dos dados é realizada por meio da rotina `readCatdata()` (de *read categorical data*). O único argumento utilizado para a análise de dados sem omissão é o TF (de *table of frequencies*), em que deve-se especificar a tabela de freqüências. TF deve receber um vetor, representando uma única população (supondo-se uma distribuição multinomial), ou uma matriz, com cada linha da matriz representando uma subpopulação (assumindo-se uma distribuição produto de multinomiais, em que cada linha representa uma das multinomiais), conforme a Tabela 1.5 (p.9), mas sem incluir os totais das linhas e colunas. Para o Exemplo 1.5 (p.5), considerando que cada combinação dos níveis de sexo e faixa etária são provenientes de uma subpopulação e para cada subpopulação são registradas 4 categorias de respostas, resultantes das combinações dos níveis de freqüência e habilidade, deve-se utilizar:

```

> e15.TF<-rbind(c(19, 5, 4, 2),
+              c( 5, 8, 0,17),
+              c(11, 6, 7, 6),
+              c( 2, 5, 1,22))
> e15.catdata<-readCatdata(TF=e15.TF)
> e15.catdata #Apresenta as proporções e erros padrões

```

```
Call: readCatdata(TF = e15.TF)
```

```
S=4 subpopulations x R=4 response categories with COMPLETE data
```

```
Proportions:
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.6333	0.1667	0.1333	0.0667
[2,]	0.1667	0.2667	0.0000	0.5667
[3,]	0.3667	0.2000	0.2333	0.2000
[4,]	0.0667	0.1667	0.0333	0.7333

```
Standard errors of the proportions:
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0880	0.0680	0.0621	0.0455
[2,]	0.0680	0.0807	0.0000	0.0905
[3,]	0.0880	0.0730	0.0772	0.0730
[4,]	0.0455	0.0680	0.0328	0.0807

```
> print(e15.catdata,digits=3) #Pode-se escolher o n° de decimais do arredondamento
```

```
Call: readCatdata(TF = e15.TF)
```

```
S=4 subpopulations x R=4 response categories with COMPLETE data
```

```
Proportions:
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.633	0.167	0.133	0.067
[2,]	0.167	0.267	0.000	0.567
[3,]	0.367	0.200	0.233	0.200
[4,]	0.067	0.167	0.033	0.733

```
Standard errors of the proportions:
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.088	0.068	0.062	0.046
[2,]	0.068	0.081	0.000	0.090
[3,]	0.088	0.073	0.077	0.073
[4,]	0.046	0.068	0.033	0.081

Os modelos lineares podem ser ajustados por MV com a rotina `linML()` (de *linear models e maximum likelihood*). O argumento `obj` deve indicar um objeto resultante da rotina `readCatdata`. Optando-se pela formulação em equações livres [termos de restrições] deve-se especificar em `X [U]` a matriz `X [C]` de (3.18) [(3.19)], p.51. Caso não seja informada a matriz `A` no argumento `A`, a rotina seleciona as primeiras $r - 1$ componentes de cada multinomial como padrão por meio de $\mathbf{A} = \mathbf{I}_s \otimes [\mathbf{I}_{r-1}, \mathbf{0}_{r-1}]$. O Exemplo 8.1 (p.228) apresenta alguns resultados do ajuste por MV do modelo descrito no Exemplo 3.1 (p.47), que se reproduz a seguir ilustrando o uso da rotina.

```
> e31.TF<-c(192,1,5,2,146,5,11,12,71)
> e31.catdata<-readCatdata(TF=e31.TF)
> e81.U<-rbind(c(0,-1, 0,1,0, 0,0,0),
+             c(0, 0,-1,0,0, 0,1,0),
+             c(0, 0, 0,0,0,-1,0,1))
> e81.X<-rbind(c(1,0,0,0,0),
```



```

+           c(0,1,0,0,0),
+           c(0,0,1,0,0),
+           c(0,1,0,0,0),
+           c(0,0,0,1,0),
+           c(0,0,0,0,1),
+           c(0,0,1,0,0),
+           c(0,0,0,0,1)
> e81.linml1<-linML(obj=e31.catdata,U=e81.U) #formulação em termos de restrições: simetria
> e81.linml2<-linML(obj=e31.catdata,X=e81.X) #formulação em equações livres: simetria
> e81.linml1 #as estatísticas de ajuste das duas formulações coincidem, conforme o esperado...

```

```
Call: linML(obj = e31.catdata, U = e81.U)
```

```
Goodness of fit of the linear model (d.f.=3):
```

	statistic	p-value
Likelihood ratio	5.6156	0.1319
Pearson	5.4657	0.1407
Neyman	6.4640	0.0911
Wald	5.5337	0.1366

```
> e81.linml2 #mas nesta formulação há informação sobre os parâmetros do modelo
```

```
Call: linML(obj = e31.catdata, X = e81.X)
```

```
Maximum likelihood estimates of the parameters of the linear model:
```

	estimate	std.error	z-value	p-value
[1,]	0.4315	0.0235	18.3768	0.0000
[2,]	0.0034	0.0019	1.7379	0.0822
[3,]	0.0180	0.0044	4.0739	0.0000
[4,]	0.3281	0.0223	14.7408	0.0000
[5,]	0.0191	0.0045	4.2042	0.0000

```
Goodness of fit of the linear model (d.f.=3):
```

	statistic	p-value
Likelihood ratio	5.6156	0.1319
Pearson	5.4657	0.1407
Neyman	6.4640	0.0911
Wald	5.5337	0.1366

```
> summary(e81.linml1) #a saída do summary apresenta mais informações
```

```
Call: linML(obj = e31.catdata, U = e81.U)
```

```
Maximum likelihood estimates of the probabilities under the linear model (LM):
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.4315	0.0034	0.0180	0.0034	0.3281	0.0191	0.0180	0.0191	0.1596

```
Standard errors:
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0235	0.0019	0.0044	0.0019	0.0223	0.0045	0.0044	0.0045	0.0174

```
Maximum likelihood estimates of the linear functions specified by the matrix A:
```

	observed	std.error	under the LM	std.error
[1,]	0.4315	0.0235	0.4315	0.0235

[2,]	0.0022	0.0022	0.0034	0.0019
[3,]	0.0112	0.0050	0.0180	0.0044
[4,]	0.0045	0.0032	0.0034	0.0019
[5,]	0.3281	0.0223	0.3281	0.0223
[6,]	0.0112	0.0050	0.0191	0.0045
[7,]	0.0247	0.0074	0.0180	0.0044
[8,]	0.0270	0.0077	0.0191	0.0045

Fisher scoring attained the convergence criterion in 2 iterations.

Goodness of fit of the linear model (d.f.=3):

	statistic	p-value
Likelihood ratio	5.6156	0.1319
Pearson	5.4657	0.1407
Neyman	6.4640	0.0911
Wald	5.5337	0.1366

Estimated frequencies under the linear model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	192.0	1.5	8.0	1.5	146.0	8.5	8.0	8.5	71.0

> summary(e81.linml2)

Call: linML(obj = e31.catdata, X = e81.X)

Maximum likelihood estimates of the probabilities under the linear model (LM):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.4315	0.0034	0.0180	0.0034	0.3281	0.0191	0.0180	0.0191	0.1596

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0235	0.0019	0.0044	0.0019	0.0223	0.0045	0.0044	0.0045	0.0174

Maximum likelihood estimates of the linear functions specified by the matrix A:

	observed	std.error	under the LM	std.error
[1,]	0.4315	0.0235	0.4315	0.0235
[2,]	0.0022	0.0022	0.0034	0.0019
[3,]	0.0112	0.0050	0.0180	0.0044
[4,]	0.0045	0.0032	0.0034	0.0019
[5,]	0.3281	0.0223	0.3281	0.0223
[6,]	0.0112	0.0050	0.0191	0.0045
[7,]	0.0247	0.0074	0.0180	0.0044
[8,]	0.0270	0.0077	0.0191	0.0045

Maximum likelihood estimates of the parameters of the linear model:

	estimate	std.error	z-value	p-value
[1,]	0.4315	0.0235	18.3768	0.0000
[2,]	0.0034	0.0019	1.7379	0.0822
[3,]	0.0180	0.0044	4.0739	0.0000
[4,]	0.3281	0.0223	14.7408	0.0000

[5,] 0.0191 0.0045 4.2042 0.0000

Fisher scoring attained the convergence criterion in 2 iterations.

Goodness of fit of the linear model (d.f.=3):

	statistic	p-value
Likelihood ratio	5.6156	0.1319
Pearson	5.4657	0.1407
Neyman	6.4640	0.0911
Wald	5.5337	0.1366

Estimated frequencies under the linear model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	192.0	1.5	8.0	1.5	146.0	8.5	8.0	8.5	71.0

A rotina `loglin.cd()` (de *log-linear models e maximum likelihood*) ajusta modelos log-lineares ordinários na formulação (5.8) ou (5.10) por MV (pp.114-115), em que \mathbf{A} deve ser informado em `A` e \mathbf{X} em `X` ou \mathbf{X}_G em `XL`. A rotina também ajusta modelos log-lineares generalizados (6.16) por MV (p.159), desde que a matriz \mathbf{A} seja ortogonal às colunas da matriz $\mathbf{I}_s \otimes \mathbf{1}_r$ definidora das restrições naturais, *i.e.*, $\mathbf{A}(\mathbf{I}_s \otimes \mathbf{1}_r) = \mathbf{0}_{u,s}$. Pode-se utilizar também as formulações em termos de restrições

$$\begin{aligned} \mathbf{U} \ln(\boldsymbol{\pi}) &= \mathbf{0}_{s(r-1)-p}, \\ \mathbf{U}_G \mathbf{A} \ln(\boldsymbol{\pi}) &= \mathbf{0}_{u-p}, \end{aligned}$$

equivalentes às formulações em equações livres, em que a matriz \mathbf{U} (\mathbf{U}_G) de dimensão $[s\{r-1\}-p] \times sr$ ($[u-p] \times u$) define as $s[r-1]-p$ ($u-p$) restrições, possui posto máximo e suas linhas são ortogonais às colunas de $[\mathbf{I}_s \otimes \mathbf{1}_r, \mathbf{X}]$ (\mathbf{X}_G), ou seja, $\mathbf{U}[\mathbf{I}_s \otimes \mathbf{1}_r, \mathbf{X}] = \mathbf{0}_{(sr-p),p}$ ($\mathbf{U}_G \mathbf{X}_G = \mathbf{0}_{(u-p),p}$). Neste caso, deve-se informar \mathbf{U} em `U` ou \mathbf{U}_G em `UL`. Caso não seja informada a matriz \mathbf{A} em `A`, a rotina utiliza os logitos de referência (com relação à categoria r) como padrão por meio de $\mathbf{A} = \mathbf{I}_s \otimes [\mathbf{I}_{r-1}, -\mathbf{1}_{r-1}]$. O Exemplo 9.1 (p.263) é utilizado para ilustrar o uso da rotina.

```
> e91.catdata<-readCatdata(TF=c(3,25,32,68))
> e91.X<-rbind(c(0,0),
+             c(0,1),
+             c(1,0),
+             c(1,1))
> e91.loglinml1<-loglinML(obj=e91.catdata,U=c(1,-1,-1,1)) #formul.em termos de restrs.: indep.
> e91.loglinml2<-loglinML(obj=e91.catdata,X=e91.X) #formulação em equações livres: independ.
> e91.loglinml1
```

Call: `loglinML(obj = e91.catdata, U = c(1, -1, -1, 1))`

Goodness of fit of the log-linear model (d.f.=1):

	statistic	p-value
Likelihood ratio	5.7400	0.0166
Pearson	4.9888	0.0255
Neyman	9.0905	0.0026

Wald 4.4535 0.0348

> e91.loglinml2

Call: loglinML(obj = e91.catdata, X = e91.X)

Maximum likelihood estimates of the parameters of the log-linear model:

	estimate	std.error	z-value	p-value
[1,]	1.2730	0.2138	5.9538	0.0000
[2,]	0.9773	0.1983	4.9281	0.0000

Goodness of fit of the log-linear model (d.f.=1):

	statistic	p-value
Likelihood ratio	5.7400	0.0166
Pearson	4.9888	0.0255
Neyman	9.0905	0.0026
Wald	4.4535	0.0348

> summary(e91.loglinml2)

Call: loglinML(obj = e91.catdata, X = e91.X)

Maximum likelihood estimates of the probabilities under the log-linear model (LLM):

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0598	0.1589	0.2136	0.5676

Standard errors:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0132	0.0279	0.0324	0.0406

Maximum likelihood estimates of the log-linear functions:

	observed	std.error	under the LLM	std.error
[1,]	-3.1209	0.5899	-2.2502	0.2916
[2,]	-1.0006	0.2339	-1.2730	0.2138
[3,]	-0.7538	0.2144	-0.9773	0.1983

Maximum likelihood estimates of the parameters of the log-linear model:

	estimate	std.error	z-value	p-value
[1,]	1.2730	0.2138	5.9538	0.0000
[2,]	0.9773	0.1983	4.9281	0.0000

Fisher scoring attained the convergence criterion in 4 iterations.

Goodness of fit of the log-linear model (d.f.=1):

	statistic	p-value
Likelihood ratio	5.7400	0.0166
Pearson	4.9888	0.0255
Neyman	9.0905	0.0026
Wald	4.4535	0.0348

Estimated frequencies under log-linear model:

```
      [,1] [,2] [,3] [,4]
[1,]  7.656 20.344 27.344 72.656
```

```
> e91.X2<-rbind(c(0,0,0),
+              c(0,1,0),
+              c(1,0,0),
+              c(1,1,1))
> e91.loglinml3<-loglinML(obj=e91.catdata,X=e91.X2) #modelo log-linear (LL) ordinário saturado
> e91.loglinml4<-loglinML(obj=e91.catdata,A=c(1,-1,-1,1),XL=1) #modelo LL generalizado saturado
> e91.loglinml3
```

Call: loglinML(obj = e91.catdata, X = e91.X2)

Maximum likelihood estimates of the parameters of the log-linear model:

	estimate	std.error	z-value	p-value
[1,]	2.3671	0.6038	3.9203	0.0001
[2,]	2.1203	0.6110	3.4701	0.0005
[3,]	-1.3665	0.6475	-2.1103	0.0348

Goodness of fit of the log-linear model (d.f.=0):

	statistic	p-value
Likelihood ratio	0	1
Pearson	0	1
Neyman	0	1
Wald	0	1

```
> e91.loglinml4
```

Call: loglinML(obj = e91.catdata, A = c(1, -1, -1, 1), XL = 1)

Maximum likelihood estimates of the parameters of the log-linear model:

	estimate	std.error	z-value	p-value
[1,]	-1.3665	0.6475	-2.1103	0.0348

Goodness of fit of the log-linear model (d.f.=0):

	statistic	p-value
Likelihood ratio	0	1
Pearson	0	1
Neyman	0	1
Wald	0	1

Outros argumentos, opcionais, de `linML()` e `loglinML()` são:

- **start** - vetor com valores iniciais para os parâmetros do modelo (como padrão, utiliza-se as estimativas de MQG). Só pode ser utilizado se $u = s(r - 1)$;
- **maxit** - número máximo de iterações (o padrão é 100);
- **epsilon1** e **epsilon2** - critérios de convergência do processo iterativo. Considera-se que o processo iterativo convergiu quando (1) a diferença absoluta das estatísticas de razão de verossimilhanças de ajuste do modelo de iterações consecutivas for menor que **epsilon1** e (2) as diferenças absolutas de

estimativas de iterações consecutivas de todos os parâmetros forem menores que `epsilon2`. Ambas têm o valor padrão 10^{-6} (`1e-6`);

- `zeroN` - valor que irá substituir uma frequência nula no denominador da estatística de Neyman (7.15), p.199, do ajuste do modelo (o padrão é que se utilize $[rN_q]^{-1}$ para a q -ésima subpopulação, $q = 1, \dots, s$);
- `trace` - com o valor 0 (padrão) nada é feito, com o valor 1 a rotina mostra o valor da estatística de razão de verossimilhanças de ajuste do modelo em cada iteração e com o valor 2, além da estatística, a rotina também apresenta as estimativas dos parâmetros obtidas em cada iteração.

Os principais objetos armazenados pela saída da rotina `linML()` e `loglinML()`, que podem ser utilizados posteriormente em outros cálculos, são:

- `thetaH` (`VthetaH`) - estimativas das probabilidades (covariâncias respectivas) sob o modelo;
- `beta` (`Vbeta`) - estimativas dos parâmetros (covariâncias respectivas) do modelo, caso se utilize a formulação em equações livres;
- `Fu` (`VFu`) - funções lineares de `linML()` ou log-lineares de `loglinML()` observadas (covariâncias respectivas), *i.e.*, sem impor as restrições do modelo;
- `FH` (`VFH`) - funções lineares de `linML()` ou log-lineares de `loglinML()` observadas (covariâncias respectivas) sob o modelo;
- `QvH`, `QpH`, `QnH` e `QwH` - estatísticas de razão de verossimilhanças, Pearson, Neyman e Wald de ajuste do modelo;
- `ystH` - frequências estimadas pelo modelo.

Como exemplo, note alguns cálculos que são realizados para o Exemplo 9.1 (p.263).

```
> qnorm(0.975) #retorna o quantil 97.5% da distribuição normal padrão
[1] 1.959964
> e91.loglinml4$beta+c(-1,1)*qnorm(0.975)*sqrt(e91.loglinml4$Vbeta) #IC(95%) para o parâmetro
[1] -2.63561831 -0.09736516
> round(e91.loglinml4$beta+c(-1,1)*qnorm(0.975)*sqrt(e91.loglinml4$Vbeta),3) #arred.3 dígitos
[1] -2.636 -0.097
> round(exp(e91.loglinml4$beta),3) #estimativa de MV da razão de chances (RR)
[1] 0.255
> round(exp(e91.loglinml4$beta+c(-1,1)*qnorm(0.975)*sqrt(e91.loglinml4$Vbeta)),3) #IC(95%) p/RR
[1] 0.072 0.907
```

Os modelos funcionais lineares podem ser ajustados por MQG com a rotina `funlinWLS()` (de *functional linear models e weighted least squares*). As funções paramétricas de interesse $\mathbf{F}(\boldsymbol{\pi})$ devem ser formadas pela composição das 4 funções básicas:

1. linear $\mathbf{F}_{\text{lin}, \mathbf{A}_1}(\boldsymbol{\pi}) = \mathbf{A}_1 \boldsymbol{\pi}$;
2. logarítmica $\mathbf{F}_{\text{log}}(\boldsymbol{\pi}) = \ln(\boldsymbol{\pi})$;

3. exponencial $\mathbf{F}_{\text{exp}}(\boldsymbol{\pi}) = \mathbf{exp}(\boldsymbol{\pi})$;
4. adição de constantes $\mathbf{F}_{\text{add},\pi_1}(\boldsymbol{\pi}) = \pi_1 + \boldsymbol{\pi}$.

A especificação das funções de interesse deve ser feita no argumento `model`, que é um vetor contendo as indicações das funções na ordem que são escritas matematicamente. O número máximo de funções lineares e adição de constantes é 9. Para cada função linear deve-se especificar uma matriz em \mathbf{A}_i , com i variando de 1 até no máximo 9, iniciando a numeração na função linear mais interna, ou seja, na ordem em que as operações são realizadas. Da mesma forma, para cada adição de constantes deve-se especificar um vetor PI_i . Alguns exemplos de funções são apresentados na Tabela 1.

Tabela 1: Exemplos de funções

Função	model	Argumentos que devem ser informados
$\mathbf{A}_1\boldsymbol{\pi}$	"lin"	\mathbf{A}_1
$\ln(\boldsymbol{\pi})$	"log"	
$\mathbf{exp}(\boldsymbol{\pi})$	"exp"	
$\pi_1 + \boldsymbol{\pi}$	"add"	PI_1
$\mathbf{A}_1 \ln(\boldsymbol{\pi})$	c("lin", "log")	\mathbf{A}_1
$\mathbf{exp}[\mathbf{A}_1 \ln(\boldsymbol{\pi})]$	c("exp", "lin", "log")	\mathbf{A}_1
$\pi_3 + \mathbf{exp}[\pi_2 + \mathbf{A}_1 \ln(\pi_1 + \boldsymbol{\pi})]$	c("add", "exp", "add", "lin", "log", "add")	$\mathbf{A}_1, \text{PI}_1, \text{PI}_2, \text{PI}_3$
$\pi_1 + \mathbf{exp}[\mathbf{A}_4 \ln\{\mathbf{A}_3 \mathbf{exp}[\mathbf{A}_2 \ln(\mathbf{A}_1\boldsymbol{\pi})]\}]$	c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin")	$\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4, \text{PI}_1$

Quando o modelo é linear, `model="lin"` (log-linear, `model=c("lin", "log")`), e não se especificar o argumento \mathbf{A}_1 , a rotina utiliza $\mathbf{A}_1 = \mathbf{I}_s \otimes [\mathbf{I}_{r-1}, \mathbf{0}_{r-1}]$ ($\mathbf{A}_1 = \mathbf{I}_s \otimes [\mathbf{I}_{r-1}, -\mathbf{1}_{r-1}]$) como padrão, que também é o argumento padrão da matriz \mathbf{A} , argumento \mathbf{A} , da rotina que estima o modelo linear (log-linear) por máxima verossimilhança, `linML` (`loglinML`).

Ao utilizar a formulação em equações livres (11.1) [termos de restrições (11.7)] deve-se especificar a matriz \mathbf{X} [\mathbf{U}] em \mathbf{X} [\mathbf{U}] (pp.372-373). A única exceção ocorre quando se utiliza o modelo log-linear (`model=c("lin", "log")`) com $\mathbf{A}_1 (\mathbf{I}_s \otimes \mathbf{1}_r) = \mathbf{0}_{u,s}$. Neste caso, os argumentos para qualquer formulação devem obedecer a mesma lógica da rotina `loglinML()`, podendo-se especificar \mathbf{X} , \mathbf{U} , \mathbf{XL} ou \mathbf{UL} .

O único argumento opcional de `funlinWLS()` é o `zeroN`, valor que substitui frequências nulas para se obter a matriz de covariâncias das proporções em (11.29), p.378. Como padrão a rotina substitui frequências nulas da q -ésima subpopulação com $[rN_q]^{-1}$, $q = 1, \dots, s$. A saída da rotina contém os mesmos objetos das rotinas `linML()` e `loglinML()`, com exceção de `QvH`, `QpH` e `QnH`.

O Exemplo 11.2 (p.382) apresenta o ajuste por MQG do Exemplo 3.1 (p.47), que descrevemos a seguir. Confronte os resultados com o ajuste por MV do Exemplo 8.1 (p.228).

```
> e112.linwls1<-funlinWLS(model="lin",obj=e31.catdata,U=e81.U) #form.em termos de restrs.: sim.
```

```
> e112.linwls2<-funlinWLS(model="lin",obj=e31.catdata,X=e81.X) #form.em eqs. livres: simetria
> e112.linwls1
```

```
Call: funlinWLS(model = "lin", obj = e31.catdata, U = e81.U)
```

```
Wald goodness of fit statistic of the model (d.f.=3): 5.5337 (p-value=0.1366)
```

```
> e112.linwls2
```

```
Call: funlinWLS(model = "lin", obj = e31.catdata, X = e81.X)
```

```
Weighted least squares estimates of the parameters of the model:
```

	estimate	std.error	z-value	p-value
[1,]	0.4368	0.0234	18.6938	0.0000
[2,]	0.0030	0.0018	1.6583	0.0972
[3,]	0.0156	0.0041	3.8143	0.0001
[4,]	0.3322	0.0222	14.9696	0.0000
[5,]	0.0161	0.0042	3.8667	0.0001

```
Wald goodness of fit statistic of the model (d.f.=3): 5.5337 (p-value=0.1366)
```

```
> summary(e112.linwls1)
```

```
Call: funlinWLS(model = "lin", obj = e31.catdata, U = e81.U)
```

```
Weighted least squares estimates of the probabilities under the model:
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.4368	0.0030	0.0156	0.0030	0.3322	0.0161	0.0156	0.0161	0.1615

```
Standard errors:
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0234	0.0018	0.0041	0.0018	0.0222	0.0042	0.0041	0.0042	0.0173

```
Weighted least squares estimates of the functions:
```

	observed	std.error	under the model	std.error
[1,]	0.4315	0.0235	0.4368	0.0234
[2,]	0.0022	0.0022	0.0030	0.0018
[3,]	0.0112	0.0050	0.0156	0.0041
[4,]	0.0045	0.0032	0.0030	0.0018
[5,]	0.3281	0.0223	0.3322	0.0222
[6,]	0.0112	0.0050	0.0161	0.0042
[7,]	0.0247	0.0074	0.0156	0.0041
[8,]	0.0270	0.0077	0.0161	0.0042

```
Wald goodness of fit statistic of the model (d.f.=3): 5.5337 (p-value=0.1366)
```

```
Estimated frequencies under the model:
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	194.388	1.350	6.960	1.350	147.815	7.147	6.960	7.147	71.883

Reajusta-se, a seguir, o Exemplo 9.1 (p.263) por MQG. Compare com os resultados anteriores.


```

> e91.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e91.catdata,
+ U=c(1,-1,-1,1)) #formulação em termos de restrições: independência
> e91.loglinwls2<-funlinWLS(model=c("lin","log"),obj=e91.catdata,
+ X=e91.X) #formulação em equações livres: independência
> e91.loglinwls1

Call: funlinWLS(model = c("lin", "log"), obj = e91.catdata, U = c(1, -1, -1, 1))

Wald goodness of fit statistic of the model (d.f.=1): 4.4535 (p-value=0.0348)

> e91.loglinwls2

Call: funlinWLS(model = c("lin", "log"), obj = e91.catdata, X = e91.X)

Weighted least squares estimates of the parameters of the model:
      estimate std.error z-value p-value
[1,]  1.1789   0.2181   5.4054  0.0000
[2,]  0.9035   0.2023   4.4667  0.0000

Wald goodness of fit statistic of the model (d.f.=1): 4.4535 (p-value=0.0348)

> summary(e91.loglinwls2)

Call: funlinWLS(model = c("lin", "log"), obj = e91.catdata, X = e91.X)

Weighted least squares estimates of the probabilities under the model:
      [,1] [,2] [,3] [,4]
[1,] 0.0678 0.1674 0.2205 0.5443

Standard errors:
      [,1] [,2] [,3] [,4]
[1,] 0.0163 0.0277 0.0315 0.0462

Weighted least squares estimates of the functions:
      observed std.error under the model std.error
[1,] -3.1209   0.5899   -2.0825      0.3254
[2,] -1.0006   0.2339   -1.1789      0.2181
[3,] -0.7538   0.2144   -0.9035      0.2023

Weighted least squares estimates of the parameters of the model:
      estimate std.error z-value p-value
[1,]  1.1789   0.2181   5.4054  0.0000
[2,]  0.9035   0.2023   4.4667  0.0000

Wald goodness of fit statistic of the model (d.f.=1): 4.4535 (p-value=0.0348)

Estimated frequencies under the model:
      [,1] [,2] [,3] [,4]
[1,]  8.682 21.430 28.224 69.665

> e91.loglinwls3<-funlinWLS(model=c("lin","log"),obj=e91.catdata,

```

```

+ X=e91.X2) #modelo log-linear ordinário saturado
> e91.loglinwls4<-funlinWLS(model=c("lin","log"),obj=e91.catdata,A1=c(1,-1,-1,1),
+ XL=1) #modelo log-linear generalizado saturado
> e91.loglinwls3

```

```
Call: funlinWLS(model = c("lin", "log"), obj = e91.catdata, X = e91.X2)
```

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	2.3671	0.6038	3.9203	0.0001
[2,]	2.1203	0.6110	3.4701	0.0005
[3,]	-1.3665	0.6475	-2.1103	0.0348

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

```
> e91.loglinwls4
```

```
Call: funlinWLS(model = c("lin", "log"), obj = e91.catdata, A1 = c(1, -1, -1, 1), XL = 1)
```

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	-1.3665	0.6475	-2.1103	0.0348

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

No Exemplo 6.8 (p.169) apresenta-se a medida de concordância *kappa* de Cohen escrita como um modelo funcional linear para os dados do Exemplo 1.2 (p.4). O Exemplo 11.12 (p.399) ajusta este modelo por MQG. A seguir, apresenta-se os cálculos, mas ressalta-se que as matrizes \mathbf{A}_1 , \mathbf{A}_2 e \mathbf{A}_3 utilizadas são diferentes das do livro, pois optou-se por escrever o modelo funcional linear de uma outra maneira.

```

> e1112.TF<-c(11,5,0,14,34,7,2,13,11)
> e1112.catdata<-readCatdata(TF=e1112.TF)
> e1112.A1<-rbind(
+ c(rep(c(1,0,0,0),2),1),
+ rep(1,9),
+ kronecker(diag(3),t(rep(1,3))),
+ kronecker(t(rep(1,3)),diag(3)) )
> e1112.A2<-rbind(
+ cbind(diag(2),matrix(0,2,6)),
+ cbind(matrix(0,3,2),kronecker(t(rep(1,2)),diag(3))) )
> e1112.A3<-cbind( c(1,0),c(1,1),-c(2,1)%*%t(rep(1,3)) )
> e1112.A4<-t(c(1,-1))
> e1112.A1
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]  1   0   0   0   1   0   0   0   1
[2,]  1   1   1   1   1   1   1   1   1
[3,]  1   1   1   0   0   0   0   0   0
[4,]  0   0   0   1   1   1   0   0   0
[5,]  0   0   0   0   0   0   1   1   1
[6,]  1   0   0   1   0   0   1   0   0
[7,]  0   1   0   0   1   0   0   1   0
[8,]  0   0   1   0   0   1   0   0   1
> e1112.A2
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]

```

```

[1,] 1 0 0 0 0 0 0 0
[2,] 0 1 0 0 0 0 0 0
[3,] 0 0 1 0 0 1 0 0
[4,] 0 0 0 1 0 0 1 0
[5,] 0 0 0 0 1 0 0 1
> e1112.A3
      [,1] [,2] [,3] [,4] [,5]
[1,] 1 1 -2 -2 -2
[2,] 0 1 -1 -1 -1
> e1112.A4
      [,1] [,2]
[1,] 1 -1
> e1112.kappawls<-funlinWLS(model=c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"),
+ obj=e1112.catdata,A1=e1112.A1,A2=e1112.A2,A3=e1112.A3,A4=e1112.A4,PI1=-1,X=1)
> e1112.kappawls

Call: funlinWLS(model = c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"), ...)

Weighted least squares estimates of the parameters of the model:
      estimate std.error z-value p-value
[1,] 0.2960 0.0827 3.5806 0.0003

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

> round(pnorm((e1112.kappawls$beta-0.35)/sqrt(e1112.kappawls$Vbeta)),3)
      [,1]
[1,] 0.257
> round(e1112.kappawls$beta+c(-1,1)*qnorm(0.975)*sqrt(e1112.kappawls$Vbeta),3)
[1] 0.134 0.458

```

Ao utilizar a formulação em equações livres, por meio da rotina `waldTest()` pode-se realizar um teste de Wald de uma hipótese

$$H : C\beta = C_0,$$

em que a matriz de contrastes C tem posto máximo igual ao seu número de linhas, o seu número de colunas é igual à dimensão de β e C_0 é um vetor de constantes conhecidas com dimensão igual ao número de linhas de C . Deve-se informar em `obj` algum objeto criado pela rotina `linML()`, `loglinML()` ou `funlinWLS()`, em `C` a matriz C e em `C0` a matriz C_0 . O argumento `C0` é opcional e, quando não for informado, a rotina utiliza como padrão $C_0 = \mathbf{0}$, que costuma ser a opção de maior interesse. Ilustra-se o uso desta rotina com o Exemplo 11.7 (p.390).

```

> e117.TF<-rbind(c(28,40,68),
+               c( 5,21,49),
+               c( 1, 4,15))
> e117.catdata<-readCatdata(TF=e117.TF)
> e117.A<-kronecker(diag(3),cbind(diag(2),rep(-1,2)))
> e117.X1<-rbind(c(1,0,0,0),
+               c(0,1,0,0),
+               c(1,0,2,0),
+               c(0,1,1,0),
+               c(1,0,0,2),
+               c(0,1,0,1))
> e117.loglinwls<-funlinWLS(model=c("lin", "log"),obj=e117.catdata,A1=e117.A,

```

```

+ XL=e117.X1) #efeito de linha
> e117.loglinwls

Call: funlinWLS(model = c("lin", "log"), obj = e117.catdata, A1 = e117.A, XL = e117.X1)

Weighted least squares estimates of the parameters of the model:
      estimate  std.error  z-value  p-value
[1,]  -0.9161    0.2225   -4.1163  0.0000
[2,]  -0.4421    0.1767   -2.5021  0.0123
[3,]  -0.5617    0.2228   -2.5205  0.0117
[4,]  -0.8886    0.4192   -2.1200  0.0340

Wald goodness of fit statistic of the model (d.f.=2): 1.0343 (p-value=0.5962)

> cbind(0*diag(2),diag(2))
      [,1] [,2] [,3] [,4]
[1,]    0    0    1    0
[2,]    0    0    0    1
> waldTest(obj=e117.loglinwls,C=cbind(0*diag(2),diag(2)))

Call: waldTest(obj = e117.loglinwls, C = cbind(0 * diag(2), diag(2)))

Wald statistic of the hypothesis (d.f.=2): 9.4324 (p-value=0.0089)

```

5 Utilização das rotinas para a análise de dados categorizados com omissão

A leitura dos dados categorizados incompletos, *i.e.*, com omissão, também é realizada por meio da rotina `readCatdata()`. Contudo, deve-se informar adicionalmente os argumentos Z_p e R_p , além de especificar TF de uma maneira diferente. Considere uma extensão à notação empregada no Capítulo 13 (p.450) para denotar as s subpopulações analogamente aos capítulos anteriores, adicionando o índice q ($= 1, \dots, s$) à Z_t , T , l , m_t e n_{tj} . Deve-se informar no argumento Z_p uma matriz com os indicadores das categorias de resposta $\{Z_{qt}\}$ de cada um dos padrões de omissão de cada subpopulação, com exceção dos padrões de classificação completa ($t = 1$), *i.e.*, Z_p deve receber

$$[(Z_{q2}, \dots, Z_{qT_q}), q = 1, \dots, s],$$

exatamente nessa ordem. Note que Z_p é uma matriz de dimensão $m \times \left(\sum_{q=1}^s l_q\right)$. A rotina irá recuperar cada uma das submatrizes Z_{qt} , $q = 1, \dots, s$, $t = 2, \dots, T_q$, utilizando o argumento R_p , que deve conter

$$m_{q2}, \dots, m_{qT_q}$$

em cada uma das suas $q = 1, \dots, s$ linhas. TF deve receber

$$n_{q11}, \dots, n_{q1m}, n_{q21}, \dots, n_{q2m_2}, \dots, n_{qT_q1}, \dots, n_{qT_qm_{T_q}}$$

em cada linha. Quando os padrões de omissão forem diferentes entre as subpopulações, os vetores supramencionados de R_p e TF não poderão ser combinados em matrizes. Nesses casos, deve-se preencher

os vetores com elementos iguais a zero até a mínima dimensão necessária ser atingida para preservar os demais vetores. Uma ilustração desse artifício é apresentada em Poletto *et al.* (2007b). Exibe-se a entrada de dados do Exemplo 13.4 (p.472; veja as matrizes $\{Z_{qt}\}$ no Exemplo 13.1, p.454) e Exemplo 13.2 (p.466).

```
> e134.TF<-c(12,4,5,2, 50,31, 27,12)
> e134.Zp<-cbind(kronecker(diag(2),rep(1,2)),kronecker(rep(1,2),diag(2)))
> e134.Rp<-c(2,2)
> e134.catdata<-readCatdata(TF=e134.TF,Zp=e134.Zp,Rp=e134.Rp)
> e134.catdata
```

```
Call: readCatdata(TF = e134.TF, Zp = e134.Zp, Rp = e134.Rp)
```

```
S=1 subpopulations x R=4 response categories with MISSING data
```

```
Proportions of the complete data:
```

```
      [,1] [,2] [,3] [,4]
[1,] 0.5217 0.1739 0.2174 0.0870
```

```
Standard errors of the proportions of the complete data:
```

```
      [,1] [,2] [,3] [,4]
[1,] 0.1042 0.0790 0.0860 0.0588
```

```
> summary(e134.catdata)
```

```
Call: readCatdata(TF = e134.TF, Zp = e134.Zp, Rp = e134.Rp)
```

```
S=1 subpopulations x R=4 response categories with MISSING data
```

```
Table of frequencies of the complete data:
```

```
[1] 12  4  5  2
```

```
Proportions of the complete data:
```

```
      [,1] [,2] [,3] [,4]
[1,] 0.5217 0.1739 0.2174 0.0870
```

```
Standard errors of the proportions of the complete data:
```

```
      [,1] [,2] [,3] [,4]
[1,] 0.1042 0.0790 0.0860 0.0588
```

```
Missing data frequencies and associated column vectors indicating
the relation with the original set of R response categories:
```

```
      [,1] [,2] [,3] [,4]
[1,] 50    1    1    0    0
[2,] 31    0    0    1    1
```

```
      [,1] [,2] [,3] [,4]
[1,] 27    1    0    1    0
[2,] 12    0    1    0    1
```

```
> e132.TF<-c(7,11,2,3,9,5,0,10,4, 8,7,3,0, 0,7,14,7)
> e132.Zp<-cbind(rbind( cbind(kronecker(rep(1,2),diag(3)),rep(0,6)),
+                      cbind(matrix(0,3,3),rep(1,3)) ),
```

```

+           rbind( cbind(rep(1,3),matrix(0,3,3)),
+                 cbind(rep(0,6),kronecker(rep(1,2),diag(3))) ) )
> e132.Rp<-c(4,4)
> e132.catdata<-readCatdata(TF=e132.TF,Zp=e132.Zp,Rp=e132.Rp)
> summary(e132.catdata)

```

Call: readCatdata(TF = e132.TF, Zp = e132.Zp, Rp = e132.Rp)

S=1 subpopulations x R=9 response categories with MISSING data

Table of frequencies of the complete data:

```
[1] 7 11 2 3 9 5 0 10 4
```

Proportions of the complete data:

```

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,] 0.1373 0.2157 0.0392 0.0588 0.1765 0.0980 0.0000 0.1961 0.0784

```

Standard errors of the proportions of the complete data:

```

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,] 0.0482 0.0576 0.0272 0.0329 0.0534 0.0416 0.0000 0.0556 0.0376

```

Missing data frequencies and associated column vectors indicating the relation with the original set of R response categories:

```

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,] 8      1      0      0      1      0      0      0      0      0
[2,] 7      0      1      0      0      1      0      0      0      0
[3,] 3      0      0      1      0      0      1      0      0      0
[4,] 0      0      0      0      0      0      0      1      1      1

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,] 0      1      1      1      0      0      0      0      0      0
[2,] 7      0      0      0      1      0      0      1      0      0
[3,] 14     0      0      0      0      1      0      0      1      0
[4,] 7      0      0      0      0      0      1      0      0      1

```

Os resultados de `print()` e `summary()` retornam as proporções (p_{ij}) e os respectivos erros padrões para o padrão de classificação completa ($t = 1$), enquanto o último também apresenta quais categorias de resposta estão associadas a cada uma das frequências, que é útil para verificar se os padrões de omissão foram corretamente especificados. Note que, no Exemplo 13.4, o cenário $t = 4$ foi omitido, pois as rotinas realizam análises apenas sob os mecanismos MAR e MCAR e cenários de omissão total, *i.e.*, $\mathbf{Z}_{qt} = \mathbf{1}_m$, não trazem informação para a estimação de $\boldsymbol{\theta}$, conforme pode-se verificar em (13.6), p.452.

A análise por MV de modelos estruturais saturados para $\boldsymbol{\theta}$, conforme apresentada na Seção 13.3 (p.456), pode ser realizada com a rotina `satMarML()` informando um objeto `readCatdata` como primeiro argumento, ou especificando o argumento `catdataobj`. Os outros argumentos são opcionais. O argumento `missing` permite que se escolha entre os erros padrões sob o mecanismo "MAR" (padrão) e "MCAR", que são calculados a partir da estimativa da matriz de informação de Fisher.

Conforme mencionado no item (c) do Exercício 13.5 (p.482), a estimativa da matriz de informação de Fisher relativa a θ sob o mecanismo MAR é igual à estimativa da matriz de informação observada relativa a θ sob os mecanismos MAR e MCAR. Portanto, efetivamente há 3 processos iterativos diferenciados para a obtenção das EMV $\hat{\theta}$ de θ que podem ser informados no argumento `method`: (a) "EM", (b) "FS-MCAR" (*scoring* de Fisher sob MCAR) e (c) "NR/FS-MAR" (*scoring* de Fisher sob MAR ou Newton-Raphson sob MAR ou MCAR). Como o EMV de θ é o mesmo sob os mecanismos MAR e MCAR, pode-se utilizar o processo iterativo (b) mesmo que se suponha o mecanismo MAR, desde que após a obtenção de $\hat{\theta}$ se utilize a estimativa da matriz de covariâncias assintótica sob o mecanismo MAR. `method="EM"` é a opção padrão, porque é mais estável que os demais, embora o número máximo de iterações (`maxit=100`) possa não ser suficiente para atingir a convergência. Em alguns casos, quando há zeros amostrais e `method="FS-MCAR"` ou, principalmente, `method="NR/FS-MAR"`, pode ser necessário relaxar o critério de convergência, pois o processo iterativo pode facilmente saltar para um estimativa negativa e/ou gerar uma matriz de covariância singular. Os argumentos `epsilon1` e `epsilon2` podem ser utilizados para alterar o critério de convergência da mesma forma como explicitados na seção anterior.

```
> e134.satmcarml<-satMarML(e134.catdata,missing="MCAR")
> e134.satmarml<-satMarML(e134.catdata,method="FS-MCAR")
> e134.satmarml2<-satMarML(e134.catdata,method="NR/FS-MAR")
> e134.satmcarml

Call: satMarML(catdataobj = e134.catdata, missing = "MCAR")

S=1 subpopulations x R=4 response categories

Maximum likelihood estimates of the probabilities:
      [,1] [,2] [,3] [,4]
[1,] 0.4603 0.1740 0.2484 0.1173

Standard errors (MCAR):
      [,1] [,2] [,3] [,4]
[1,] 0.0670 0.0597 0.0607 0.0533

Goodness of fit statistics of MCAR given MAR assumption (d.f.=2)
              statistic p-value
Likelihood ratio    0.6229  0.7324
Pearson             0.6100  0.7371
Neyman              0.6602  0.7188

> summary(e134.satmcarml)#compare as estimativas das probabilidades e

Call: satMarML(catdataobj = e134.catdata, missing = "MCAR")

S=1 subpopulations x R=4 response categories

Maximum likelihood estimates of the probabilities:
      [,1] [,2] [,3] [,4]
[1,] 0.4603 0.1740 0.2484 0.1173

Standard errors (MCAR):
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0670	0.0597	0.0607	0.0533

EM attained the convergence criterion in 43 iterations.

Goodness of fit statistics of MCAR given MAR assumption (d.f.=2):

	statistic	p-value
Likelihood ratio	0.6229	0.7324
Pearson	0.6100	0.7371
Neyman	0.6602	0.7188

Augmented estimated frequencies under MCAR:

	[,1]	[,2]	[,3]	[,4]
[1,]	10.586	4.003	5.713	2.699
[2,]	37.282	14.096	20.119	9.504
[3,]	17.951	6.787	9.687	4.576

> summary(e134.satmarm1) #frequências ampliadas, erros padrões e

Call: satMarML(catdataobj = e134.catdata, method = "FS-MCAR")

S=1 subpopulations x R=4 response categories

Maximum likelihood estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.4603	0.1740	0.2484	0.1173

Standard errors (MAR):

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0666	0.0603	0.0638	0.0580

FS-MCAR attained the convergence criterion in 7 iterations.

Goodness of fit statistics of MCAR given MAR assumption (d.f.=2):

	statistic	p-value
Likelihood ratio	0.6229	0.7324
Pearson	0.6100	0.7371
Neyman	0.6602	0.7188

Augmented estimated frequencies under MAR:

	[,1]	[,2]	[,3]	[,4]
[1,]	12.000	4.000	5.000	2.000
[2,]	36.282	13.718	21.054	9.946
[3,]	17.537	7.167	9.463	4.833

> summary(e134.satmarm12)#os números de iterações

Call: satMarML(catdataobj = e134.catdata, method = "NR/FS-MAR")

S=1 subpopulations x R=4 response categories

Maximum likelihood estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.4603	0.1740	0.2484	0.1173

Standard errors (MAR):

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0666	0.0603	0.0638	0.0580

NR/FS-MAR attained the convergence criterion in 4 iterations.

Goodness of fit statistics of MCAR given MAR assumption (d.f.=2):

	statistic	p-value
Likelihood ratio	0.6229	0.7324
Pearson	0.6100	0.7371
Neyman	0.6602	0.7188

Augmented estimated frequencies under MAR:

	[,1]	[,2]	[,3]	[,4]
[1,]	12.000	4.000	5.000	2.000
[2,]	36.282	13.718	21.054	9.946
[3,]	17.537	7.167	9.463	4.833

```
> e132.satmarm1<-satMarML(e132.catdata,method="NR/FS-MAR")
```

```
Erro em satMarML(e132.catdata, method = "NR/FS-MAR") :
```

```
Any of the estimated probabilities obtained by the iterative process are outside the parameter space.
```

```
Try another iterative process or starting values. (iteration 1)
```

```
> e132.satmarm1<-satMarML(e132.catdata)
```

```
> e132.satmarm1<-satMarML(e132.catdata,method="FS-MCAR",missing="MCAR")
```

```
> e132.TF2<-c(7,11,2,3,9,5,1e-5,10,4, 8,7,3,0, 0,7,14,7) #subst.zero por valor peq.
```

```
> e132.catdata2<-readCatdata(TF=e132.TF2,Zp=e132.Zp,Rp=e132.Rp)
```

```
> e132.satmarm12<-satMarML(e132.catdata2,method="NR/FS-MAR")
```

```
> summary(e132.satmarm1)#compare todos os resultados
```

```
Call: satMarML(catdataobj = e132.catdata, missing = "MCAR", method = "FS-MCAR")
```

S=1 subpopulations x R=9 response categories

Maximum likelihood estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1061	0.1418	0.0260	0.1516	0.2188	0.1241	0.0000	0.1652	0.0664

Standard errors (MCAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0387	0.0431	0.0215	0.0384	0.0496	0.0372	0.0000	0.0447	0.0318

FS-MCAR attained the convergence criterion in 13 iterations.

Goodness of fit statistics of MCAR given MAR assumption (d.f.=6):

	statistic	p-value
Likelihood ratio	35.9325	0.0000
Pearson	24.4088	0.0004
Neyman	7854.0934	0.0000

Augmented estimated frequencies under MCAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	5.4124	7.2305	1.3244	7.7320	11.1599	6.3315	0.0000	8.4240	3.3854
[2,]	1.9103	2.5519	0.4674	2.7289	3.9388	2.2346	0.0000	2.9732	1.1948
[3,]	2.9715	3.9697	0.7271	4.2450	6.1270	3.4761	0.0000	4.6249	1.8586

```
> summary(e132.satmarm1)
```

Call: satMarML(catdataobj = e132.catdata)

S=1 subpopulations x R=9 response categories

Maximum likelihood estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1061	0.1418	0.0260	0.1516	0.2188	0.1241	0.0000	0.1652	0.0664

Standard errors (MAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0359	0.0389	0.0179	0.0654	0.0528	0.0388	0.0795	0.0455	0.0303

EM attained the convergence criterion in 14 iterations.

Goodness of fit statistics of MCAR given MAR assumption (d.f.=6):

	statistic	p-value
Likelihood ratio	35.9325	0.0000
Pearson	24.4088	0.0004
Neyman	7854.1061	0.0000

Augmented estimated frequencies under MAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	7.000	11.000	2.000	3.000	9.000	5.000	0.000	10.000	4.000
[2,]	3.294	2.752	0.519	4.706	4.248	2.481	0.000	0.000	0.000
[3,]	0.000	0.000	0.000	7.000	7.978	4.561	0.000	6.022	2.439

```
> summary(e132.satmarm12)#avalie o efeito da substituição pelo valor pequeno
```

Call: satMarML(catdataobj = e132.catdata2, method = "NR/FS-MAR")

S=1 subpopulations x R=9 response categories

Maximum likelihood estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1061	0.1418	0.0260	0.1516	0.2188	0.1241	0.0000	0.1652	0.0664

Standard errors (MAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0358	0.0385	0.0179	0.0404	0.0520	0.0384	0.0001	0.0450	0.0302

NR/FS-MAR attained the convergence criterion in 6 iterations.

Goodness of fit statistics of MCAR given MAR assumption (d.f.=6):

	statistic	p-value
Likelihood ratio	35.9325	0.0000
Pearson	24.4088	0.0004
Neyman	7854.0962	0.0000

Augmented estimated frequencies under MAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	7.000	11.000	2.000	3.000	9.000	5.000	0.000	10.000	4.000
[2,]	3.294	2.752	0.519	4.706	4.248	2.481	0.000	0.000	0.000
[3,]	0.000	0.000	0.000	7.000	7.978	4.561	0.000	6.022	2.439

A análise por MQG em uma fase de modelos saturados para θ sob o mecanismo MCAR (Seção 13.5.1, p.463) pode ser realizada com a função `satMcarWLS()`. Note que os zeros amostrais também podem causar problemas na estimação por MQG.

```
> e134.satmcarwls<-satMcarWLS(e134.catdata)
> e134.satmcarwls
```

```
Call: satMcarWLS(catdataobj = e134.catdata)
```

```
S=1 subpopulations x R=4 response categories
```

Weighted least squares estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.4586	0.1773	0.2514	0.1128

Standard errors:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0674	0.0592	0.0577	0.0484

Neyman goodness of fit statistic of MCAR (d.f.=2): 0.651 (p-value=0.7221)

```
> summary(e134.satmcarwls) #compare com a abordagem por MV
```

```
Call: satMcarWLS(catdataobj = e134.catdata)
```

```
S=1 subpopulations x R=4 response categories
```

Weighted least squares estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.4586	0.1773	0.2514	0.1128

Standard errors:

	[,1]	[,2]	[,3]	[,4]
[1,]	0.0674	0.0592	0.0577	0.0484

Neyman goodness of fit statistic of MCAR (d.f.=2): 0.651 (p-value=0.7221)

Augmented estimated frequencies under MCAR:

	[,1]	[,2]	[,3]	[,4]
[1,]	10.548	4.078	5.781	2.593
[2,]	37.146	14.361	20.360	9.133
[3,]	17.885	6.914	9.803	4.397

```
> e132.satmcarwls<-satMcarWLS(e132.catdata)
```

Warning message:

Any of the estimated probabilities are outside the parameter space.

```
in: satMcarWLS(e132.catdata)
```

```
> e132.satmcarwls2<-satMcarWLS(e132.catdata2) #mesmo com a subst.do zero por valor peq.
```

Warning message:

Any of the estimated probabilities are outside the parameter space.

```
in: satMcarWLS(e132.catdata2)
```

```
> summary(e132.satmcarwls2) #estimativas negativas
```

```
Call: satMcarWLS(catdataobj = e132.catdata2)
```

S=1 subpopulations x R=9 response categories

Weighted least squares estimates of the probabilities:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0269	-0.0175	-0.0075	0.1977	0.5191	0.2748	0.0000	0.0062	0.0003

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0394	0.0410	0.0253	0.0281	0.0357	0.0319	0.0001	0.0312	0.0309

Neyman goodness of fit statistic of MCAR (d.f.=6): 105.8264 (p-value=0)

Augmented estimated frequencies under MCAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	1.3699	-0.8950	-0.3845	10.0839	26.4741	14.0160	0.0000	0.3181	0.0174
[2,]	0.4835	-0.3159	-0.1357	3.5590	9.3438	4.9468	0.0000	0.1123	0.0062
[3,]	0.7521	-0.4914	-0.2111	5.5363	14.5348	7.6951	0.0000	0.1747	0.0096

```
> e132.satmarm1$alphast #Tabela 13.3 - EMV das prob.condicionais de omissão
$st1.1
```

```
[1] 0.6799999 0.7998732 0.7939726 0.2040000 0.4240139 0.4152062 0.0000000 0.6241379 0.6212325
```

```
$st1.2
```

```
[1] 0.3200000 0.2001264 0.2060272 0.0000000
```

```
$st1.3
```

```
[1] 0.0000000 0.4760000 0.3758609 0.3787669
```

As rotinas `linML()` e `loglinML()` permitem a análise de modelos lineares e log-lineares por MV da mesma forma como foi apresentado na seção anterior para dados completos, bastando informar no argumento `obj` um objeto resultante da rotina `satMarML()`. As rotinas herdam a suposição do mecanismo de omissão (MAR ou MCAR) de `satMarML()`. Note algumas diferenças dos resultados apresentados quando há dados com omissão.

```
> e132.U<-rbind(c(0, 1,1,-1,0,0,-1, 0),
+              c(0,-1,0, 1,0,1, 0,-1) )
> e132.linml<-linML(e132.satmarm12,U=e132.U) #homog.marg.: formul.restrições
> e132.A<-rbind(c(1,1,1,0,0,0,0,0,0),
+              c(0,0,0,1,1,1,0,0,0),
+              c(1,0,0,1,0,0,1,0,0),
+              c(0,1,0,0,1,0,0,1,0) )
> e132.X<-rbind(c(1,0),
+              c(0,1),
+              c(1,0),
+              c(0,1) )
> e132.linml2<-linML(e132.satmarm12,A=e132.A,X=e132.X) #homog.marg.: formul.eq.s.livres
> e132.linml
```

```
Call: linML(obj = e132.satmarm12, U = e132.U)
```

```
Goodness of fit of the linear model given MAR (d.f.=2):
```

	statistic	p-value
Likelihood ratio	0.1287	0.9377
Pearson	0.1288	0.9376
Neyman	0.1286	0.9377
Wald	0.1285	0.9378

```
Goodness of fit of the linear model and MCAR given MAR (d.f.=8):
```

	statistic	p-value
Likelihood ratio	36.0612	0.0000
Pearson	24.7743	0.0017
Neyman	7327.0080	0.0000

```
> summary(e132.linml)
```

```
Call: linML(obj = e132.satmarm12, U = e132.U)
```

```
Maximum likelihood estimates of the probabilities under the linear model (LM):
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1048	0.1346	0.0256	0.1601	0.2225	0.1308	0.0000	0.1564	0.0653

```
Standard errors (MAR):
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
--	------	------	------	------	------	------	------	------	------

```
[1,] 0.0354 0.0274 0.0166 0.0266 0.0507 0.0295 0.0001 0.0275 0.0298
```

Maximum likelihood estimates of the linear functions specified by the matrix A:

	observed	std.error	under the LM	std.error
[1,]	0.1061	0.0358	0.1048	0.0354
[2,]	0.1418	0.0385	0.1346	0.0274
[3,]	0.0260	0.0179	0.0256	0.0166
[4,]	0.1516	0.0404	0.1601	0.0266
[5,]	0.2188	0.0520	0.2225	0.0507
[6,]	0.1241	0.0384	0.1308	0.0295
[7,]	0.0000	0.0001	0.0000	0.0001
[8,]	0.1652	0.0450	0.1564	0.0275

Fisher scoring attained the convergence criterion in 3 iterations.

Goodness of fit of the linear model given MAR (d.f.=2):

	statistic	p-value
Likelihood ratio	0.1287	0.9377
Pearson	0.1288	0.9376
Neyman	0.1286	0.9377
Wald	0.1285	0.9378

Goodness of fit of the linear model and MCAR given MAR (d.f.=8):

	statistic	p-value
Likelihood ratio	36.0612	0.0000
Pearson	24.7743	0.0017
Neyman	7327.0080	0.0000

Augmented estimated frequencies under the linear model and MAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	6.9119	10.4407	1.9680	3.1684	9.1518	5.2690	0.0000	9.4674	3.9321
[2,]	3.2527	2.6122	0.5107	4.9701	4.3195	2.6145	0.0000	0.0000	0.0000
[3,]	0.0000	0.0000	0.0000	7.3930	8.1125	4.8066	0.0000	5.7013	2.3974

```
> summary(e132.linml2)
```

```
Call: linML(obj = e132.satmarm12, A = e132.A, X = e132.X)
```

Maximum likelihood estimates of the probabilities under the linear model (LM):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1048	0.1346	0.0256	0.1601	0.2225	0.1308	0.0000	0.1564	0.0653

Standard errors (MAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0354	0.0274	0.0166	0.0266	0.0507	0.0295	0.0001	0.0275	0.0298

Maximum likelihood estimates of the linear functions specified by the matrix A:

	observed	std.error	under the LM	std.error
[1,]	0.2739	0.0514	0.2649	0.0361

[2,]	0.4946	0.0647	0.5135	0.0372
[3,]	0.2577	0.0444	0.2649	0.0361
[4,]	0.5258	0.0507	0.5135	0.0372

Maximum likelihood estimates of the parameters of the linear model under MAR:

	estimate	std.error	z-value	p-value
[1,]	0.2649	0.0361	7.3332	0.0000
[2,]	0.5135	0.0372	13.7931	0.0000

Fisher scoring attained the convergence criterion in 3 iterations.

Goodness of fit of the linear model given MAR (d.f.=2):

	statistic	p-value
Likelihood ratio	0.1287	0.9377
Pearson	0.1288	0.9376
Neyman	0.1286	0.9377
Wald	0.1285	0.9378

Goodness of fit of the linear model and MCAR given MAR (d.f.=8):

	statistic	p-value
Likelihood ratio	36.0612	0.0000
Pearson	24.7743	0.0017
Neyman	7327.0080	0.0000

Augmented estimated frequencies under the linear model and MAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	6.9119	10.4407	1.9680	3.1684	9.1518	5.2690	0.0000	9.4674	3.9321
[2,]	3.2527	2.6122	0.5107	4.9701	4.3195	2.6145	0.0000	0.0000	0.0000
[3,]	0.0000	0.0000	0.0000	7.3930	8.1125	4.8066	0.0000	5.7013	2.3974

```

> e133.TF<-c(77,87,94,70,67,36,143,78, 14,8,3,9, 25,18,43,16, 14,12)
> e133.Zp<-cbind(kronecker(diag(4),rep(1,2)),
+               kronecker(diag(2),kronecker(rep(1,2),diag(2))),
+               kronecker(diag(2),rep(1,4)) )
> e133.Rp<-c(4,4,2)
> e133.catdata<-readCatdata(TF=e133.TF,Zp=e133.Zp,Rp=e133.Rp)
> e133.satmcarml<-satMarML(e133.catdata,missing="MCAR")
> e133.satmarmml<-satMarML(e133.catdata)
> e133.X<-rbind(c( 1, 1, 1, 1, 1, 1),
+              c( 1, 1,-1, 1,-1,-1),
+              c( 1,-1, 1, -1, 1,-1),
+              c( 1,-1,-1, -1,-1, 1),
+              c(-1, 1, 1, -1,-1, 1),
+              c(-1, 1,-1, -1, 1,-1),
+              c(-1,-1, 1, 1,-1,-1),
+              c(-1,-1,-1, 1, 1, 1))
> e133.U<-c(1,-1,-1,1, -1,1,1,-1)
> e133.loglinml<-loglinML(obj=e133.satmcarml,X=e133.X)
> e133.loglinml2<-loglinML(obj=e133.satmcarml,U=e133.U)
> e133.loglinmlmar<-loglinML(obj=e133.satmarmml,X=e133.X)
> e133.loglinml #Tabela 13.6

```

Call: loglinML(obj = e133.satmcarml, X = e133.X)

Maximum likelihood estimates of the parameters of the log-linear model under MCAR:

	estimate	std.error	z-value	p-value
[1,]	0.0600	0.0377	1.5941	0.1109
[2,]	-0.1760	0.0404	-4.3610	0.0000
[3,]	0.1850	0.0381	4.8550	0.0000
[4,]	0.1956	0.0400	4.8895	0.0000
[5,]	-0.1281	0.0383	-3.3464	0.0008
[6,]	-0.0563	0.0415	-1.3577	0.1746

Goodness of fit of the log-linear model given MCAR (d.f.=1):

	statistic	p-value
Likelihood ratio	1.6643	0.1970
Pearson	1.6398	0.2004
Neyman	1.6636	0.1971
Wald	1.6353	0.2010

Goodness of fit of the log-linear model and MCAR given MAR (d.f.=8):

	statistic	p-value
Likelihood ratio	11.2237	0.1893
Pearson	11.5595	0.1720
Neyman	11.2607	0.1874

> summary(e133.loglinml2) #Tabela 13.7

Call: loglinML(obj = e133.satmcarml, U = e133.U)

Maximum likelihood estimates of the probabilities under the log-linear model (LLM):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.1271	0.1270	0.1368	0.1091	0.0985	0.0589	0.2318	0.1107

Standard errors (MCAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.0117	0.0116	0.0120	0.0108	0.0103	0.0077	0.0148	0.0107

Maximum likelihood estimates of the log-linear functions:

	observed	std.error	under the LLM	std.error
[1,]	0.0382	0.1524	0.1381	0.1314
[2,]	0.1315	0.1483	0.1369	0.1523
[3,]	0.2027	0.1459	0.2115	0.1496
[4,]	-0.1199	0.1590	-0.0149	0.1349
[5,]	-0.1083	0.1590	-0.1169	0.1638
[6,]	-0.7840	0.2032	-0.6306	0.1578
[7,]	0.6613	0.1346	0.7389	0.1219

Fisher scoring attained the convergence criterion in 4 iterations.

Goodness of fit of the log-linear model given MCAR (d.f.=1):

	statistic	p-value
Likelihood ratio	1.6643	0.1970

Pearson	1.6398	0.2004
Neyman	1.6636	0.1971
Wald	1.6353	0.2010

Goodness of fit of the log-linear model and MCAR given MAR (d.f.=8):

	statistic	p-value
Likelihood ratio	11.2237	0.1893
Pearson	11.5595	0.1720
Neyman	11.2607	0.1874

Augmented estimated frequencies under log-linear model and MCAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	82.888	82.786	89.200	71.126	64.228	38.425	151.153	72.194
[2,]	4.322	4.317	4.652	3.709	3.349	2.004	7.882	3.765
[3,]	12.967	12.951	13.955	11.127	10.048	6.011	23.647	11.294
[4,]	3.305	3.301	3.557	2.836	2.561	1.532	6.028	2.879

> summary(e133.loglinml) #Tabela 13.6/13.7

Call: loglinML(obj = e133.satmcarml, X = e133.X)

Maximum likelihood estimates of the probabilities under the log-linear model (LLM):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.1271	0.1270	0.1368	0.1091	0.0985	0.0589	0.2318	0.1107

Standard errors (MCAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.0117	0.0116	0.0120	0.0108	0.0103	0.0077	0.0148	0.0107

Maximum likelihood estimates of the log-linear functions:

	observed	std.error	under the LLM	std.error
[1,]	0.0382	0.1524	0.1381	0.1314
[2,]	0.1315	0.1483	0.1369	0.1523
[3,]	0.2027	0.1459	0.2115	0.1496
[4,]	-0.1199	0.1590	-0.0149	0.1349
[5,]	-0.1083	0.1590	-0.1169	0.1638
[6,]	-0.7840	0.2032	-0.6306	0.1578
[7,]	0.6613	0.1346	0.7389	0.1219

Maximum likelihood estimates of the parameters of the log-linear model under MCAR:

	estimate	std.error	z-value	p-value
[1,]	0.0600	0.0377	1.5941	0.1109
[2,]	-0.1760	0.0404	-4.3610	0.0000
[3,]	0.1850	0.0381	4.8550	0.0000
[4,]	0.1956	0.0400	4.8895	0.0000
[5,]	-0.1281	0.0383	-3.3464	0.0008
[6,]	-0.0563	0.0415	-1.3577	0.1746

Fisher scoring attained the convergence criterion in 3 iterations.

Goodness of fit of the log-linear model given MCAR (d.f.=1):

	statistic	p-value
Likelihood ratio	1.6643	0.1970
Pearson	1.6398	0.2004
Neyman	1.6636	0.1971
Wald	1.6353	0.2010

Goodness of fit of the log-linear model and MCAR given MAR (d.f.=8):

	statistic	p-value
Likelihood ratio	11.2237	0.1893
Pearson	11.5595	0.1720
Neyman	11.2607	0.1874

Augmented estimated frequencies under log-linear model and MCAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	82.888	82.786	89.200	71.126	64.228	38.425	151.153	72.194
[2,]	4.322	4.317	4.652	3.709	3.349	2.004	7.882	3.765
[3,]	12.967	12.951	13.955	11.127	10.048	6.011	23.647	11.294
[4,]	3.305	3.301	3.557	2.836	2.561	1.532	6.028	2.879

> summary(e133.loglinmlmar)

Call: loglinML(obj = e133.satmarm1, X = e133.X)

Maximum likelihood estimates of the probabilities under the log-linear model (LLM):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.1271	0.1270	0.1368	0.1091	0.0985	0.0589	0.2318	0.1107

Standard errors (MAR):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.0117	0.0116	0.0120	0.0107	0.0104	0.0076	0.0149	0.0107

Maximum likelihood estimates of the log-linear functions:

	observed	std.error	under the LLM	std.error
[1,]	0.0382	0.1523	0.1381	0.1312
[2,]	0.1315	0.1478	0.1369	0.1517
[3,]	0.2027	0.1452	0.2115	0.1489
[4,]	-0.1199	0.1576	-0.0149	0.1346
[5,]	-0.1083	0.1594	-0.1169	0.1641
[6,]	-0.7840	0.2004	-0.6306	0.1570
[7,]	0.6613	0.1338	0.7389	0.1211

Maximum likelihood estimates of the parameters of the log-linear model under MAR:

	estimate	std.error	z-value	p-value
[1,]	0.0600	0.0377	1.5941	0.1109
[2,]	-0.1760	0.0402	-4.3730	0.0000
[3,]	0.1850	0.0381	4.8511	0.0000
[4,]	0.1956	0.0400	4.8894	0.0000
[5,]	-0.1281	0.0382	-3.3546	0.0008
[6,]	-0.0563	0.0412	-1.3649	0.1723

Fisher scoring attained the convergence criterion in 2 iterations.

Goodness of fit of the log-linear model given MAR (d.f.=1):

	statistic	p-value
Likelihood ratio	1.6643	0.1970
Pearson	1.6571	0.1980
Neyman	1.6824	0.1946
Wald	1.6548	0.1983

Goodness of fit of the log-linear model and MCAR given MAR (d.f.=8):

	statistic	p-value
Likelihood ratio	11.2237	0.1893
Pearson	11.5595	0.1720
Neyman	11.2607	0.1874

Augmented estimated frequencies under log-linear model and MAR:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	80.865	83.126	90.117	73.886	63.124	39.881	146.867	74.124
[2,]	7.009	7.000	4.448	3.547	1.873	1.121	6.096	2.912
[3,]	12.050	9.674	12.967	8.312	12.825	5.556	30.182	10.439
[4,]	3.560	3.555	3.831	3.054	2.364	1.414	5.564	2.657

A rotina `funlinWLS()` permite a análise de modelos funcionais lineares por MQG da mesma forma como foi apresentado na seção anterior para dados completos. A metodologia dos MQG em duas fases (Seção 13.5.2, p.464) é empregada quando se informa no argumento `obj` um objeto resultante da rotina `satMcarWLS()`. A abordagem híbrida mencionada no último parágrafo da Seção 13.5.2 (p.466) pode ser realizada especificando um objeto resultante da rotina `satMarML()` no argumento `obj`. Nesse caso, a rotina herda a suposição do mecanismo de omissão (MAR ou MCAR) de `satMarML()`. Outra possibilidade para a abordagem híbrida, principalmente ao se assumir mecanismos MNAR, é maximizar a verossimilhança do modelo probabilístico multinomial ou produto de multinomiais com alguma rotina de otimização não-linear do R (*e.g.*, `optim()`, `nlm()` ou `nlminb()`, ou `.nlmP()` do pacote `geoR`) com um modelo saturado para θ e impondo as restrições do mecanismo de omissão informativo; posteriormente, informar a estimativa de MV $\hat{\theta}$ de θ e de sua correspondente matriz de covariâncias, $\hat{V}_{\hat{\theta}}$, nos argumentos `theta` e `Vtheta`, sem especificar o argumento `obj`.

```
> e132.linwls<-funlinWLS(model="lin",obj=e132.satmarm12,U=e132.U)
> e132.linwls2<-funlinWLS(model="lin",obj=e132.satmarm12,A1=e132.A,X=e132.X)
> e132.linwls

Call: funlinWLS(model = "lin", obj = e132.satmarm12, U = e132.U)

Wald goodness of fit statistic of the model (d.f.=2): 0.1285 (p-value=0.9378)

> summary(e132.linwls)

Call: funlinWLS(model = "lin", obj = e132.satmarm12, U = e132.U)
```

Weighted least squares estimates of the probabilities under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1049	0.1344	0.0256	0.1599	0.2229	0.1307	0.0000	0.1562	0.0654

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0355	0.0276	0.0167	0.0265	0.0506	0.0293	0.0001	0.0275	0.0299

Weighted least squares estimates of the functions:

	observed	std.error	under the model	std.error
[1,]	0.1061	0.0358	0.1049	0.0355
[2,]	0.1418	0.0385	0.1344	0.0276
[3,]	0.0260	0.0179	0.0256	0.0167
[4,]	0.1516	0.0404	0.1599	0.0265
[5,]	0.2188	0.0520	0.2229	0.0506
[6,]	0.1241	0.0384	0.1307	0.0293
[7,]	0.0000	0.0001	0.0000	0.0001
[8,]	0.1652	0.0450	0.1562	0.0275

Wald goodness of fit statistic of the model (d.f.=2): 0.1285 (p-value=0.9378)

Augmented estimated frequencies under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	6.922	10.425	1.969	3.165	9.168	5.263	0.000	9.459	3.938
[2,]	3.257	2.608	0.511	4.965	4.327	2.611	0.000	0.000	0.000
[3,]	0.000	0.000	0.000	7.385	8.127	4.801	0.000	5.696	2.401

> summary(e132.linwls2)

Call: funlinWLS(model = "lin", obj = e132.satmarm12, A1 = e132.A, X = e132.X)

Weighted least squares estimates of the probabilities under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.1049	0.1344	0.0256	0.1599	0.2229	0.1307	0.0000	0.1562	0.0654

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.0355	0.0276	0.0167	0.0265	0.0506	0.0293	0.0001	0.0275	0.0299

Weighted least squares estimates of the functions:

	observed	std.error	under the model	std.error
[1,]	0.2739	0.0514	0.2649	0.0361
[2,]	0.4946	0.0647	0.5135	0.0373
[3,]	0.2577	0.0444	0.2649	0.0361
[4,]	0.5258	0.0507	0.5135	0.0373

Weighted least squares estimates of the parameters of the model:

estimate	std.error	z-value	p-value
----------	-----------	---------	---------

[1,]	0.2649	0.0361	7.3363	0.0000
[2,]	0.5135	0.0373	13.7855	0.0000

Wald goodness of fit statistic of the model (d.f.=2): 0.1285 (p-value=0.9378)

Augmented estimated frequencies under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	6.922	10.425	1.969	3.165	9.168	5.263	0.000	9.459	3.938
[2,]	3.257	2.608	0.511	4.965	4.327	2.611	0.000	0.000	0.000
[3,]	0.000	0.000	0.000	7.385	8.127	4.801	0.000	5.696	2.401

```
> e133.satmcarwls<-satMcarWLS(e133.catdata)
> e133.wlswlsmcar<-funlinWLS(model=c("lin","log"),obj=e133.satmcarwls,X=e133.X)
> e133.mlwlsmcar<-funlinWLS(model=c("lin","log"),obj=e133.satmcarml,X=e133.X)
> e133.mlwlsmar<-funlinWLS(model=c("lin","log"),obj=e133.satmarmml,X=e133.X)
> summary(e133.wlswlsmcar)
```

Call: funlinWLS(model = c("lin", "log"), obj = e133.satmcarwls, X = e133.X)

Weighted least squares estimates of the probabilities under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.1266	0.1256	0.1386	0.1094	0.0976	0.0592	0.2313	0.1116

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.0119	0.0112	0.0118	0.0112	0.0099	0.0081	0.0150	0.0105

Weighted least squares estimates of the functions:

	observed	std.error	under the model	std.error
[1,]	0.0293	0.1515	0.1262	0.1312
[2,]	0.1127	0.1485	0.1182	0.1484
[3,]	0.2061	0.1450	0.2163	0.1448
[4,]	-0.1282	0.1595	-0.0201	0.1353
[5,]	-0.1288	0.1584	-0.1344	0.1583
[6,]	-0.7986	0.2058	-0.6346	0.1610
[7,]	0.6536	0.1343	0.7287	0.1208

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.0601	0.0379	1.5869	0.1125
[2,]	-0.1812	0.0406	-4.4582	0.0000
[3,]	0.1842	0.0382	4.8208	0.0000
[4,]	0.1932	0.0400	4.8292	0.0000
[5,]	-0.1231	0.0385	-3.1958	0.0014
[6,]	-0.0571	0.0421	-1.3582	0.1744

Wald goodness of fit statistic of the model (d.f.=1): 1.6363 (p-value=0.2008)

Augmented estimated frequencies under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	82.567	81.915	90.359	71.331	63.630	38.586	150.832	72.781
[2,]	4.306	4.272	4.712	3.720	3.318	2.012	7.865	3.795
[3,]	12.917	12.815	14.136	11.159	9.954	6.037	23.596	11.386
[4,]	3.293	3.267	3.603	2.845	2.537	1.539	6.015	2.902

```
> summary(e133.mlwlsmar)
```

```
Call: funlinWLS(model = c("lin", "log"), obj = e133.satmcarml, X = e133.X)
```

Weighted least squares estimates of the probabilities under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.1269	0.1270	0.1368	0.1092	0.0988	0.0593	0.2313	0.1107

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.0120	0.0113	0.0117	0.0112	0.0101	0.0081	0.0151	0.0105

Weighted least squares estimates of the functions:

	observed	std.error	under the model	std.error
[1,]	0.0382	0.1524	0.1363	0.1316
[2,]	0.1315	0.1483	0.1376	0.1482
[3,]	0.2027	0.1459	0.2119	0.1457
[4,]	-0.1199	0.1590	-0.0135	0.1355
[5,]	-0.1083	0.1590	-0.1140	0.1589
[6,]	-0.7840	0.2032	-0.6244	0.1604
[7,]	0.6613	0.1346	0.7371	0.1208

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.0592	0.0380	1.5578	0.1193
[2,]	-0.1750	0.0406	-4.3147	0.0000
[3,]	0.1840	0.0383	4.8074	0.0000
[4,]	0.1939	0.0401	4.8300	0.0000
[5,]	-0.1279	0.0385	-3.3230	0.0009
[6,]	-0.0567	0.0418	-1.3572	0.1747

Wald goodness of fit statistic of the model (d.f.=1): 1.6353 (p-value=0.201)

Augmented estimated frequencies under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	82.715	82.822	89.207	71.205	64.395	38.653	150.832	72.172
[2,]	4.313	4.319	4.652	3.713	3.358	2.016	7.865	3.764
[3,]	12.940	12.957	13.956	11.139	10.074	6.047	23.596	11.291
[4,]	3.298	3.303	3.557	2.840	2.568	1.541	6.015	2.878

```
> summary(e133.mlwlsmar)
```

```
Call: funlinWLS(model = c("lin", "log"), obj = e133.satmarm1, X = e133.X)
```

Weighted least squares estimates of the probabilities under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.1271	0.1270	0.1368	0.1091	0.0986	0.0591	0.2316	0.1108

Standard errors:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.0120	0.0113	0.0117	0.0110	0.0101	0.0080	0.0152	0.0104

Weighted least squares estimates of the functions:

	observed	std.error	under the model	std.error
[1,]	0.0382	0.1523	0.1373	0.1314
[2,]	0.1315	0.1478	0.1366	0.1477
[3,]	0.2027	0.1452	0.2111	0.1451
[4,]	-0.1199	0.1576	-0.0156	0.1351
[5,]	-0.1083	0.1594	-0.1164	0.1593
[6,]	-0.7840	0.2004	-0.6278	0.1595
[7,]	0.6613	0.1338	0.7374	0.1200

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.0595	0.0380	1.5663	0.1173
[2,]	-0.1754	0.0404	-4.3376	0.0000
[3,]	0.1845	0.0383	4.8178	0.0000
[4,]	0.1950	0.0402	4.8564	0.0000
[5,]	-0.1277	0.0384	-3.3264	0.0009
[6,]	-0.0565	0.0415	-1.3602	0.1738

Wald goodness of fit statistic of the model (d.f.=1): 1.6548 (p-value=0.1983)

Augmented estimated frequencies under the model:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	80.828	83.136	90.112	73.868	63.185	40.012	146.702	74.155
[2,]	7.005	7.001	4.448	3.546	1.875	1.124	6.090	2.913
[3,]	12.044	9.676	12.967	8.310	12.837	5.574	30.148	10.443
[4,]	3.558	3.556	3.830	3.054	2.366	1.419	5.558	2.659

```
> e132.kA1<-rbind(
+ c(rep(c(1,0,0,0),2),1),
+ rep(1,9),
+ kronecker(diag(3),t(rep(1,3))),
+ kronecker(t(rep(1,3)),diag(3)) )
> e132.kA2<-rbind(
+ cbind(diag(2),matrix(0,2,6)),
+ cbind(matrix(0,3,2),kronecker(t(rep(1,2)),diag(3))) )
> e132.kA3<-cbind( c(1,0),c(1,1),-c(2,1)%*%t(rep(1,3)) )
> e132.kA4<-t(c(1,-1))
> e132.kappa<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
+ obj=e132.satmarm1,A1=e132.kA1,A2=e132.kA2,A3=e132.kA3,A4=e132.kA4,PI1=-1,X=1)
> W1<-c(1,0.75,0,0.75,1,0.75,0,0.75,1) #pesos cuadráticos Fleiss e Cohen (1973)
> W2<-c(1,0.5,0,0.5,1,0.5,0,0.5,1) #pesos absolutos Agresti (2002)
> e132.kw1A1<-rbind(
+ t(W1),
```

```

+ rep(1,9),
+ kronecker(diag(3),t(rep(1,3))),
+ kronecker(t(rep(1,3)),diag(3)) )
> e132.kw2A1<-rbind(
+ t(W2),
+ rep(1,9),
+ kronecker(diag(3),t(rep(1,3))),
+ kronecker(t(rep(1,3)),diag(3)) )
> e132.kwA2<-rbind(
+ cbind(diag(2),matrix(0,2,6)),
+ cbind(matrix(0,9,2),cbind(kronecker(diag(3),rep(1,3)),kronecker(rep(1,3),diag(3))))))
> e132.kw1A3<-cbind( c(1,0),c(1,1),kronecker(-c(2,1),t(W1)) )
> e132.kw2A3<-cbind( c(1,0),c(1,1),kronecker(-c(2,1),t(W2)) )
> e132.kappaw1<-funlinWLS(model=c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"),
+ obj=e132.satmarm1,A1=e132.kw1A1,A2=e132.kwA2,A3=e132.kw1A3,A4=e132.kA4,PI1=-1,X=1)
> e132.kappaw2<-funlinWLS(model=c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"),
+ obj=e132.satmarm1,A1=e132.kw2A1,A2=e132.kwA2,A3=e132.kw2A3,A4=e132.kA4,PI1=-1,X=1)
> e132.kappa

```

```
Call: funlinWLS(model = c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"), ...)
```

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.0171	0.1046	0.1633	0.8703

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

```
> e132.kappaw1
```

```
Call: funlinWLS(model = c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"), ...)
```

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.2967	0.2058	1.4412	0.1495

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

```
> e132.kappaw2
```

```
Call: funlinWLS(model = c("add", "exp", "lin", "log", "lin", "exp", "lin", "log", "lin"), ...)
```

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.1403	0.1119	1.2534	0.2101

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

Considere a fatoração para as probabilidades condicionais de omissão descrita no Exemplo 13.1 (p.455), o mecanismo MNAR_{sat} do Exemplo 13.4, explicitado em (13.10), e a log-verossimilhança apresentada na p.473. A verossimilhança sob este mecanismo MNAR pode ser expressa em função de θ e $\alpha = (\alpha_{10}, \alpha_{20}, \alpha_{30}, \alpha_1, \alpha_2)$ por

$$L(\theta, \alpha) \propto \prod_{i=1}^2 \prod_{j=1}^2 (\theta_{ij} \lambda_{1(ij)})^{n_{1ij}} \times \prod_{i=1}^2 (\theta_{i1} \lambda_{2(i1)} + \theta_{i2} \lambda_{2(i2)})^{n_{2i}} \times$$

$$\begin{aligned}
& \prod_{j=1}^2 (\theta_{1j}\lambda_{3(1j)} + \theta_{2j}\lambda_{3(2j)})^{n_{3j}} \times \left(\sum_{i=1}^2 \sum_{j=1}^2 \theta_{ij}\lambda_{4(ij)} \right)^{N_4} \\
&= \prod_{i=1}^2 \prod_{j=1}^2 (\theta_{ij}\psi_{1(ij)}\psi_{21(ij)})^{n_{1ij}} \times \prod_{i=1}^2 \left(\sum_{j=1}^2 \theta_{ij}\psi_{1(ij)} (1 - \psi_{21(ij)}) \right)^{n_{2i}} \times \\
& \quad \prod_{j=1}^2 \left(\sum_{i=1}^2 \theta_{ij} (1 - \psi_{1(ij)}) \psi_{20(ij)} \right)^{n_{3j}} \times \left(\sum_{i=1}^2 \sum_{j=1}^2 \theta_{ij} (1 - \psi_{1(ij)}) (1 - \psi_{20(ij)}) \right)^{N_4} \\
&= \prod_{i=1}^2 \prod_{j=1}^2 \left(\theta_{ij} \frac{e^{\alpha_{10} + \alpha_1(i-1) + \alpha_2(j-1)}}{1 + e^{\alpha_{10} + \alpha_1(i-1) + \alpha_2(j-1)}} \frac{e^{\alpha_{20} + \alpha_1(i-1) + \alpha_2(j-1)}}{1 + e^{\alpha_{20} + \alpha_1(i-1) + \alpha_2(j-1)}} \right)^{n_{1ij}} \times \\
& \quad \prod_{i=1}^2 \left(\sum_{j=1}^2 \theta_{ij} \frac{e^{\alpha_{10} + \alpha_1(i-1) + \alpha_2(j-1)}}{1 + e^{\alpha_{10} + \alpha_1(i-1) + \alpha_2(j-1)}} \frac{1}{1 + e^{\alpha_{20} + \alpha_1(i-1) + \alpha_2(j-1)}} \right)^{n_{2i}} \times \\
& \quad \prod_{j=1}^2 \left(\sum_{i=1}^2 \theta_{ij} \frac{1}{1 + e^{\alpha_{10} + \alpha_1(i-1) + \alpha_2(j-1)}} \frac{e^{\alpha_{30} + \alpha_1(i-1) + \alpha_2(j-1)}}{1 + e^{\alpha_{30} + \alpha_1(i-1) + \alpha_2(j-1)}} \right)^{n_{3j}} \times \\
& \quad \left(\sum_{i=1}^2 \sum_{j=1}^2 \theta_{ij} \frac{1}{1 + e^{\alpha_{10} + \alpha_1(i-1) + \alpha_2(j-1)}} \frac{1}{1 + e^{\alpha_{30} + \alpha_1(i-1) + \alpha_2(j-1)}} \right)^{N_4}
\end{aligned}$$

Como as rotinas de otimização do R, em geral, buscam o valor dos parâmetros que minimizam a função, deve-se explicitar o valor negativo da log-verossimilhança para obter a estimativa de MV.

```

> e134.TF2<-c(e134.TF,24) #para mecanismos MNAR, cenários de omissão total trazem inf.na estim.
> mnarsat.mlv<-function(p,n111=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
+ n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
+ #p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1,\alpha_2
+ t11<-p[1];t12<-p[2];t21<-p[3]
+ a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7];a2<-p[8]
+ value<- -(
+ n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
+ n112*log( t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))) )+
+ n121*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
+ n122*log( (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
+ (exp(a20+a1+a2)/(1+exp(a20+a1+a2))) )+
+ n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
+ t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))) )+
+ n22*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
+ (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
+ (1/(1+exp(a20+a1+a2))) )+
+ n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
+ t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
+ n32*log( t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))) +
+ (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*
+ (exp(a30+a1+a2)/(1+exp(a30+a1+a2))) )+
+ N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
+ t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))) +
+ t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
+ (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2))) )
+ )
+ value

```

```
+ }
> require(geoR) #.nlmP adapta nlm p/restringir o espaço paramétrico. Isso é importante,
Carregando pacotes exigidos: geoR
Carregando pacotes exigidos: sp
```

```
-----
Analysis of geostatistical data
For an Introduction to geoR go to http://www.est.ufpr.br/geoR
geoR version 1.6-13 (built on 2006/12/26) is now loaded
-----
```

```
[1] TRUE
> inipars<-c(0.25,0.25,0.25,0,0,0,0,0) #pois mecanismos MNAR resultam facilmente em estims.
> minpars<-c(0,0,0,-Inf,-Inf,-Inf,-Inf,-Inf) #p/probs. >1 ou <0 quando não se usa o EM ou
> maxpars<-c(1,1,1,Inf,Inf,Inf,Inf,Inf) #funções ligações próprias para probs.(e.g.,logito)
> mnarsat<-nlmP(objfunc=mnarsat.mlv,params=inipars,lower=minpars,upper=maxpars,hessian=T)
Warning messages:
1: NaNs produzidos in: log(x)
2: NaNs produzidos in: log(x)
3: NaNs produzidos in: log(x)
4: NaNs produzidos in: log(x)
5: NA/Inf substituído pelo máximo valor positivo
6: NaNs produzidos in: log(x)
7: NaNs produzidos in: log(x)
8: NaNs produzidos in: log(x)
9: NA/Inf substituído pelo máximo valor positivo
> round(mnarsat$est,3) #estimativas de MV
[1] 0.432 0.170 0.268 0.686 -1.100 0.692 -0.343 -0.150
> round(-mnarsat$min ,3) #valor máximo da log-verossimilhança
[1] -312.773
> mnarsat$code #veja o significado dos códigos de retorno em ?nlm
[1] 1
> mnarsat$it #número de iterações
[1] 23
> mnarsat.der<-deriv3(~-(
+ n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
+ n112*log( t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))) )+
+ n121*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
+ n122*log( (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
+ (exp(a20+a1+a2)/(1+exp(a20+a1+a2))) )+
+ n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
+ t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))) )+
+ n22*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
+ (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
+ (1/(1+exp(a20+a1+a2))) )+
+ n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
+ t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
+ n32*log( t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))) +
+ (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*
+ (exp(a30+a1+a2)/(1+exp(a30+a1+a2))) )+
+ N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
+ t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))) +
+ t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
+ (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2))) )
+ ),c("t11","t12","t21","a10","a20","a30","a1","a2"),
+ c("t11","t12","t21","a10","a20","a30","a1","a2",
```

```

+ "n111","n112","n121","n122","n21","n22","n31","n32","N4")
+ ) #obtem o gradiente e a hessiana analiticamente
> p<-mnarsat$est
> mnarsat.infobs<-attr(mnarsat.der(p[1],p[2],p[3],p[4],p[5],p[6],p[7],p[8],e134.TF2[1],e134.TF2[2],
+ e134.TF2[3],e134.TF2[4],e134.TF2[5],e134.TF2[6],e134.TF2[7],e134.TF2[8],e134.TF2[9]),
+ "hessian")[1,,]
> mnarsat.infobs2<-mnarsat$hess
> mnarsat.infobs #matriz informaçao observada estimada obtida analiticamente
      t11      t12      t21      a10      a20      a30      a1      a2
t11 713.952827 458.7869163 315.215193 8.97857823 1.77667252 6.2658116 -30.240002 -9.221777
t12 458.786916 584.1760446 123.605033 5.84010860 -0.44729489 5.9402887 -31.673994 8.628583
t21 315.215193 123.6050333 395.289574 4.18490448 1.70223217 1.3514951 5.781306 -9.581616
a10 8.978578 5.8401086 4.184904 38.78733698 -0.01474706 -0.1096258 16.863347 12.033076
a20 1.776673 -0.4472949 1.702232 -0.01474706 17.80757990 0.0000000 5.697800 4.689270
a30 6.265812 5.9402887 1.351495 -0.10962576 0.00000000 14.6246245 7.587926 5.035780
a1 -30.240002 -31.6739942 5.781306 16.86334733 5.69780027 7.5879265 20.711145 9.649646
a2 -9.221777 8.6285829 -9.581616 12.03307571 4.68927045 5.0357796 9.649646 13.230589
> round(mnarsat.infobs2,7)#matriz informaçao observada estimada obtida numericamente
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,] 43.0161833 15.9210344 15.1746804 2.2037739 0.4360857 1.5379271 -7.422756 -2.263732
[2,] 15.9210344 11.6771218 3.4275729 0.8256393 -0.0632326 0.8397876 -4.478085 1.219894
[3,] 15.1746804 3.4275729 15.2039718 0.8206825 0.3338130 0.2650609 1.133623 -1.879283
[4,] 2.2037739 0.8256393 0.8206825 38.7863963 -0.0147509 -0.1096225 16.862964 12.032803
[5,] 0.4360857 -0.0632326 0.3338130 -0.0147509 17.8085543 -0.0000057 5.698161 4.689610
[6,] 1.5379271 0.8397876 0.2650609 -0.1096225 -0.0000057 14.6242883 7.587681 5.035543
[7,] -7.4227557 -4.4780847 1.1336226 16.8629640 5.6981605 7.5876812 20.709956 9.649483
[8,] -2.2637323 1.2198939 -1.8792832 12.0328025 4.6896105 5.0355425 9.649483 13.230311
> mnarsat.cov<-solve(mnarsat.infobs)
> mnarsat.cov2<-solve(mnarsat.infobs2)
> round(mnarsat.cov,6) #compare os resultados
      t11      t12      t21      a10      a20      a30      a1      a2
t11 0.005998 -0.002893 -0.003739 -0.010505 -0.007907 -0.012750 0.016955 0.008203
t12 -0.002893 0.004389 0.000619 -0.000378 0.000420 -0.001214 0.007135 -0.008977
t21 -0.003739 0.000619 0.005489 0.007372 0.005079 0.009494 -0.017762 0.001801
a10 -0.010505 -0.000378 0.007372 0.093390 0.050602 0.081659 -0.113047 -0.053238
a20 -0.007907 0.000420 0.005079 0.050602 0.094319 0.060618 -0.080172 -0.046158
a30 -0.012750 -0.001214 0.009494 0.081659 0.060618 0.166906 -0.140349 -0.058136
a1 0.016955 0.007135 -0.017762 -0.113047 -0.080172 -0.140349 0.259097 -0.010019
a2 0.008203 -0.008977 0.001801 -0.053238 -0.046158 -0.058136 -0.010019 0.182673
> round(mnarsat.cov2,6)#sugere-se usar a matriz analitica
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,] 0.099545 -0.083333 -0.077665 -0.042802 -0.032219 -0.051949 0.069091 0.033413
[2,] -0.083333 0.219526 0.022302 -0.002684 0.002967 -0.008597 0.050479 -0.063487
[3,] -0.077665 0.022302 0.142708 0.037598 0.025902 0.048418 -0.090588 0.009192
[4,] -0.042802 -0.002684 0.037598 0.093404 0.050611 0.081674 -0.113076 -0.053239
[5,] -0.032219 0.002967 0.025902 0.050611 0.094323 0.060629 -0.080192 -0.046159
[6,] -0.051949 -0.008597 0.048418 0.081674 0.060629 0.166925 -0.140383 -0.058135
[7,] 0.069091 0.050479 -0.090588 -0.113076 -0.080192 -0.140383 0.259161 -0.010022
[8,] 0.033413 -0.063487 0.009192 -0.053239 -0.046159 -0.058135 -0.010022 0.182678
> round(sqrt(diag(mnarsat.cov)),4)
      t11      t12      t21      a10      a20      a30      a1      a2
0.0774 0.0662 0.0741 0.3056 0.3071 0.4085 0.5090 0.4274
> round(sqrt(diag(mnarsat.cov2)),4)
[1] 0.3155 0.4685 0.3778 0.3056 0.3071 0.4086 0.5091 0.4274
> mnarsat.esp<-function(p,N){
+ #p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1,\alpha_2

```

```

+ t11<-p[1];t12<-p[2];t21<-p[3]
+ a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7];a2<-p[8]
+ value<-N*c(
+ t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
+ t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))),
+ t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),
+ (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*(exp(a20+a1+a2)/(1+exp(a20+a1+a2))),
+ t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
+ t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))),
+ t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
+ (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*(1/(1+exp(a20+a1+a2))),
+ t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
+ t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))),
+ t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
+ (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(exp(a30+a1+a2)/(1+exp(a30+a1+a2))),
+ t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
+ t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))),
+ t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))),
+ (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2)))
+ )
+ value
+ }
> mnarsat.esp(p=p,sum(e134.TF2)) #precisa-se organizar para ficar como na Tabela 13.11
[1] 11.999993 4.000002 4.999982 1.999996 36.043030 13.956975 21.165034 9.834911 16.116336
[10] 6.639642 10.883793 5.360405 8.067486 3.861067 7.678253 4.393095
> matrix(mnarsat.esp(p=p,sum(e134.TF2))[rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2)
[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,] 11.999993 4.000002 36.04303 13.956975 16.11634 6.639642 8.067486 3.861067
[2,] 4.999982 1.999996 21.16503 9.834911 10.88379 5.360405 7.678253 4.393095
> b<-c(rep(0,3),1)
> B<-rbind(diag(3),rep(-1,3))
> c(b+B%*p[1:3]) #vetor completo das estimativas das probs, incluindo 1-p[1]-p[2]-p[3]
[1] 0.4324961 0.1704053 0.2678267 0.1292719
> B%*mnarsat.cov[1:3,1:3]%%t(B) #correspondente vetor de covariâncias
[,1] [,2] [,3] [,4]
[1,] 0.005997926 -0.0028927133 -0.0037386683 0.000633456
[2,] -0.002892713 0.0043885599 0.0006188606 -0.002114707
[3,] -0.003738668 0.0006188606 0.0054886579 -0.002368850
[4,] 0.000633456 -0.0021147072 -0.0023688501 0.003850101
> sqrt(diag(B%*mnarsat.cov[1:3,1:3]%%t(B)))
[1] 0.07744628 0.06624621 0.07408548 0.06204918
> e134.A<-rbind(c(1,1,0,0),c(1,0,1,0))
> mnarsat.wls<-funlinWLS(model="lin",theta=c(b+B%*p[1:3]),Vtheta=
+ B%*mnarsat.cov[1:3,1:3]%%t(B),A1=e134.A,X=rep(1,2))
> mnarsat.wls2<-funlinWLS(model="lin",theta=c(b+B%*p[1:3]),Vtheta=
+ B%*mnarsat.cov[1:3,1:3]%%t(B),A1=t(c(0,1,-1,0)),X=1)
> mnarsat.wls

```

Call: funlinWLS(model = "lin", theta = ..., Vtheta = ..., A1 = e134.A, X = rep(1, 2))

Weighted least squares estimates of the parameters of the model:

	estimate	std.error	z-value	p-value
[1,]	0.6549	0.0462	14.1740	0.0000

Wald goodness of fit statistic of the model (d.f.=1): 1.0986 (p-value=0.2946)

```

> summary(mnarsat.wls)

Call: funlinWLS(model = "lin", theta = ..., Vtheta = ..., A1 = e134.A, X = rep(1, 2))

Weighted least squares estimates of the functions:
      observed  std.error  under the model  std.error
[1,]  0.6029    0.0678    0.6549             0.0462
[2,]  0.7003    0.0633    0.6549             0.0462

Weighted least squares estimates of the parameters of the model:
      estimate  std.error  z-value  p-value
[1,]  0.6549    0.0462    14.1740  0.0000

Wald goodness of fit statistic of the model (d.f.=1): 1.0986 (p-value=0.2946)

> mnarsat.wls2

Call: funlinWLS(model = "lin", theta = ..., Vtheta = ..., A1 = t(c(0, 1, -1, 0)), X = 1)

Weighted least squares estimates of the parameters of the model:
      estimate  std.error  z-value  p-value
[1,] -0.0974    0.0929    -1.0481  0.2946

Wald goodness of fit statistic of the model (d.f.=0): 0 (p-value=1)

```

A rotina `waldTest()` também pode ser utilizada como explicado na seção anterior.

As Figuras 1 e 2 resumem as informações apresentadas nesta seção e na antecedente.

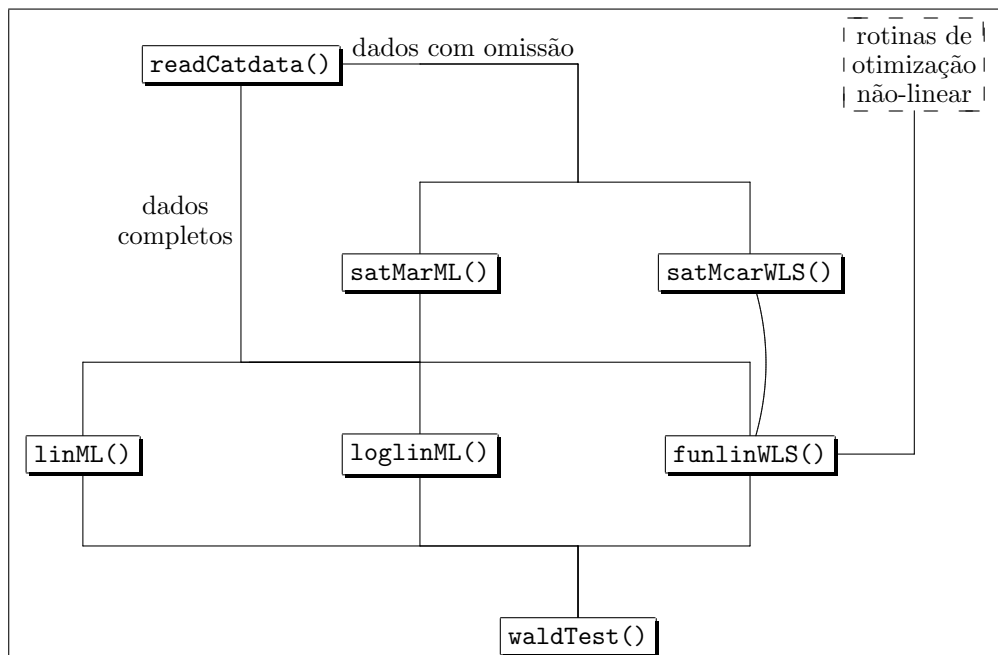


Figura 1: Hierarquia de uso das rotinas

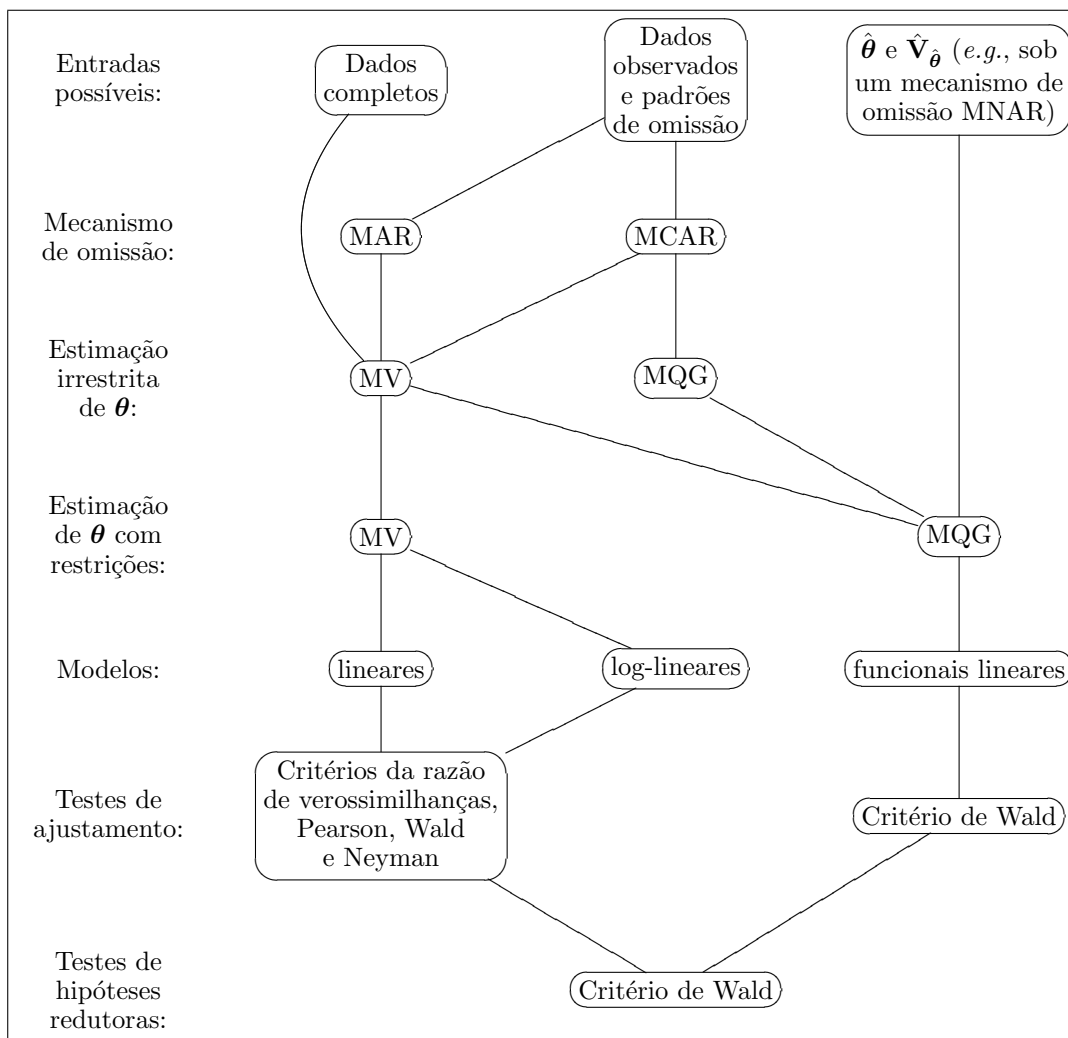


Figura 2: Análises possíveis de serem realizadas com a biblioteca *Catdata*

6 Comandos para reproduzir as análises dos exemplos

Apresenta-se os comandos utilizados para obter os resultados discutidos nos exemplos dos Capítulos 8 ao 13. Os ajustes dos modelos são apenas armazenados em objetos. O leitor deve obter os resultados utilizando o `print()` ou `summary()` do objeto. Resultados que não possam ser obtidos dessas saídas possuem comandos adicionais. Algumas vezes ajusta-se mais de um modelo para ilustrar outras maneiras de obter um resultado além das discutidas no livro. Outros exemplos inter-relacionados do livro também são mencionados, juntamente com sua página inicial.

As rotinas e todos os comandos descritos neste texto foram testados no R versões 1.5.1 a 2.4.1 no sistema Windows XP. Não se sabe como pode ser o comportamento em outras versões. Agradece-se a comunicação de erros e sugestões.

Exemplo 8.1 (p.228) / 3.1 (p.47): *Problema da intenção de voto*

```
e81.TF<-c(192,1,5,2,146,5,11,12,71)
e81.catdata<-readCatdata(TF=e81.TF)
e81.U<-rbind(c(0,-1, 0,1,0, 0,0,0),
             c(0, 0,-1,0,0, 0,1,0),
             c(0, 0, 0,0,0,-1,0,1))
e81.X<-rbind(c(1,0,0,0,0),
             c(0,1,0,0,0),
             c(0,0,1,0,0),
             c(0,1,0,0,0),
             c(0,0,0,1,0),
             c(0,0,0,0,1),
             c(0,0,1,0,0),
             c(0,0,0,0,1))
e81.linml1<-linML(e81.catdata,U=e81.U) #simetria
e81.linml2<-linML(e81.catdata,X=e81.X) #simetria
```

Exemplo 8.2 (p.233) / 3.2 (p.49) / 1.2 (p.4): *Problema do risco de cárie dentária*

```
e82.TF<-c(11,5,0,14,34,7,2,13,11)
e82.catdata<-readCatdata(TF=e82.TF)
e82.U<-rbind(c(0, 1,1,-1,0,0,-1, 0),
             c(0,-1,0, 1,0,1, 0,-1))
e82.X<-rbind(c(1, 0, 0,0,0,0),
             c(0, 1, 0,0,0,0),
             c(0,-1, 1,0,1,0),
             c(0, 0, 1,0,0,0),
             c(0, 0, 0,1,0,0),
             c(0, 1,-1,0,0,1),
             c(0, 0, 0,0,1,0),
             c(0, 0, 0,0,0,1))
e82.linml1<-linML(e82.catdata,U=e82.U) #homogeneidade marginal
e82.linml2<-linML(e82.catdata,X=e82.X) #homogeneidade marginal
e82.A<-rbind(c(1,1,1,0,0,0,0,0,0),
             c(0,0,0,1,1,1,0,0,0),
             c(1,0,0,1,0,0,1,0,0),
             c(0,1,0,0,1,0,0,1,0) )
e82.U2<-rbind(c(1,0,-1, 0),
              c(0,1, 0,-1))
e82.X2<-rbind(c(1,0),
              c(0,1),
              c(1,0),
              c(0,1) )
e82.linml3<-linML(e82.catdata,A=e82.A,U=e82.U2) #homogeneidade marginal
e82.linml4<-linML(e82.catdata,A=e82.A,X=e82.X2) #homogeneidade marginal
```

Exemplo 8.3 (p.236) / 3.3 (p.50) / 1.9 (p.12): *Problema do tamanho da ninhada*

```
e83.TF<-rbind(c(10,21, 96,23),
              c( 4, 6, 28, 8),
              c( 9, 7, 58, 7),
              c( 8,19, 44, 1),
              c( 5,17, 56, 1),
              c( 1, 5, 20, 2),
              c(22,95,103, 4),
              c(18,49, 62, 0),
```

```

      c( 4,12, 16, 2))
e83.catdata<-readCatdata(TF=e83.TF)
e83.A<-kronecker(diag(9),t(c(0,1,2,3)))
e83.U<-rbind(c(-1,1,0,1,-1, 0,0, 0, 0),
             c(-1,1,0,0, 0, 0,1,-1, 0),
             c(-1,0,1,1, 0,-1,0, 0, 0),
             c(-1,0,1,0, 0, 0,1, 0,-1))
e83.X<-rbind(c(1,0,0,0,0),
             c(1,1,0,0,0),
             c(1,0,1,0,0),
             c(1,0,0,1,0),
             c(1,1,0,1,0),
             c(1,0,1,1,0),
             c(1,0,0,0,1),
             c(1,1,0,0,1),
             c(1,0,1,0,1))
e83.linml1<-linML(e83.catdata,A=e83.A,U=e83.U,epsilon2=1e-4)
e83.linml2<-linML(e83.catdata,A=e83.A,X=e83.X,epsilon2=1e-5)
waldTest(e83.linml2,rbind(c(0,0,0,1,0),c(0,0,0,0,1)))
waldTest(e83.linml2,rbind(c(0,1,0,0,0),c(0,0,1,0,0)))

```

Exemplo 9.1 (p.263): *Problema da anemia*

```

e91.TF<-c(3,25,32,68)
e91.catdata<-readCatdata(TF=e91.TF)
e91.U<-c(1,-1,-1,1)
e91.X<-rbind(c(0,0),
             c(0,1),
             c(1,0),
             c(1,1))
e91.X2<-rbind(c(0,0,0),
             c(0,1,0),
             c(1,0,0),
             c(1,1,1))
e91.loglinml1<-loglinML(e91.catdata,U=e91.U) #independência
e91.loglinml2<-loglinML(e91.catdata,X=e91.X) #independência
e91.loglinml3<-loglinML(e91.catdata,X=e91.X2) #modelo saturado
e91.loglinml4<-loglinML(e91.catdata,A=c(1,-1,-1,1),XL=1) #modelo saturado
round(e91.loglinml4$beta+c(-1,1)*qnorm(0.975)*sqrt(e91.loglinml4$Vbeta),3)
round(exp(e91.loglinml4$beta),3) #razão de chances
round(exp(e91.loglinml4$beta+c(-1,1)*qnorm(0.975)*sqrt(e91.loglinml4$Vbeta)),3)

```

Exemplo 9.2 (p.267): *Problema da acuidade visual*

```

e92.TF<-c(1520,266,124,66, 234,1512,432,78, 117,362,1772,205, 36,82,179,492)
e92.catdata<-readCatdata(TF=e92.TF)
e92.X1<-rbind(c(1,0,0,0,0,0,0,0,0),
             c(0,1,0,0,0,0,0,0,0),
             c(0,0,1,0,0,0,0,0,0),
             c(0,0,0,1,0,0,0,0,0),

             c(0,1,0,0,0,0,0,0,0),
             c(0,0,0,0,1,0,0,0,0),
             c(0,0,0,0,0,1,0,0,0),
             c(0,0,0,0,0,0,1,0,0),
             c(0,0,0,0,0,0,0,1,0,0),

```



```

c(0,0,1,0,0,0,0,0,0),
c(0,0,0,0,0,1,0,0,0),
c(0,0,0,0,0,0,0,1,0),
c(0,0,0,0,0,0,0,0,1),

c(0,0,0,1,0,0,0,0,0),
c(0,0,0,0,0,0,1,0,0),
c(0,0,0,0,0,0,0,0,1))
e92.linml1<-linML(e92.catdata,X=e92.X1) #simetria em formulação linear
e92.A1<-rbind(c(0,1,0,0, -1,0,0,0, 0, 0,0,0, 0, 0, 0,0),
c(0,0,1,0, 0,0,0,0, -1, 0,0,0, 0, 0, 0,0),
c(0,0,0,1, 0,0,0,0, 0, 0,0,0, -1, 0, 0,0),
c(0,0,0,0, 0,0,1,0, 0,-1,0,0, 0, 0, 0,0),
c(0,0,0,0, 0,0,0,1, 0, 0,0,0, 0,-1, 0,0),
c(0,0,0,0, 0,0,0,0, 0, 0,0,1, 0, 0,-1,0))
e92.linml2<-linML(e92.catdata,U=e92.A1[,1:15]) #simetria em formulação linear
#u_1,u_2,u_3, u_{11},u_{12},u_{13},u_{22},u_{23},u_{33}
e92.X2<-rbind(c( 2, 0, 0, 1, 0, 0, 0, 0, 0),
c( 1, 1, 0, 0, 1, 0, 0, 0, 0),
c( 1, 0, 1, 0, 0, 1, 0, 0, 0),
c( 0,-1,-1, -1,-1,-1, 0, 0, 0),

c( 1, 1, 0, 0, 1, 0, 0, 0, 0),
c( 0, 2, 0, 0, 0, 0, 1, 0, 0),
c( 0, 1, 1, 0, 0, 0, 0, 1, 0),
c(-1, 0,-1, 0,-1, 0,-1,-1, 0),

c( 1, 0, 1, 0, 0, 1, 0, 0, 0),
c( 0, 1, 1, 0, 0, 0, 0, 1, 0),
c( 0, 0, 2, 0, 0, 0, 0, 0, 1),
c(-1,-1, 0, 0, 0,-1, 0,-1,-1),

c( 0,-1,-1, -1,-1,-1, 0, 0, 0),
c(-1, 0,-1, 0,-1, 0,-1,-1, 0),
c(-1,-1, 0, 0, 0,-1, 0,-1,-1),
c(-2,-2,-2, 1, 2, 2, 1, 2, 1)) #análogo à matriz da pág.71
e92.loglinml1<-loglinML(e92.catdata,X=e92.X2) #simetria em formulação log-linear
e92.A2<-rbind(cbind(kronecker(diag(3),t(rep(1,4))),matrix(0,3,4)),
kronecker(t(rep(1,4)),cbind(diag(3),rep(0,3))))
e92.linml3<-linML(e92.catdata,A=e92.A2,X=kronecker(rep(1,2),diag(3))) #homogeneidade marginal
#u_1^A,u_2^A,u_3^A, u_1^B,u_2^B,u_3^B, u_{11},u_{12},u_{13},u_{22},u_{23},u_{33}
e92.X3<-rbind(c( 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0),
c( 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0),
c( 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0),
c( 1, 0, 0, -1,-1,-1, -1,-1,-1, 0, 0, 0),

c( 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0),
c( 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0),
c( 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0),
c( 0, 1, 0, -1,-1,-1, 0,-1, 0,-1,-1, 0),

c( 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0),
c( 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0),
c( 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1),
c( 0, 0, 1, -1,-1,-1, 0, 0,-1, 0,-1,-1),

```

```

      c(-1,-1,-1, 1, 0, 0, -1,-1,-1, 0, 0, 0),
      c(-1,-1,-1, 0, 1, 0, 0,-1, 0,-1,-1, 0),
      c(-1,-1,-1, 0, 0, 1, 0, 0,-1, 0,-1,-1),
      c(-1,-1,-1, -1,-1,-1, 1, 2, 2, 1, 2, 1))
#e92.X2<-cbind(e92.X3[,1]+e92.X3[,4],e92.X3[,2]+e92.X3[,5],e92.X3[,3]+e92.X3[,6],e92.X3[,7:12])
e92.loglinml2<-loglinML(e92.catdata,X=e92.X3) #quasi-simetria
#u_1,u_2,u_3, u_{11},u_{12},u_{13},u_{22},u_{23},u_{33},gama
e92.X4<-rbind(c( 2, 0, 0, 1, 0, 0, 0, 0, 0, 0),
              c( 1, 1, 0, 0, 1, 0, 0, 0, 0, 1),
              c( 1, 0, 1, 0, 0, 1, 0, 0, 0, 1),
              c( 0,-1,-1, -1,-1,-1, 0, 0, 0, 1),

              c( 1, 1, 0, 0, 1, 0, 0, 0, 0, 0),
              c( 0, 2, 0, 0, 0, 0, 1, 0, 0, 0),
              c( 0, 1, 1, 0, 0, 0, 0, 1, 0, 1),
              c(-1, 0,-1, 0,-1, 0,-1,-1, 0, 1),

              c( 1, 0, 1, 0, 0, 1, 0, 0, 0, 0),
              c( 0, 1, 1, 0, 0, 0, 0, 1, 0, 0),
              c( 0, 0, 2, 0, 0, 0, 0, 0, 1, 0),
              c(-1,-1, 0, 0, 0,-1, 0,-1,-1, 1),

              c( 0,-1,-1, -1,-1,-1, 0, 0, 0, 0),
              c(-1, 0,-1, 0,-1, 0,-1,-1, 0, 0),
              c(-1,-1, 0, 0, 0,-1, 0,-1,-1, 0),
              c(-2,-2,-2, 1, 2, 2, 1, 2, 1, 0))
e92.loglinml3<-loglinML(e92.catdata,X=e92.X4) #simetria condicional
e92.loglinml4<-loglinML(e92.catdata,A=e92.A1,XL=rep(1,6)) #simetria condicional
round(e92.loglinml1$QvH-e92.loglinml3$QvH,3)
round(1-pchisq(e92.loglinml1$QvH-e92.loglinml3$QvH,1),3)
round(e92.loglinml1$QwH-e92.loglinml3$QwH,3)
round(1-pchisq(e92.loglinml1$QwH-e92.loglinml3$QwH,1),3)
round(exp(e92.loglinml4$beta),3)
round(kronecker(diag(2),rbind(c(1,0,0),c(1,1,0),c(1,1,1))))*%e92.A2*%e92.loglinml4$thetaH,3)

```

Exemplo 9.3 (p.269) / 1.6 (p.11): *Problema dos defeitos de fibras têxteis*

```

e93.TF<-c(28,40,68,5,21,49,1,4,15)
e93.catdata<-readCatdata(TF=e93.TF)
e93.U1<-rbind(c(1,-1, 0,-1, 1, 0, 0, 0,0),
              c(0, 1,-1, 0,-1, 1, 0, 0,0),
              c(0, 0, 0, 1,-1, 0,-1, 1,0),
              c(0, 0, 0, 0, 1,-1, 0,-1,1))
e93.X1<-rbind(c( 1, 0, 1, 0),
              c( 1, 0, 0, 1),
              c( 1, 0,-1,-1),
              c( 0, 1, 1, 0),
              c( 0, 1, 0, 1),
              c( 0, 1,-1,-1),
              c(-1,-1, 1, 0),
              c(-1,-1, 0, 1),
              c(-1,-1,-1,-1))
e93.loglinml1<-loglinML(e93.catdata,U=e93.U1) #indep.
e93.loglinml2<-loglinML(e93.catdata,X=e93.X1) #indep.
e93.X2<-rbind(c( 1, 0, 1, 0,-1, 0),
              c( 1, 0, 0, 1, 0, 0),

```

```

      c( 1, 0,-1,-1, 1, 0),
      c( 0, 1, 1, 0, 0,-1),
      c( 0, 1, 0, 1, 0, 0),
      c( 0, 1,-1,-1, 0, 1),
      c(-1,-1, 1, 0, 1, 1),
      c(-1,-1, 0, 1, 0, 0),
      c(-1,-1,-1,-1,-1,-1))
e93.loglinml3<-loglinML(e93.catdata,X=e93.X2) #efeito de linha
round(-2*e93.loglinml3$beta[5]-e93.loglinml3$beta[6],3) #w_3=-w_1-w_2
round(exp(-2*e93.loglinml3$beta[5]-e93.loglinml3$beta[6]),2)
round(e93.loglinml1$QvH-e93.loglinml3$QvH,2)
round(1-pchisq(e93.loglinml1$QvH-e93.loglinml3$QvH,2),2)
round(e93.loglinml1$QwH-e93.loglinml3$QwH,2)
round(1-pchisq(e93.loglinml1$QwH-e93.loglinml3$QwH,2),2)
e93.X3<-rbind(c( 1, 0, 1, 0, 1),
              c( 1, 0, 0, 1, 0),
              c( 1, 0,-1,-1,-1),
              c( 0, 1, 1, 0, 0),
              c( 0, 1, 0, 1, 0),
              c( 0, 1,-1,-1, 0),
              c(-1,-1, 1, 0,-1),
              c(-1,-1, 0, 1, 0),
              c(-1,-1,-1,-1, 1))
e93.loglinml4<-loglinML(e93.catdata,X=e93.X3) #associação uniforme
round(e93.loglinml4$QvH-e93.loglinml3$QvH,2)
round(1-pchisq(e93.loglinml4$QvH-e93.loglinml3$QvH,1),2)
round(e93.loglinml4$QwH-e93.loglinml3$QwH,2)
round(1-pchisq(e93.loglinml4$QwH-e93.loglinml3$QwH,1),2)
round(exp(2*2*e93.loglinml4$beta[5]),2)
round(exp(e93.loglinml4$beta[5]),2)
round(exp(e93.loglinml4$beta[5])*exp(c(-1,1)*qnorm(0.975)*sqrt(e93.loglinml4$Vbeta[5,5])),2)
round(e93.loglinml1$QvH-e93.loglinml4$QvH,2)
round(1-pchisq(e93.loglinml1$QvH-e93.loglinml4$QvH,1),2)
round(e93.loglinml1$QwH-e93.loglinml4$QwH,2)
round(1-pchisq(e93.loglinml1$QwH-e93.loglinml4$QwH,1),2)

```

Exemplo 9.4 (p.274) / 1.11 (p.13): *Problema da fobia em alcoólatras*

```

e94.TF<-c(10,24,6,12,13,17,4,7)
e94.catdata<-readCatdata(TF=e94.TF)
e94.X<-rbind(c(0,0,0, 0,0,0, 0),
             c(0,0,1, 0,0,0, 0),
             c(0,1,0, 0,0,0, 0),
             c(0,1,1, 0,0,1, 0),
             c(1,0,0, 0,0,0, 0),
             c(1,0,1, 0,1,0, 0),
             c(1,1,0, 1,0,0, 0),
             c(1,1,1, 1,1,1, 1))
e94.loglinml1<-loglinML(e94.catdata,X=e94.X) #ABC
e94.loglinml2<-loglinML(e94.catdata,X=e94.X[,-7]) #AB,AC,BC
e94.loglinml3<-loglinML(e94.catdata,X=e94.X[-(6:7)]) #AB,AC
e94.loglinml4<-loglinML(e94.catdata,X=e94.X[-(5:7)]) #AB,C
e94.loglinml5<-loglinML(e94.catdata,X=e94.X[-(4:7)]) #A,B,C
exp(e94.loglinml1$beta[4:6]) #Assoc.parcial ABC nível 1
exp(e94.loglinml1$beta[4:6]+e94.loglinml1$beta[7]) #Assoc.parcial ABC nível 2
exp(e94.loglinml2$beta[4:6]) #Assoc.parcial AB,AC,BC

```

```

exp(e94.loglinml3$beta[4:5]) #Assoc.parcial AB,AC
exp(e94.loglinml4$beta[4]) #Assoc.parcial AB,C
exp(t(c(1,-1,-1,1))*%log(e94.loglinml1$thetaH[c(1,3,5,7)]+e94.loglinml1$thetaH[c(2,4,6,8)])
) #Assoc.marg.ABC - AB
exp(t(c(1,-1,-1,1))*%log(e94.loglinml1$thetaH[c(1,2,5,6)]+e94.loglinml1$thetaH[c(3,4,7,8)])
) #Assoc.marg.ABC - AC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml1$thetaH[1:4]+e94.loglinml1$thetaH[5:8])
) #Assoc.marg.ABC - BC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml2$thetaH[c(1,3,5,7)]+e94.loglinml2$thetaH[c(2,4,6,8)])
) #Assoc.marg.AB,AC,BC - AB
exp(t(c(1,-1,-1,1))*%log(e94.loglinml2$thetaH[c(1,2,5,6)]+e94.loglinml2$thetaH[c(3,4,7,8)])
) #Assoc.marg.AB,AC,BC - AC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml2$thetaH[1:4]+e94.loglinml2$thetaH[5:8])
) #Assoc.marg.AB,AC,BC - BC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml3$thetaH[c(1,3,5,7)]+e94.loglinml3$thetaH[c(2,4,6,8)])
) #Assoc.marg.AB,AC - AB
exp(t(c(1,-1,-1,1))*%log(e94.loglinml3$thetaH[c(1,2,5,6)]+e94.loglinml3$thetaH[c(3,4,7,8)])
) #Assoc.marg.AB,AC - AC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml3$thetaH[1:4]+e94.loglinml3$thetaH[5:8])
) #Assoc.marg.AB,AC - BC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml4$thetaH[c(1,3,5,7)]+e94.loglinml4$thetaH[c(2,4,6,8)])
) #Assoc.marg.AB,C - AB
exp(t(c(1,-1,-1,1))*%log(e94.loglinml4$thetaH[c(1,2,5,6)]+e94.loglinml4$thetaH[c(3,4,7,8)])
) #Assoc.marg.AB,C - AC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml4$thetaH[1:4]+e94.loglinml4$thetaH[5:8])
) #Assoc.marg.AB,C - BC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml5$thetaH[c(1,3,5,7)]+e94.loglinml5$thetaH[c(2,4,6,8)])
) #Assoc.marg.A,B,C - AB
exp(t(c(1,-1,-1,1))*%log(e94.loglinml5$thetaH[c(1,2,5,6)]+e94.loglinml5$thetaH[c(3,4,7,8)])
) #Assoc.marg.A,B,C - AC
exp(t(c(1,-1,-1,1))*%log(e94.loglinml5$thetaH[1:4]+e94.loglinml5$thetaH[5:8])
) #Assoc.marg.A,B,C - BC
e94.X2<-rbind(c( 1, 1, 1, 1, 1, 1, 1),
             c( 1, 1,-1, 1,-1,-1,-1),
             c( 1,-1, 1,-1, 1,-1,-1),
             c( 1,-1,-1,-1,-1, 1, 1),
             c(-1, 1, 1,-1,-1, 1,-1),
             c(-1, 1,-1,-1, 1,-1, 1),
             c(-1,-1, 1, 1,-1,-1, 1),
             c(-1,-1,-1, 1, 1, 1,-1)) #para obter os resultados da Tabela 9.10
e94.loglinml6<-loglinML(e94.catdata,X=e94.X2) #ABC
e94.loglinml7<-loglinML(e94.catdata,X=e94.X2[,-(6:7)]) #AB,AC
e94.loglinml8<-loglinML(e94.catdata,X=e94.X2[,-(4:7)]) #A,B,C

```

Exemplo 9.5 (p.278) / Exercício 8.12 (p.244): *Problema da obesidade juvenil*

```

e95.TF<-c(300,17,18,19,18,7,17,52)
e95.catdata<-readCatdata(TF=e95.TF)
#u_1^A,u_1^B,u_1^C,u_{11},u_{111}
e95.X1<-rbind(c( 1, 1, 1, 3, 1),
             c( 1, 1,-1,-1,-1),
             c( 1,-1, 1,-1,-1),
             c( 1,-1,-1,-1, 1),
             c(-1, 1, 1,-1,-1),
             c(-1, 1,-1,-1, 1),
             c(-1,-1, 1,-1, 1),

```

```

      c(-1,-1,-1, 3,-1))
e95.loglinml1<-loglinML(e95.catdata,X=e95.X1) #quasi-simetria
round(1-rbind(c(rep(1,4),rep(0,4)),rep(c(1,1,0,0),2),rep(c(1,0),4)))*%e95.loglinml1$thetaH,3)
#u_1,u_{11},u_{111}
e95.X2<-rbind(c( 3, 3, 1),
              c( 1,-1,-1),
              c( 1,-1,-1),
              c(-1,-1, 1),
              c( 1,-1,-1),
              c(-1,-1, 1),
              c(-1,-1, 1),
              c(-3, 3,-1))
e95.loglinml2<-loglinML(e95.catdata,X=e95.X2) #simetria completa (ver exerc.4.18-c)
round(e95.loglinml2$QvH-e95.loglinml1$QvH,2)
round(1-pchisq(e95.loglinml2$QvH-e95.loglinml1$QvH,2),2)
round(e95.loglinml2$QwH-e95.loglinml1$QwH,2)
round(1-pchisq(e95.loglinml2$QwH-e95.loglinml1$QwH,2),2)

```

Exemplo 9.6 (p.281) / 9.3 (p.269) / 1.6 (p.11): *Problema dos defeitos de fibras têxteis*

```

e96.TF<-c(28,40,68,5,21,49,1,4,15, 31,70,69,5,12,10,0,1,2)
e96.catdata<-readCatdata(TF=e96.TF)
e96.X1<-rbind(c(0,0,0,0,0, 0,0,0,0,0,0,0, 1),
              c(0,1,0,0,0, 0,0,0,0,0,0,0, 0),
              c(0,0,1,0,0, 0,0,0,0,0,0,0,-1),
              c(0,0,0,1,0, 0,0,0,0,0,0,0, 0),
              c(0,1,0,1,0, 0,0,0,0,1,0,0, 0),
              c(0,0,1,1,0, 0,0,0,0,0,0,1, 0),
              c(0,0,0,0,1, 0,0,0,0,0,0,0,-1),
              c(0,1,0,0,1, 0,0,0,0,0,1,0, 0),
              c(0,0,1,0,1, 0,0,0,0,0,0,1, 1),

              c(1,0,0,0,0, 0,0,0,0,0,0,0,-1),
              c(1,1,0,0,0, 1,0,0,0,0,0,0, 0),
              c(1,0,1,0,0, 0,1,0,0,0,0,0, 1),
              c(1,0,0,1,0, 0,0,1,0,0,0,0, 0),
              c(1,1,0,1,0, 1,0,1,0,1,0,0, 0),
              c(1,0,1,1,0, 0,1,1,0,0,0,1, 0),
              c(1,0,0,0,1, 0,0,0,1,0,0,0, 1),
              c(1,1,0,0,1, 1,0,0,1,0,1,0, 0),
              c(1,0,1,0,1, 0,1,0,1,0,0,1,-1))
e96.loglinml1<-loglinML(e96.catdata,X=e96.X1) #associação uniforme comum
e96.loglinml2<-loglinML(e96.catdata,X=e96.X1[,-14]) #sob ausência de interação de 2a.ordem
e96.loglinml3<-loglinML(e96.catdata,X=e96.X1[,-c(8,9,14)]) #AC,BC
e96.loglinml4<-loglinML(e96.catdata,X=e96.X1[,-(10:14)]) #AB,AC
e96.loglinml5<-loglinML(e96.catdata,X=e96.X1[,-c(6,7,14)]) #AB,BC
e96.X2<-rbind(c( 1, 1, 0, 1, 0,  1, 1, 1),
              c( 1, 1, 0, 0, 1,  1, 0, 0),
              c( 1, 1, 0,-1,-1,  1,-1,-1),
              c( 1, 0, 1, 1, 0,  0, 1, 0),
              c( 1, 0, 1, 0, 1,  0, 0, 0),
              c( 1, 0, 1,-1,-1,  0,-1, 0),
              c( 1,-1,-1, 1, 0, -1, 1,-1),
              c( 1,-1,-1, 0, 1, -1, 0, 0),
              c( 1,-1,-1,-1,-1, -1,-1, 1),

```



```

c(-1, 1, -1, 1, 1, -1, -1, 1),
c(-1, -1, -1, -1, 1, 1, 1, -1),
c( 1, 1, -1, -1, 1, 1, 1, 1),
c( 1, -1, -1, 1, 1, -1, -1, -1),
c(-1, 1, 1, -1, -1, 1, -1, -1),
c(-1, -1, 1, 1, -1, -1, 1, 1),
c( 1, 1, -1, -1, -1, -1, 1, -1),
c( 1, -1, -1, 1, -1, 1, -1, 1),
c(-1, 1, 1, -1, 1, -1, -1, 1),
c(-1, -1, 1, 1, 1, 1, 1, -1))
e911.loglinml5<-loglinML(e911.catdata,X=e911.X) #AB,AC,AD,BCD
e911.loglinml6<-loglinML(e911.catdata,X=e911.X[,-4]) #AB,AC,BCD
e911.loglinml7<-loglinML(e911.catdata,X=e911.X[-(3:4)]) #AB,BCD
e911.loglinml8<-loglinML(e911.catdata,X=e911.X[-c(4,8)]) #AB,AC,BC,BD,CD
e911.loglinml9<-loglinML(e911.catdata,X=e911.X[-8]) #AB,AC,AD,BC,BD,CD
round(e911.loglinml9$QvH-e911.loglinml5$QvH,2) #BCD=0|H_2
round(1-pchisq(e911.loglinml9$QvH-e911.loglinml5$QvH,1),2)
round(e911.loglinml6$QvH-e911.loglinml5$QvH,2) #AD=0|H_2
round(1-pchisq(e911.loglinml6$QvH-e911.loglinml5$QvH,1),2)
round(e911.loglinml8$QvH-e911.loglinml6$QvH,2) #BCD=0|H_3
round(1-pchisq(e911.loglinml8$QvH-e911.loglinml6$QvH,1),2)
round(exp(4*(e911.loglinml6$beta[4:6]+e911.loglinml6$beta[7])),2) #RPC 5-8 anos: AD, AC, AB
round(exp(4*(e911.loglinml6$beta[4:6]-e911.loglinml6$beta[7])),2) #RPC 9-12 anos: AD, AC, AB
round(exp(4*e911.loglinml6$beta[3]),2) #RPC BD

```

Exemplo 9.12 (pp.316, 317, 319, 324, 326): *Problema da aterosclerose coronariana*

```

e912.TF<-c(31,17,42,27,55,42,94,104,80,112,70,130,74,188,68,314)
e912.catdata<-readCatdata(TF=e912.TF)
#A,B,C,D, AB,AC,AD,BC,BD,CD, ABC,ABD,ACD,BCD
e912.X<-rbind(c( 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
c( 1, 1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, -1, -1, -1),
c( 1, 1, -1, 1, 1, -1, 1, -1, 1, -1, -1, 1, -1, -1, -1),
c( 1, 1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, 1, 1, 1),
c( 1, -1, 1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1, -1, -1),
c( 1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, -1, 1, -1, 1),
c( 1, -1, -1, 1, -1, -1, 1, 1, -1, -1, 1, -1, -1, 1, 1),
c( 1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1),
c(-1, 1, 1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -1, 1, 1),
c(-1, 1, 1, -1, -1, -1, 1, 1, -1, -1, -1, 1, 1, -1, -1),
c(-1, 1, -1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1),
c(-1, 1, -1, -1, -1, 1, 1, -1, -1, 1, 1, 1, -1, 1, 1),
c(-1, -1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1),
c(-1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, 1, 1),
c(-1, -1, -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, 1, 1, 1),
c(-1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1))
e912.loglinml1<-loglinML(e912.catdata,X=e912.X)
e912.loglinml2<-loglinML(e912.catdata,X=e912.X[,1:10])
e912.loglinml3<-loglinML(e912.catdata,X=e912.X[,1:4])
round(e912.loglinml2$QvH-e912.loglinml1$QvH,2)
round(1-pchisq(e912.loglinml2$QvH-e912.loglinml1$QvH,e912.loglinml2$glH-e912.loglinml1$glH),2)
round(e912.loglinml3$QvH-e912.loglinml2$QvH,2)
round(1-pchisq(e912.loglinml3$QvH-e912.loglinml2$QvH,e912.loglinml3$glH-e912.loglinml2$glH),2)
#continuação p.317
e912.loglinml4<-loglinML(e912.catdata,X=cbind(e912.X,e912.X[,1]*e912.X[,2]*e912.X[,3]*e912.X[,4]))
e912.loglinml5<-loglinML(e912.catdata,X=e912.X[-14])

```

```

e912.loglinml6<-loglinML(e912.catdata,X=e912.X[,-13])
e912.loglinml7<-loglinML(e912.catdata,X=e912.X[,-12])
e912.loglinml8<-loglinML(e912.catdata,X=e912.X[,-11])
e912.loglinml9<-loglinML(e912.catdata,X=e912.X[-c(10:14)])
e912.loglinml10<-loglinML(e912.catdata,X=e912.X[-c(9,11:14)])
e912.loglinml11<-loglinML(e912.catdata,X=e912.X[-c(8,11:14)])
e912.loglinml12<-loglinML(e912.catdata,X=e912.X[-c(7,11:14)])
e912.loglinml13<-loglinML(e912.catdata,X=e912.X[-c(6,11:14)])
e912.loglinml14<-loglinML(e912.catdata,X=e912.X[-c(5,11:14)])
round(c(e912.loglinml14$QvH-e912.loglinml2$QvH,
        e912.loglinml13$QvH-e912.loglinml2$QvH,
        e912.loglinml12$QvH-e912.loglinml2$QvH,
        e912.loglinml11$QvH-e912.loglinml2$QvH,
        e912.loglinml10$QvH-e912.loglinml2$QvH,
        e912.loglinml9$QvH-e912.loglinml2$QvH,
        e912.loglinml8$QvH-e912.loglinml1$QvH,
        e912.loglinml7$QvH-e912.loglinml1$QvH,
        e912.loglinml6$QvH-e912.loglinml1$QvH,
        e912.loglinml5$QvH-e912.loglinml1$QvH,
        e912.loglinml1$QvH),2) #parcial
round(1-pchisq(c(e912.loglinml14$QvH-e912.loglinml2$QvH,
                 e912.loglinml13$QvH-e912.loglinml2$QvH,
                 e912.loglinml12$QvH-e912.loglinml2$QvH,
                 e912.loglinml11$QvH-e912.loglinml2$QvH,
                 e912.loglinml10$QvH-e912.loglinml2$QvH,
                 e912.loglinml9$QvH-e912.loglinml2$QvH,
                 e912.loglinml8$QvH-e912.loglinml1$QvH,
                 e912.loglinml7$QvH-e912.loglinml1$QvH,
                 e912.loglinml6$QvH-e912.loglinml1$QvH,
                 e912.loglinml5$QvH-e912.loglinml1$QvH,
                 e912.loglinml1$QvH),1),2) #parcial
e912.loglinml15<-loglinML(e912.catdata,X=e912.X[,c(1:10,14)])
e912.loglinml16<-loglinML(e912.catdata,X=e912.X[,c(1:10,13)])
e912.loglinml17<-loglinML(e912.catdata,X=e912.X[,c(1:10,12)])
e912.loglinml18<-loglinML(e912.catdata,X=e912.X[,c(1:11)]) #ABC,AD,BD,CD: modelo final
e912.loglinml19<-loglinML(e912.catdata,X=e912.X[,c(1:4,10)])
e912.loglinml20<-loglinML(e912.catdata,X=e912.X[,c(1:4,9)])
e912.loglinml21<-loglinML(e912.catdata,X=e912.X[,c(1:4,8)])
e912.loglinml22<-loglinML(e912.catdata,X=e912.X[,c(1:4,7)])
e912.loglinml23<-loglinML(e912.catdata,X=e912.X[,c(1:4,6)])
e912.loglinml24<-loglinML(e912.catdata,X=e912.X[,c(1:4,5)])
round(c(e912.loglinml3$QvH-e912.loglinml24$QvH,
        e912.loglinml3$QvH-e912.loglinml23$QvH,
        e912.loglinml3$QvH-e912.loglinml22$QvH,
        e912.loglinml3$QvH-e912.loglinml21$QvH,
        e912.loglinml3$QvH-e912.loglinml20$QvH,
        e912.loglinml3$QvH-e912.loglinml19$QvH,
        e912.loglinml2$QvH-e912.loglinml18$QvH,
        e912.loglinml2$QvH-e912.loglinml17$QvH,
        e912.loglinml2$QvH-e912.loglinml16$QvH,
        e912.loglinml2$QvH-e912.loglinml15$QvH,
        e912.loglinml1$QvH),2) #marginal
round(1-pchisq(c(e912.loglinml3$QvH-e912.loglinml24$QvH,
                 e912.loglinml3$QvH-e912.loglinml23$QvH,
                 e912.loglinml3$QvH-e912.loglinml22$QvH,
                 e912.loglinml3$QvH-e912.loglinml21$QvH,

```

```

e912.loglinml3$QvH-e912.loglinml20$QvH,
e912.loglinml3$QvH-e912.loglinml19$QvH,
e912.loglinml2$QvH-e912.loglinml18$QvH,
e912.loglinml2$QvH-e912.loglinml17$QvH,
e912.loglinml2$QvH-e912.loglinml16$QvH,
e912.loglinml2$QvH-e912.loglinml15$QvH,
e912.loglinml1$QvH),1),2) #marginal
#continuação p.326
e9122.TF<-rbind(c(31,17),c(42,27),c(55,42),c(94,104),c(80,112),c(70,130),c(74,188),c(68,314))
e9122.catdata<-readCatdata(TF=e9122.TF)
#ABC,AD,BD,CD
e9122.XL<-rbind(c(1,1,1,1),
               c(1,1,1,0),
               c(1,1,0,1),
               c(1,1,0,0),
               c(1,0,1,1),
               c(1,0,1,0),
               c(1,0,0,1),
               c(1,0,0,0))
e9122.loglinml<-loglinML(e9122.catdata,XL=e9122.XL)
round(exp(e9122.loglinml$beta[2:4]),2)
round(exp(e9122.loglinml$beta[2:4]-qnorm(0.975)*sqrt(diag(e9122.loglinml$Vbeta))[2:4]),2)
round(exp(e9122.loglinml$beta[2:4]+qnorm(0.975)*sqrt(diag(e9122.loglinml$Vbeta))[2:4]),2)
round(kronecker(diag(8),t(c(1,0)))%*%e912.loglinml18$thetaH/kronecker(diag(8),t(c(1,1)))%*%
      e912.loglinml18$thetaH,3) #o mesmo que se encontra nas prob.estim. de summary(e9122.loglinml)

```

Exemplo 10.1 (p.348) / 6.1 (p.149): *Problema da intoxicação de besouros*

```

e101.TF<-cbind(c(6,13,18,28,52,53,61,60),c(59,60,62,56,63,59,62,60))
e101.TF[,2]<-e101.TF[,2]-e101.TF[,1]
e101.catdata<-readCatdata(TF=e101.TF)
e101.XL<-cbind(rep(1,8),c(1.6907,1.7242,1.7552,1.7842,1.8113,1.8369,1.8610,1.8839))
e101.loglinml<-loglinML(e101.catdata,XL=e101.XL)
round(exp(e101.loglinml$beta[2]*log10(2)),2)
round(exp((e101.loglinml$beta[2]+c(-1,1)*qnorm(0.975)*sqrt(e101.loglinml$Vbeta[2,2]))*log10(2)),2)

```

Exemplo 10.2 (p.349) / 6.3 (p.156) / 1.2 (p.4): *Problema do risco de cárie dentária*

```

e102.TF<-c(11,5,0,14,34,7,2,13,11)
e102.catdata<-readCatdata(TF=e102.TF)
e102.B<-rbind(c(1,-1,0),c(0,1,-1))
e102.A<-kronecker(e102.B,e102.B)
e102.XL<-rep(1,4)
e102.loglinml1<-loglinML(e102.catdata,A=e102.A,XL=e102.XL)
e102.X<-rbind(c(0,0,0,0, 1),
              c(1,0,0,0, 0),
              c(0,1,0,0,-1),
              c(0,0,1,0, 0),
              c(1,0,1,0, 0),
              c(0,1,1,0, 0),
              c(0,0,0,1,-1),
              c(1,0,0,1, 0),
              c(0,1,0,1, 1))
e102.loglinml2<-loglinML(e102.catdata,X=e102.X)
round(e102.loglinml1$beta,2)
round(e102.loglinml1$beta+c(-1,1)*qnorm(0.975)*sqrt(e102.loglinml1$Vbeta),2)

```

```
round(exp(e102.loglinml1$beta),2)
round(exp(e102.loglinml1$beta+c(-1,1)*qnorm(0.975)*sqrt(e102.loglinml1$Vbeta)),2)
```

Exemplo 10.3 (p.349) / 6.4 (p.157) / 1.5 (p.5): *Problema do uso do fio dental*

```
e103.TF<-rbind(c(19,5,4, 2),
              c( 5,8,0,17),
              c(11,6,7, 6),
              c( 2,5,1,22))
e103.catdata<-readCatdata(TF=e103.TF)
e103.A<-kronecker(diag(4),t(c(1,-1,-1,1)))
e103.XL<-cbind(rep(1,4),c(1,1,0,0),c(1,0,1,0))
e103.loglinml<-loglinML(e103.catdata,A=e103.A,XL=e103.XL)
round(exp(e103.loglinml$beta),2)
round(exp(e103.loglinml$beta-qnorm(0.975)*sqrt(diag(e103.loglinml$Vbeta))),2)
round(exp(e103.loglinml$beta+qnorm(0.975)*sqrt(diag(e103.loglinml$Vbeta))),2)
round(exp(t(c(0,1,1))%*%e103.loglinml$beta),2)
round(exp(t(c(0,1,1))%*%e103.loglinml$beta+
        c(-1,1)*qnorm(0.975)*sqrt(t(c(0,1,1))%*%e103.loglinml$Vbeta%*%c(0,1,1))),2)
```

Exemplo 10.4 (p.351) / 6.5 (p.160): *Problema da complicação pulmonar*

```
e104.catdata<-readCatdata(TF=cbind(c(737,243,39),c(48,74,21)))
e104.A<-rbind(c(0,-1,0,1,0,0),c(0,0,0,-1,0,1))
e104.loglinml<-loglinML(e104.catdata,A=e104.A,XL=c(1,1))
#A rotina loglinML não ajusta este tipo de modelo log-linear generalizado,
#portanto utiliza-se a funlinWLS para o ajuste por MQG
e104.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e104.catdata,A1=e104.A,X=c(1,1))
e104.loglinwls2<-funlinWLS(model=c("lin","log"),obj=e104.catdata,A1=e104.A,X=c(2,1))
exp(e104.loglinwls2$beta)
exp(e104.loglinwls2$beta+c(-1,1)*qnorm(0.975)*sqrt(e104.loglinwls2$Vbeta))
exp(2*e104.loglinwls2$beta)
exp(2*(e104.loglinwls2$beta+c(-1,1)*qnorm(0.975)*sqrt(e104.loglinwls2$Vbeta)))
```

Exemplo 10.5 (p.353) / 6.6 (p.163) / 1.3 (p.4): *Problema do peso de recém-nascidos*

```
e105.TF<-rbind(c( 2, 11, 31),c( 5, 24, 95),
              c( 3, 32, 91),c( 11, 57, 238),
              c( 15, 58,134),c( 25,105, 445),
              c(130,362,695),c(231,694,2485),
              c( 94,225,340),c(105,339,1053))
e105.catdata<-readCatdata(TF=e105.TF)
e105.A1<-kronecker(diag(10),rbind(c(1,0,0),c(0,1,1),c(0,1,0),c(0,0,1)))
e105.A2<-kronecker(diag(10),kronecker(diag(2),t(c(1,-1))))
e105.X1<-kronecker(rbind(c(1, 1, 0, 0, 0, 1),
                        c(1, 1, 0, 0, 0,-1),
                        c(1, 0, 1, 0, 0, 1),
                        c(1, 0, 1, 0, 0,-1),
                        c(1, 0, 0, 1, 0, 1),
                        c(1, 0, 0, 1, 0,-1),
                        c(1, 0, 0, 0, 1, 1),
                        c(1, 0, 0, 0, 1,-1),
                        c(1,-1,-1,-1,-1, 1),
                        c(1,-1,-1,-1,-1,-1)),diag(2))
e105.X2<-kronecker(rbind(c(1,-2, 1),
                        c(1,-2,-1),
```

```

c(1,-1, 1),
c(1,-1,-1),
c(1, 0, 1),
c(1, 0,-1),
c(1, 1, 1),
c(1, 1,-1),
c(1, 2, 1),
c(1, 2,-1)),diag(2))
e105.funlinwls1<-funlinWLS(model=c("lin","log","lin"),obj=e105.catdata,
A1=e105.A1,A2=e105.A2,X=e105.X1)
e105.funlinwls2<-funlinWLS(model=c("lin","log","lin"),obj=e105.catdata,
A1=e105.A1,A2=e105.A2,X=e105.X2)

e105.TF2<-matrix(c(e105.A1%%c(t(e105.TF))),20,2,byrow=T)
e105.catdata2<-readCatdata(TF=e105.TF2)
e105.A3<-kronecker(diag(20),t(c(1,-1)))
e105.loglinml1<-loglinML(e105.catdata2,A=e105.A3,XL=e105.X1)
e105.loglinml2<-loglinML(e105.catdata2,A=e105.A3,XL=e105.X2)
round(exp(-e105.loglinml2$beta[3:4]),2)
round(exp(-e105.loglinml2$beta[3:4]-qnorm(0.975)*sqrt(diag(e105.loglinml2$Vbeta))[3:4]),2)
round(exp(-e105.loglinml2$beta[3:4]+qnorm(0.975)*sqrt(diag(e105.loglinml2$Vbeta))[3:4]),2)
round(exp(e105.loglinml2$beta[5:6]),2)
round(exp(e105.loglinml2$beta[5:6]-qnorm(0.975)*sqrt(diag(e105.loglinml2$Vbeta))[5:6]),2)
round(exp(e105.loglinml2$beta[5:6]+qnorm(0.975)*sqrt(diag(e105.loglinml2$Vbeta))[5:6]),2)

```

Exemplo 10.6 (p.357) / 6.7 (p.167) / 1.3 (p.4): *Problema do peso de recém-nascidos*

```

e106.TF<-rbind(c( 2, 11, 31),c( 5, 24, 95),
c( 3, 32, 91),c( 11, 57, 238),
c( 15, 58,134),c( 25,105, 445),
c(130,362,695),c(231,694,2485),
c( 94,225,340),c(105,339,1053))
e106.catdata<-readCatdata(TF=e106.TF)
e106.A1<-kronecker(diag(10),rbind(c(1,0,0),c(0,1,1),c(1,1,0),c(0,0,1)))
e106.A2<-kronecker(diag(10),kronecker(diag(2),t(c(1,-1))))
e106.X2<-rbind(c(1,0,-2, 0, 1),
c(0,1,-2, 0, 1),
c(1,0, 0,-2,-1),
c(0,1, 0,-2,-1),
c(1,0,-1, 0, 1),
c(0,1,-1, 0, 1),
c(1,0, 0,-1,-1),
c(0,1, 0,-1,-1),
c(1,0, 0, 0, 1),
c(0,1, 0, 0, 1),
c(1,0, 0, 0,-1),
c(0,1, 0, 0,-1),
c(1,0, 1, 0, 1),
c(0,1, 1, 0, 1),
c(1,0, 0, 1,-1),
c(0,1, 0, 1,-1),
c(1,0, 2, 0, 1),
c(0,1, 2, 0, 1),
c(1,0, 0, 2,-1),
c(0,1, 0, 2,-1))
e106.X1<-cbind(e106.X2[,1:2],e106.X2[,3]+e106.X2[,4],e106.X2[,5])

```

```

#Estes resultados (MQG) não são exatamente iguais ao do livro (MV)
e106.funlinwls1<-funlinWLS(model=c("lin","log","lin"),obj=e106.catdata,
  A1=e106.A1,A2=e106.A2,X=e106.X1)
e106.funlinwls2<-funlinWLS(model=c("lin","log","lin"),obj=e106.catdata,
  A1=e106.A1,A2=e106.A2,X=e106.X2)
round(e106.funlinwls1$QwH-e106.funlinwls2$QwH,2)
round(1-pchisq(e106.funlinwls1$QwH-e106.funlinwls2$QwH,1),2)

```

Exemplo 10.8 (pp.360, 367): *Problema da susceptibilidade a malária cerebral*

```

e108.TF<-rbind(c(35,10),
  c(25,23),
  c(27,21),
  c( 9,40))
e108.catdata<-readCatdata(TF=e108.TF)
e108.X<-rbind(c(1,0,0),
  c(1,0,1),
  c(1,1,0),
  c(1,1,1))
e108.linml<-linML(e108.catdata,X=e108.X) #aditivo
e108.A<-kronecker(diag(4),t(c(1,0)))
#Este resultado (MQG) não é exatamente igual ao do livro (MV)
e108.funlinwls<-funlinWLS(model=c("lin","log"),obj=e108.catdata,
  A1=e108.A,X=e108.X) #multiplicativo
e108.loglinml<-loglinML(e108.catdata,XL=e108.X) #logístico

```

Exemplo 11.1 (p.376) / 7.1 (p.202) / 1.7 (p.11): *Problema do grupo sanguíneo ABO*

```

e111.TF<-cbind(4219,890,313,4578)
e111.catdata<-readCatdata(TF=e111.TF)
e111.A1<-rbind(diag(4),
  c(0,0,0,0.5))
e111.A2<-rbind(c(1,0,0 ,1,0),
  c(0,1,0 ,1,0),
  c(0,0,0.5,0,1))
e111.A3<-cbind(1,1,-2)
e111.funlinwls<-funlinWLS(model=c("lin","log","lin","exp","lin","log"),
  obj=e111.catdata,A1=e111.A1,A2=e111.A2,A3=e111.A3,X=1)

e111.TF<-rbind(c(4219,890,313,4578),
  c( 96, 18, 5, 181),
  c( 214, 39, 13, 298))
e111.catdata<-readCatdata(TF=e111.TF)
e111.A1<-kronecker(diag(3),rbind(diag(4),
  c(0,0,0,0.5)))
e111.A2<-kronecker(diag(3),rbind(c(1,0,0 ,1,0),
  c(0,1,0 ,1,0),
  c(0,0,0.5,0,1)))
e111.A3<-kronecker(diag(3),cbind(1,1,-2))
e111.funlinwls<-funlinWLS(model=c("lin","log","lin","exp","lin","log"),
  obj=e111.catdata,A1=e111.A1,A2=e111.A2,A3=e111.A3,X=diag(3))
waldTest(e111.funlinwls,diag(3))

```

Exemplo 11.2 (p.382) / 8.1 (p.228) / 3.1 (p.47): *Problema da intenção de voto*

```

e112.TF<-c(192,1,5,2,146,5,11,12,71)

```

```

e112.catdata<-readCatdata(TF=e112.TF)
e112.U<-rbind(c(0,-1, 0,1,0, 0,0,0),
              c(0, 0,-1,0,0, 0,1,0),
              c(0, 0, 0,0,0,-1,0,1))
e112.X<-rbind(c(1,0,0,0,0),
              c(0,1,0,0,0),
              c(0,0,1,0,0),
              c(0,1,0,0,0),
              c(0,0,0,1,0),
              c(0,0,0,0,1),
              c(0,0,1,0,0),
              c(0,0,0,0,1))
e112.linwls1<-funlinWLS(model="lin",obj=e112.catdata,U=e112.U) #simetria
e112.linwls2<-funlinWLS(model="lin",obj=e112.catdata,X=e112.X) #simetria

```

Exemplo 11.3 (p.383) / 8.2 (p.233) / 3.2 (p.49) / 1.2 (p.4): *Problema do risco de cárie dentária*

```

e113.TF<-c(11,5,0,14,34,7,2,13,11)
e113.catdata<-readCatdata(TF=e113.TF)
e113.U<-rbind(c(0, 1,1,-1,0,0,-1, 0),
              c(0,-1,0, 1,0,1, 0,-1))
e113.X<-rbind(c(1, 0, 0,0,0,0,0),
              c(0, 1, 0,0,0,0,0),
              c(0,-1, 1,0,1,0,0),
              c(0, 0, 1,0,0,0,0),
              c(0, 0, 0,1,0,0,0),
              c(0, 1,-1,0,0,0,1),
              c(0, 0, 0,0,1,0,0),
              c(0, 0, 0,0,0,1))
e113.linwls1<-funlinWLS(model="lin",obj=e113.catdata,U=e113.U) #homogeneidade marginal (HM)
e113.linwls2<-funlinWLS(model="lin",obj=e113.catdata,X=e113.X) #homogeneidade marginal (HM)
e113.A<-rbind(c(1,1,1,0,0,0,0,0,0),
              c(0,0,0,1,1,1,0,0,0),
              c(1,0,0,1,0,0,1,0,0),
              c(0,1,0,0,1,0,0,1,0) )
e113.U2<-rbind(c(1,0,-1, 0),
               c(0,1, 0,-1))
e113.X2<-rbind(c(1,0),
               c(0,1),
               c(1,0),
               c(0,1) )
e113.linwls3<-funlinWLS(model="lin",obj=e113.catdata,A1=e113.A,U=e113.U2) #HM
e113.linwls4<-funlinWLS(model="lin",obj=e113.catdata,A1=e113.A,X=e113.X2) #HM

```

Exemplo 11.4 (p.384) / 8.3 (p.236) / 3.3 (p.50) / 1.9 (p.12): *Problema do tamanho da ninhada*

```

e114.TF<-rbind(c(10,21, 96,23),
               c( 4, 6, 28, 8),
               c( 9, 7, 58, 7),
               c( 8,19, 44, 1),
               c( 5,17, 56, 1),
               c( 1, 5, 20, 2),
               c(22,95,103, 4),
               c(18,49, 62, 0),

```

```

      c( 4,12, 16, 2))
e114.catdata<-readCatdata(TF=e114.TF)
e114.A<-kronecker(diag(9),t(c(0,1,2,3)))
e114.X<-rbind(c(1, 1, 0, 1, 0, 1, 0, 0, 0),
             c(1, 1, 0, 0, 1, 0, 1, 0, 0),
             c(1, 1, 0,-1,-1, -1,-1, 0, 0),
             c(1, 0, 1, 1, 0, 0, 0, 1, 0),
             c(1, 0, 1, 0, 1, 0, 0, 0, 1),
             c(1, 0, 1,-1,-1, 0, 0,-1,-1),
             c(1,-1,-1, 1, 0, -1, 0,-1, 0),
             c(1,-1,-1, 0, 1, 0,-1, 0,-1),
             c(1,-1,-1,-1,-1, 1, 1, 1, 1))
e114.linwls1<-funlinWLS(model="lin",obj=e114.catdata,A1=e114.A,X=e114.X)
waldTest(e114.linwls1,cbind(matrix(0,4,5),diag(4)))
waldTest(e114.linwls1,cbind(rep(0,2),diag(2),matrix(0,2,6)))
waldTest(e114.linwls1,cbind(matrix(0,2,3),diag(2),matrix(0,2,4)))
e114.linwls2<-funlinWLS(model="lin",obj=e114.catdata,A1=e114.A,X=e114.X[,1:5])
waldTest(e114.linwls2,cbind(rep(0,2),diag(2),matrix(0,2,2)))
waldTest(e114.linwls2,cbind(matrix(0,2,3),diag(2)))
e114.linwls3<-funlinWLS(model="lin",obj=e114.catdata,A1=e114.A,X=e114.X[,1:3])

```

Exemplo 11.5 (p.388) / 9.1 (p.263): *Problema da anemia*

```

e115.TF<-c(3,25,32,68)
e115.catdata<-readCatdata(TF=e115.TF)
e115.U<-c(1,-1,-1,1)
e115.X<-rbind(c(0,0),
             c(0,1),
             c(1,0),
             c(1,1))
e115.X2<-rbind(c(0,0,0),
             c(0,1,0),
             c(1,0,0),
             c(1,1,1))
e115.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e115.catdata,U=e115.U) #independência
e115.loglinwls2<-funlinWLS(model=c("lin","log"),obj=e115.catdata,X=e115.X) #independência
e115.loglinwls3<-funlinWLS(model=c("lin","log"),obj=e115.catdata,X=e115.X2) #modelo saturado
e115.loglinwls4<-funlinWLS(model=c("lin","log"),obj=e115.catdata,A1=c(1,-1,-1,1),
  XL=1) #modelo saturado
round(e115.loglinwls4$beta+c(-1,1)*qnorm(0.975)*sqrt(e115.loglinwls4$Vbeta),3)
round(exp(e115.loglinwls4$beta),3) #razão de chances
round(exp(e115.loglinwls4$beta+c(-1,1)*qnorm(0.975)*sqrt(e115.loglinwls4$Vbeta)),3)

```

Exemplo 11.6 (p.388) / 9.2 (p.267): *Problema da acuidade visual*

```

e116.TF<-c(1520,266,124,66, 234,1512,432,78, 117,362,1772,205, 36,82,179,492)
e116.catdata<-readCatdata(TF=e116.TF)
e116.X1<-rbind(c(1,0,0,0,0,0,0,0,0),
             c(0,1,0,0,0,0,0,0,0),
             c(0,0,1,0,0,0,0,0,0),
             c(0,0,0,1,0,0,0,0,0),

             c(0,1,0,0,0,0,0,0,0),
             c(0,0,0,0,1,0,0,0,0),
             c(0,0,0,0,0,1,0,0,0),
             c(0,0,0,0,0,0,1,0,0))

```



```

c(0,0,1,0,0,0,0,0,0),
c(0,0,0,0,0,1,0,0,0),
c(0,0,0,0,0,0,0,1,0),
c(0,0,0,0,0,0,0,0,1),

c(0,0,0,1,0,0,0,0,0),
c(0,0,0,0,0,0,1,0,0),
c(0,0,0,0,0,0,0,0,1)
e116.linwls1<-funlinWLS(model="lin",obj=e116.catdata,X=e116.X1) #simetria em formulação linear
e116.A1<-rbind(c(0,1,0,0, -1,0,0,0, 0, 0,0,0, 0, 0, 0,0),
c(0,0,1,0, 0,0,0,0, -1, 0,0,0, 0, 0, 0,0),
c(0,0,0,1, 0,0,0,0, 0, 0,0,0, -1, 0, 0,0),
c(0,0,0,0, 0,0,1,0, 0,-1,0,0, 0, 0, 0,0),
c(0,0,0,0, 0,0,0,1, 0, 0,0,0, 0,-1, 0,0),
c(0,0,0,0, 0,0,0,0, 0, 0,0,1, 0, 0,-1,0))
e116.linwls2<-funlinWLS(model="lin",obj=e116.catdata,U=e116.A1[,1:15]) #simetria em form.linear
#u_1,u_2,u_3, u_{11},u_{12},u_{13},u_{22},u_{23},u_{33}
e116.X2<-rbind(c( 2, 0, 0, 1, 0, 0, 0, 0, 0),
c( 1, 1, 0, 0, 1, 0, 0, 0, 0),
c( 1, 0, 1, 0, 0, 1, 0, 0, 0),
c( 0,-1,-1, -1,-1,-1, 0, 0, 0),

c( 1, 1, 0, 0, 1, 0, 0, 0, 0),
c( 0, 2, 0, 0, 0, 0, 1, 0, 0),
c( 0, 1, 1, 0, 0, 0, 0, 1, 0),
c(-1, 0,-1, 0,-1, 0,-1,-1, 0),

c( 1, 0, 1, 0, 0, 1, 0, 0, 0),
c( 0, 1, 1, 0, 0, 0, 0, 1, 0),
c( 0, 0, 2, 0, 0, 0, 0, 0, 1),
c(-1,-1, 0, 0, 0,-1, 0,-1,-1),

c( 0,-1,-1, -1,-1,-1, 0, 0, 0),
c(-1, 0,-1, 0,-1, 0,-1,-1, 0),
c(-1,-1, 0, 0, 0,-1, 0,-1,-1),
c(-2,-2,-2, 1, 2, 2, 1, 2, 1)) #análogo à matriz da pág.71
e116.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e116.catdata,X=e116.X2) #simetr.form.log-lin.
e116.loglinwls2<-funlinWLS(model=c("lin","log"),obj=e116.catdata,U=e116.A1) #simetr.form.log-lin.
e116.A2<-rbind(cbind(kronecker(diag(3),t(rep(1,4))),matrix(0,3,4)),
kronecker(t(rep(1,4)),cbind(diag(3),rep(0,3))))
e116.linwls3<-funlinWLS(model="lin",obj=e116.catdata,A1=e116.A2,X=kronecker(rep(1,2),diag(3)))#HM
#u_1^A,u_2^A,u_3^A, u_1^B,u_2^B,u_3^B, u_{11},u_{12},u_{13},u_{22},u_{23},u_{33}
e116.X3<-rbind(c( 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0),
c( 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0),
c( 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0),
c( 1, 0, 0, -1,-1,-1, -1,-1,-1, 0, 0, 0),

c( 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0),
c( 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0),
c( 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0),
c( 0, 1, 0, -1,-1,-1, 0,-1, 0,-1,-1, 0),

c( 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0),
c( 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0),
c( 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1),

```

```

c( 0, 0, 1, -1,-1,-1, 0, 0,-1, 0,-1,-1),

c(-1,-1,-1, 1, 0, 0, -1,-1,-1, 0, 0, 0),
c(-1,-1,-1, 0, 1, 0, 0,-1, 0,-1,-1, 0),
c(-1,-1,-1, 0, 0, 1, 0, 0,-1, 0,-1,-1),
c(-1,-1,-1, -1,-1,-1, 1, 2, 2, 1, 2, 1))
#e92.X2<-cbind(e92.X3[,1]+e92.X3[,4],e92.X3[,2]+e92.X3[,5],e92.X3[,3]+e92.X3[,6],e92.X3[,7:12])
cbind(diag(15),rep(-1,15))%*%e116.X3 #matriz X_G do livro
e116.loglinwls3<-funlinWLS(model=c("lin","log"),obj=e116.catdata,X=e116.X3) #quasi-simetria
#u_1,u_2,u_3, u_{11},u_{12},u_{13},u_{22},u_{23},u_{33},gamma
e116.X4<-rbind(c( 2, 0, 0, 1, 0, 0, 0, 0, 0, 0),
c( 1, 1, 0, 0, 1, 0, 0, 0, 0, 1),
c( 1, 0, 1, 0, 0, 1, 0, 0, 0, 1),
c( 0,-1,-1, -1,-1,-1, 0, 0, 0, 1),

c( 1, 1, 0, 0, 1, 0, 0, 0, 0, 0),
c( 0, 2, 0, 0, 0, 0, 1, 0, 0, 0),
c( 0, 1, 1, 0, 0, 0, 0, 1, 0, 1),
c(-1, 0,-1, 0,-1, 0,-1,-1, 0, 1),

c( 1, 0, 1, 0, 0, 1, 0, 0, 0, 0),
c( 0, 1, 1, 0, 0, 0, 0, 1, 0, 0),
c( 0, 0, 2, 0, 0, 0, 0, 0, 1, 0),
c(-1,-1, 0, 0, 0,-1, 0,-1,-1, 1),

c( 0,-1,-1, -1,-1,-1, 0, 0, 0, 0),
c(-1, 0,-1, 0,-1, 0,-1,-1, 0, 0),
c(-1,-1, 0, 0, 0,-1, 0,-1,-1, 0),
c(-2,-2,-2, 1, 2, 2, 1, 2, 1, 0))
e116.loglinwls4<-funlinWLS(model=c("lin","log"),obj=e116.catdata,X=e116.X4) #simetr.condicional
e116.loglinwls5<-funlinWLS(model=c("lin","log"),obj=e116.catdata,A1=e116.A1,XL=rep(1,6))#idem

```

Exemplo 11.7 (p.390) / 9.3 (p.269) / 1.6 (p.11): *Problema dos defeitos de fibras têxteis*

```

e117.TF<-rbind(c(28,40,68),
c( 5,21,49),
c( 1, 4,15))
e117.catdata<-readCatdata(TF=e117.TF)
e117.A<-kronecker(diag(3),cbind(diag(2),rep(-1,2)))
e117.X1<-rbind(c(1,0,0,0),
c(0,1,0,0),
c(1,0,2,0),
c(0,1,1,0),
c(1,0,0,2),
c(0,1,0,1))
e117.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e117.catdata,A1=e117.A,
XL=e117.X1) #efeito de linha
round(exp(-e117.loglinwls1$beta[3:4]),2)
round(exp(e117.loglinwls1$beta[3]-e117.loglinwls1$beta[4]),2)
waldTest(obj=e117.loglinwls1,C=cbind(0*diag(2),diag(2)))
e117.X2<-rbind(c(1,0,0),
c(0,1,0),
c(1,0,2),
c(0,1,1),
c(1,0,4),

```

```

c(0,1,2))
e117.loglinwls2<-funlinWLS(model=c("lin","log"),obj=e117.catdata,A1=e117.A,
XL=e117.X2) #associação uniforme
round(exp(-e117.loglinwls2$beta[3]),2)
round(exp(-e117.loglinwls2$beta[3])*sqrt(e117.loglinwls2$Vbeta[3,3]),2)
e117.A2<-rbind(c(1,-1, 0,-1, 1, 0, 0, 0,0),
c(0, 1,-1, 0,-1, 1, 0, 0,0),
c(0, 0, 0, 1,-1, 0,-1, 1,0),
c(0, 0, 0, 0, 1,-1, 0,-1,1))
e117.loglinwls3<-funlinWLS(model=c("lin","log"),obj=e117.catdata,A1=e117.A2,
XL=rep(1,4)) #associação uniforme

```

Exemplo 11.8 (p.392) / 10.2 (p.349) / 6.3 (p.156) / 1.2 (p.4): *Problema do risco de cárie dentária*

```

e118.TF<-c(11,5,0,14,34,7,2,13,11)
e118.catdata<-readCatdata(TF=e118.TF)
e118.B<-rbind(c(1,-1,0),c(0,1,-1))
e118.A<-kronecker(e118.B,e118.B)
e118.loglinwls1a<-funlinWLS(model=c("lin","log"),
obj=e118.catdata,A1=e118.A,XL=rep(1,4)) #subst.padrão de freq=0 por 1/[9*97]=0.001145475
e118.loglinwls1b<-funlinWLS(model=c("lin","log"),
obj=readCatdata(TF=c(11,5,1/(9*97),14,34,7,2,13,11)),A1=e118.A,XL=rep(1,4)) #parecido
e118.loglinwls2a<-funlinWLS(model=c("lin","log"),
obj=e118.catdata,A1=e118.A,XL=rep(1,4),zeroN=1/2) #subst.freq=0 por 1/2
e118.loglinwls2b<-funlinWLS(model=c("lin","log"),
obj=readCatdata(TF=c(11,5,1/2,14,34,7,2,13,11)),A1=e118.A,XL=rep(1,4)) #parecido
e118.loglinwls3a<-funlinWLS(model=c("lin","log"),
obj=e118.catdata,A1=e118.A,XL=rep(1,4),zeroN=1/16) #subst.freq=0 por 1/16
e118.loglinwls3b<-funlinWLS(model=c("lin","log"),
obj=readCatdata(TF=c(11,5,1/16,14,34,7,2,13,11)),A1=e118.A,XL=rep(1,4)) #parecido
e118.loglinwls4 <-funlinWLS(model=c("lin","log"),obj=readCatdata(TF=1/2+
c(11,5,0,14,34,7,2,13,11)),A1=e118.A,XL=rep(1,4)) #soma 1/2 a todas as frequências
#ao deixar que a rotina substitua os zeros amostrais pelo valor contido em zeroN, a
#substituição será realizada apenas onde for necessário (p/obter a matriz de covariâncias das
#proporções), enquanto que ao substituir diretamente em TF, todas as quantidades serão afetadas

```

Exemplo 11.9 (p.393) / 10.3 (p.349) / 6.4 (p.157) / 1.5 (p.5): *Problema do uso do fio dental*

```

e119.TF<-rbind(c(19,5,4, 2),c( 5,8,0,17),c(11,6,7, 6),c( 2,5,1,22))
e119.catdata<-readCatdata(TF=e119.TF)
e119.A<-kronecker(diag(4),t(c(1,-1,-1,1)))
e119.XL<-cbind(rep(1,4),c(1,1,0,0),c(0,1,0,1))
e119.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e119.catdata,A1=e119.A,XL=e119.XL)
e119.loglinwls2<-funlinWLS(model=c("lin","log"),obj=readCatdata(TF=
rbind(c(19,5,4,2),c(5,8,1/2,17),c(11,6,7,6),c(2,5,1,22))),A1=e119.A,XL=e119.XL) #livro
e119.loglinwls3<-funlinWLS(model=c("lin","log"),obj=readCatdata(TF=
rbind(c(19,5,4,2),c(5,8,1/16,17),c(11,6,7,6),c(2,5,1,22))),A1=e119.A,XL=e119.XL) #livro
e119.loglinwls4<-funlinWLS(model=c("lin","log"),obj=readCatdata(TF=
1/2+rbind(c(19,5,4,2),c(5,8,0,17),c(11,6,7,6),c(2,5,1,22))),A1=e119.A,XL=e119.XL) #livro
e119.loglinwls5<-funlinWLS(model=c("lin","log"),obj=readCatdata(TF=
rbind(c(19,5,4,2),1/2+c(5,8,0,17),c(11,6,7,6),c(2,5,1,22))),A1=e119.A,
XL=e119.XL) #adic.1/2 apenas às celas da subpopulação que teve 0
e119.loglinwls6<-funlinWLS(model=c("lin","log"),obj=readCatdata(TF=
1/2+rbind(c(19,5,4,2),c(5,8,0,17),c(11,6,7,6),c(2,5,1,22))),A1=e119.A,XL=e119.XL[,-2]) #livro

```

```

round(exp(e119.loglinwls6$beta),2)
round(exp(e119.loglinwls6$beta-qnorm(0.975)*sqrt(diag(e119.loglinwls6$Vbeta))),2)
round(exp(e119.loglinwls6$beta+qnorm(0.975)*sqrt(diag(e119.loglinwls6$Vbeta))),2)

```

Exemplo 11.10 (p.395) / 10.4 (p.351) / 6.5 (p.160): *Problema da complicação pulmonar*

```

e1110.catdata<-readCatdata(TF=cbind(c(737,243,39),c(48,74,21)))
e1110.A<-rbind(c(0,-1,0,1,0,0),c(0,-1,0,0,0,1))
e1110.loglinwls1<-funlinWLS(model=c("lin","log"),obj=e1110.catdata,A1=e1110.A,X=diag(2))
e1110.loglinwls2<-funlinWLS(model=c("lin","log"),obj=e1110.catdata,A1=e1110.A,X=c(1,1))
round(exp(e1110.loglinwls1$beta),2)
round(exp(e1110.loglinwls1$beta-1*qnorm(0.975)*sqrt(diag(e1110.loglinwls1$Vbeta))),2)
round(exp(e1110.loglinwls1$beta+1*qnorm(0.975)*sqrt(diag(e1110.loglinwls1$Vbeta))),2)

```

Exemplo 11.11 (p.395) / 10.5 (p.353) / 6.6 (p.163) / 1.3 (p.4): *Problema do peso de recém-nascidos*

```

e1111.TF<-rbind(c( 2, 11, 31),c( 5, 24, 95),
               c( 3, 32, 91),c( 11, 57, 238),
               c( 15, 58,134),c( 25,105, 445),
               c(130,362,695),c(231,694,2485),
               c( 94,225,340),c(105,339,1053))
e1111.catdata<-readCatdata(TF=e1111.TF)
e1111.A1<-kronecker(diag(10),rbind(c(1,0,0),c(0,1,1),c(0,1,0),c(0,0,1)))
e1111.A2<-kronecker(diag(10),kronecker(diag(2),t(c(1,-1))))
e1111.X1<-kronecker(rbind(c(1,0,0,0,0,0),
                          c(1,0,0,0,0,1),
                          c(1,1,0,0,0,0),
                          c(1,1,0,0,0,1),
                          c(1,0,1,0,0,0),
                          c(1,0,1,0,0,1),
                          c(1,0,0,1,0,0),
                          c(1,0,0,1,0,1),
                          c(1,0,0,0,1,0),
                          c(1,0,0,0,1,1)),diag(2))
e1111.funlinwls1<-funlinWLS(model=c("lin","log","lin"),obj=e1111.catdata,
  A1=e1111.A1,A2=e1111.A2,X=e1111.X1)
e1111.X2<-kronecker(rbind(c(1,0,0),
                          c(1,0,1),
                          c(1,1,0),
                          c(1,1,1),
                          c(1,2,0),
                          c(1,2,1),
                          c(1,3,0),
                          c(1,3,1),
                          c(1,4,0),
                          c(1,4,1)),diag(2))
e1111.funlinwls2<-funlinWLS(model=c("lin","log","lin"),obj=e1111.catdata,
  A1=e1111.A1,A2=e1111.A2,X=e1111.X2)
waldTest(e1111.funlinwls2,c(0,0,1,-1,0,0))
waldTest(e1111.funlinwls2,c(0,0,0,0,1,-1))
round(exp(e1111.funlinwls2$beta),2)
round(exp(e1111.funlinwls2$beta-qnorm(0.975)*sqrt(diag(e1111.funlinwls2$Vbeta))),2)
round(exp(e1111.funlinwls2$beta+qnorm(0.975)*sqrt(diag(e1111.funlinwls2$Vbeta))),2)
e1111.funlinwls3<-funlinWLS(model=c("lin","log","lin"),obj=e1111.catdata,A1=e1111.A1,

```

```

A2=e1111.A2,X=cbind(e1111.X2[,1:2],e1111.X2[,3]+e1111.X2[,4],e1111.X2[,5]+e1111.X2[,6]))
round(exp(e1111.funlinwls3$beta),2)
round(exp(e1111.funlinwls3$beta-qnorm(0.975)*sqrt(diag(e1111.funlinwls3$Vbeta))),2)
round(exp(e1111.funlinwls3$beta+qnorm(0.975)*sqrt(diag(e1111.funlinwls3$Vbeta))),2)
e1111.A3<-kronecker(diag(10),rbind(c(1,0,0),c(0,1,1),c(1,1,0),c(0,0,1)))
e1111.funlinwls4<-funlinWLS(model=c("lin","log","lin"),obj=e1111.catdata,A1=e1111.A3,
A2=e1111.A2,X=cbind(e1111.X2[,1:2],e1111.X2[,3]+e1111.X2[,4],e1111.X2[,5]+e1111.X2[,6]))
round(exp(e1111.funlinwls4$beta),2)
round(exp(e1111.funlinwls4$beta-qnorm(0.975)*sqrt(diag(e1111.funlinwls4$Vbeta))),2)
round(exp(e1111.funlinwls4$beta+qnorm(0.975)*sqrt(diag(e1111.funlinwls4$Vbeta))),2)

```

Exemplo 11.12 (p.399) / 6.8 (p.169) / 1.2 (p.4): *Problema do risco de cárie dentária*

```

e1112.TF<-c(11,5,0,14,34,7,2,13,11)
e1112.catdata<-readCatdata(TF=e1112.TF)
e1112.A1<-rbind(
  c(rep(c(1,0,0,0),2),1),
  rep(1,9),
  kronecker(diag(3),t(rep(1,3))),
  kronecker(t(rep(1,3)),diag(3)) )
e1112.A2<-rbind(
  cbind(diag(2),matrix(0,2,6)),
  cbind(matrix(0,3,2),kronecker(t(rep(1,2)),diag(3))) )
e1112.A3<-cbind( c(1,0),c(1,1),-c(2,1))*%t(rep(1,3)) )
e1112.A4<-t(c(1,-1))
e1112.kappa<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
  obj=e1112.catdata,A1=e1112.A1,A2=e1112.A2,A3=e1112.A3,A4=e1112.A4,PI1=-1,X=1)
round(pnorm((e1112.kappa$beta-0.35)/sqrt(e1112.kappa$Vbeta)),3)
#equivalente a round((1-pchisq(waldTest(obj=e1112.kappa,C=1,C0=0.35)$QwH,1))/2,3)
round(e1112.kappa$beta+c(-1,1)*qnorm(0.975)*sqrt(e1112.kappa$Vbeta),3)
#kappa ponderado (Spitzer, Cohen, Fleiss e Endicott, 1967)
W1<-c(1,0.75,0,0.75,1,0.75,0,0.75,1) #pesos quadráticos (Fleiss e Cohen, 1973)
W2<-c(1,0.5,0,0.5,1,0.5,0,0.5,1) #pesos absolutos (Cicchetti e Allison, 1971)
e1112.w1A1<-rbind(
  t(W1),
  rep(1,9),
  kronecker(diag(3),t(rep(1,3))),
  kronecker(t(rep(1,3)),diag(3)) )
e1112.w2A1<-rbind(
  t(W2),
  rep(1,9),
  kronecker(diag(3),t(rep(1,3))),
  kronecker(t(rep(1,3)),diag(3)) )
e1112.wA2<-rbind(
  cbind(diag(2),matrix(0,2,6)),
  cbind(matrix(0,9,2), cbind(kronecker(diag(3),rep(1,3)) , kronecker(rep(1,3),diag(3))) ) )
e1112.w1A3<-cbind( c(1,0),c(1,1),kronecker(-c(2,1),t(W1)) )
e1112.w2A3<-cbind( c(1,0),c(1,1),kronecker(-c(2,1),t(W2)) )
e1112.kappaw1<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
  obj=e1112.catdata,A1=e1112.w1A1,A2=e1112.wA2,A3=e1112.w1A3,A4=e1112.A4,PI1=-1,X=1)
e1112.kappaw2<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
  obj=e1112.catdata,A1=e1112.w2A1,A2=e1112.wA2,A3=e1112.w2A3,A4=e1112.A4,PI1=-1,X=1)

```

Exemplo 11.13 (p.399): *Problema da poluição por petróleo*

```

e1113.TF<-rbind(

```

```

c(2,0,4,0,6,0,3,0,1,0,1,0,15),
c(0,0,1,0,0,0,2,0,2,0,5,0,22),
c(0,0,2,0,1,0,3,0,0,0,5,0,21),
c(0,0,2,0,1,0,2,0,0,0,6,0,21),
c(1,0,4,0,2,0,6,0,1,0,0,0,18),
c(0,0,1,0,0,0,0,0,0,0,1,0,30),
c(0,0,1,0,0,0,3,0,1,0,0,1,26),
c(0,0,0,0,2,0,0,0,0,1,1,0,28))
e1113.catdata<-readCatdata(TF=e1113.TF)
e1113.A1<-diag(8)%x%rbind(c(rep(c(1,0),6),0),c(rep((0:5)+0.5,rep(2,6)),6))
e1113.A2<-diag(8)%x%t(c(1,-1))
e1113.X<-cbind(rep(1,8),c(0,1)%x%rep(1,4),c(1,1)%x%rbind(rep(0,3),diag(3)),
  rbind(matrix(0,5,3),diag(3)))
e1113.X2<-cbind(e1113.X[,1:2],rep(c(1,0,0,0),2))
#Análises das log-taxas de mortalidades
e1113.ltm1<-funlinWLS(model=c("lin","log","lin"),obj=e1113.catdata,
  A1=e1113.A1,A2=e1113.A2,X=e1113.X) #modelo saturado
waldTest(e1113.ltm1,cbind(matrix(0,3,5),diag(3)))
e1113.ltm2<-funlinWLS(model=c("lin","log","lin"),obj=e1113.catdata,
  A1=e1113.A1,A2=e1113.A2,X=e1113.X[,-(6:8)]) #modelo reduzido
e1113.ltm3<-funlinWLS(model=c("lin","log","lin"),obj=e1113.catdata,
  A1=e1113.A1,A2=e1113.A2,X=e1113.X2) #modelo reduzido2
#Análises das taxas de mortalidades
e1113.tm1<-funlinWLS(model=c("exp","lin","log","lin"),obj=e1113.catdata,
  A1=e1113.A1,A2=e1113.A2,X=e1113.X) #modelo saturado
waldTest(e1113.tm1,cbind(matrix(0,3,5),diag(3)))
e1113.tm2<-funlinWLS(model=c("exp","lin","log","lin"),obj=e1113.catdata,
  A1=e1113.A1,A2=e1113.A2,X=e1113.X[,-(6:8)]) #modelo reduzido
e1113.tm3<-funlinWLS(model=c("exp","lin","log","lin"),obj=e1113.catdata,
  A1=e1113.A1,A2=e1113.A2,X=e1113.X2) #modelo reduzido2
round(e1113.tm3$FH[6],3)
round(e1113.tm3$FH[6]+c(-1,1)*qnorm(0.975)*sqrt(e1113.tm3$VFH[6,6]),3)
round(-e1113.tm3$beta[2],3)
round(-e1113.tm3$beta[2]+c(-1,1)*qnorm(0.975)*sqrt(e1113.tm3$Vbeta[2,2]),3)
round(e1113.tm3$beta[3],3)
round(e1113.tm3$beta[3]+c(-1,1)*qnorm(0.975)*sqrt(e1113.tm3$Vbeta[3,3]),3)

```

Exemplo 12.1 (pp.419, 427, 442): *Problema da infecção urinária*

```

e121.raw<-data.frame(
  inicial=c(1,2,1,2,2,2,2,1,3,2,2,1,2,2,2,2,2,1,1,2,1,3,3,2,2,
    2,2,3,2,2,3,0,1,1,1,1,0,0,1,2,1,1,2,2,2,2,3,3,2,3),
  dias14 =c(0,0,0,0,1,2,2,1,0,1,1,1,0,0,1,1,1,0,0,0,1,1,0,1,0,
    0,3,0,2,0,2,0,1,0,0,1,0,0,0,0,1,0,2,2,0,2,1,0,1,0),
  dias21 =c(0,0,0,0,1,2,2,1,0,2,3,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0,
    0,3,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,2,2,1,0,0,0,1,0))
table(e121.raw[,1])
table(e121.raw[,2])
table(e121.raw[,3])
e121.TF<-c(table(e121.raw[,3:1]))
e121.catdata<-readCatdata(TF=e121.TF)
e121.v1<-c(0,0,1,1)
e121.A1<-rbind(e121.v1%x%rep(1,16) , rep(1,4)%x%e121.v1%x%rep(1,4) , rep(1,16)%x%e121.v1)
e121.v2<-c(0,1,2,3)
e121.A2<-rbind(e121.v2%x%rep(1,16) , rep(1,4)%x%e121.v2%x%rep(1,4) , rep(1,16)%x%e121.v2)
e121.X<-rbind(c(1,0,0),c(1,1,0),c(1,0,1))

```

```

e121.propwls<-funlinWLS(model="lin",obj=e121.catdata,A1=e121.A1,X=e121.X)
waldTest(e121.propwls,rbind(c(0,1,0),c(0,0,1)))
waldTest(e121.propwls,c(0,1,-1))
e121.scorwls<-funlinWLS(model="lin",obj=e121.catdata,A1=e121.A2,X=e121.X)
waldTest(e121.scorwls,rbind(c(0,1,0),c(0,0,1)))
waldTest(e121.scorwls,c(0,1,-1))
e121.X2<-rbind(c(1,0),c(1,1),c(1,1))
e121.propwls2<-funlinWLS(model="lin",obj=e121.catdata,A1=e121.A1,X=e121.X2)
e121.scorwls2<-funlinWLS(model="lin",obj=e121.catdata,A1=e121.A2,X=e121.X2)
round(e121.propwls2$beta[1],2)
round(e121.propwls2$beta[1]+c(-1,1)*qnorm(0.975)*sqrt(e121.propwls2$Vbeta[1,1]),2)
round(-e121.propwls2$beta[2],2)
round(-e121.propwls2$beta[2]+c(-1,1)*qnorm(0.975)*sqrt(e121.propwls2$Vbeta[2,2]),2)
round(e121.scorwls2$beta[1],2)
round(e121.scorwls2$beta[1]+c(-1,1)*qnorm(0.975)*sqrt(e121.scorwls2$Vbeta[1,1]),2)
round(-e121.scorwls2$beta[2],2)
round(-e121.scorwls2$beta[2]+c(-1,1)*qnorm(0.975)*sqrt(e121.scorwls2$Vbeta[2,2]),2)
e121.v3<-rbind(c(1,0,0,0),c(0,1,1,1),c(1,1,0,0),c(0,0,1,1),c(1,1,1,0),c(0,0,0,1))
e121.A31<-rbind(e121.v3%x%t(rep(1,16)) , t(rep(1,4))%x%e121.v3%x%t(rep(1,4)) ,
  t(rep(1,16))%x%e121.v3)
e121.A32<-diag(9)%x%t(c(1,-1))
e121.X3<-cbind(rep(1,3)%x%diag(3),cbind(c(0,1,0),c(0,0,1))%x%rep(1,3))
e121.chprwls<-funlinWLS(model=c("lin","log","lin"),obj=e121.catdata,
  A1=e121.A31,A2=e121.A32,X=e121.X3)
round(e121.chprwls$beta,2)
round(exp(e121.chprwls$beta),2)
round(exp(e121.chprwls$beta-qnorm(0.975)*sqrt(diag(e121.chprwls$Vbeta))),2)
round(exp(e121.chprwls$beta+qnorm(0.975)*sqrt(diag(e121.chprwls$Vbeta))),2)
waldTest(e121.chprwls,c(0,0,0,1,-1))
e121.X4<-cbind(e121.X3[,1:3],e121.X3[,4]+e121.X3[,5])
e121.chprwls2<-funlinWLS(model=c("lin","log","lin"),obj=e121.catdata,
  A1=e121.A31,A2=e121.A32,X=e121.X4)
round(e121.chprwls2$beta[4],2)
round(exp(e121.chprwls2$beta[4]),2)
round(exp(e121.chprwls2$beta[4]+c(-1,1)*qnorm(0.975)*sqrt(e121.chprwls2$Vbeta[4,4])),2)
table(e121.raw[,2:1])
table(e121.raw[,c(3,1)])
table(e121.raw[,3:2])
e121.A41<-rbind(diag(16)%x%t(rep(1,4)),diag(4)%x%t(rep(1,4))%x%diag(4),t(rep(1,4))%x%diag(16))
e121.A42<-rbind(c(1,0,0),c(0,0,1))%x%rbind(c(0,0,0,0,1,0,0,0,1,1,0,0,1,1,0))
e121.probsbiv<-funlinWLS(model=c("lin","lin"),obj=e121.catdata,A1=e121.A41,A2=e121.A42,
  X=diag(2))
waldTest(e121.probsbiv,c(1,-1))
round(e121.probsbiv$beta,2)
round(e121.probsbiv$beta-qnorm(0.975)*sqrt(diag(e121.probsbiv$Vbeta)),2)
round(e121.probsbiv$beta+qnorm(0.975)*sqrt(diag(e121.probsbiv$Vbeta)),2)
#pp.442-444
e121.raw2<-data.frame(paciente=rep(1:50,3),corrimento=with(e121.raw,c(inicial,dias14,dias21)),
  avaliacao=c(rep("inicial",50),rep("dias14",50),rep("dias21",50)))
e121.raw2<-e121.raw2[order(e121.raw2$paciente),]
require(gee)
e121.gee<-with(e121.raw2,gee(I(ifelse(corrimento>=2,1,0))~C(avaliacao,base=3),id=paciente,
  family=quasi(link="identity",variance="mu(1-mu)"),corstr="exchangeable",scale.value=1,
  scale.fix=TRUE))
e121.gee2<-with(e121.raw2,gee(I(ifelse(corrimento>=2,1,0))~C(avaliacao,base=3),id=paciente,
  family=quasi(link="identity",variance="mu(1-mu)"),corstr="unstructured",scale.value=1,

```

```

scale.fix=TRUE))
round(matrix(c(e121.propwls$beta,sqrt(diag(e121.propwls$Vbeta)),
  e121.gee$coef,sqrt(diag(e121.gee$robust)),sqrt(diag(e121.gee$naive)),
  e121.gee2$coef,sqrt(diag(e121.gee2$robust)),sqrt(diag(e121.gee2$naive))),3),3) #Tabela 12.10
cov2cor(e121.propwls$VFH) #compare a estimativa da matriz de correlações da análise por MQG...
e121.gee2$work #...com a estimativa da matriz de de correlação de trabalho não estruturada
e121.gee$work

```

Exemplo 12.2 (pp.421, 431, 444): *Problema da sensibilidade dentinária*

```

e122.TF<-rbind(c(22,1,3,6),c(12,10,7,4),c(10,6,12,3),c(5,13,11,3))
e122.catdata<-readCatdata(TF=e122.TF)
e122.A1<-diag(4)%x%(1-rbind(diag(2)%x%t(c(1,1)),t(c(1,1))%x%diag(2)))
e122.A2<-diag(4)%x%t(c(-1,1,1,-1))
e122.X<-cbind(rep(1,4),c(1,1,-1,-1),rep(c(1,-1),2),c(1,-1,-1,1))
e122.wls<-funlinWLS(model=c("lin","log","lin"),obj=e122.catdata,A1=e122.A1,A2=e122.A2,
  X=e122.X)
e122.wls2<-funlinWLS(model=c("lin","log","lin"),obj=e122.catdata,A1=e122.A1,A2=e122.A2,
  X=rep(1,4))
round(exp(e122.wls2$beta),2)
round(exp(e122.wls2$beta+c(-1,1)*qnorm(0.975)*sqrt(e122.wls2$Vbeta)),2)
#pp.444-445
e122.A3<-diag(4)%x%diag(2)%x%t(c(-1,1))
e122.X2<-e122.X %x% diag(2)
e122.wls3<-funlinWLS(model=c("lin","log","lin"),obj=e122.catdata,A1=e122.A1,A2=e122.A3,
  X=e122.X2)
waldTest(e122.wls3,c(0,0,1,1,0,0,0,0))
waldTest(e122.wls3,c(0,0,0,0,1,1,0,0))
waldTest(e122.wls3,c(0,0,0,0,0,0,1,1))
waldTest(e122.wls3,rep(1,4)%x%c(1,-1))
waldTest(e122.wls3,c(0,0,1,-1,0,0,0,0))
waldTest(e122.wls3,c(0,0,0,0,1,-1,0,0))
waldTest(e122.wls3,c(0,0,0,0,0,0,1,-1))
cov2cor(e122.wls3$VFu)

```

Exemplo 12.3 (pp.422, 432): *Problema da maturação do colo do útero*

```

e123.raw<-data.frame(
  paridade=c(rep("N",46),rep("M",37)),
  cons00=c(rep(2,28),1,1,2,2,1,2,2,2,2,1,1,1,1,1,2,1,1,1,
    2,2,2,1,2,2,2,rep(1,11),2,2,1,2,1,1,1,2,2,1,1,2,1,2,2,2,2,2),
  cons24=c(2,0,2,0,1,0,0,0,2,0,0,0,0,2,2,2,0,2,2,1,1,1,0,2,1,2,0,2,1,1,2,0,0,1,0,0,0,0,0,1,
    0,1,1,1,1,0, 2,2,2,1,1,2,2,2,1,1,1,0,1,0,1,1,0,0,0,1,0,0,2,1,1,0,0,2,0,1,1,1,0,2,0,2,0),
  cons48=c(1,0,2,0,1,rep(0,8),1,2,1,0,0,0,1,1,1,0,1,0,2,0,0,1,1,2,0,0,1,rep(0,9),1,1,0,
    2,2,2,1,1,2,1,1,1,0,1,0,1,0,1,1,rep(0,7),1,1,0,0,2,0,1,1,0,0,2,0,2,0),
  cons72=c(rep(0,14),1,rep(0,10),2,0,0,0,1,2,rep(0,12),1,1,0,
    2,1,2,0,0,0,1,0,0,0,1,0,1,0,0,1,rep(0,7),1,rep(0,9),2,0,2,0),
  cons96=c(rep(0,14),1,rep(0,10),2,0,0,0,1,2,rep(0,12),1,1,0,
    2,0,2,rep(0,7),1,0,1,0,0,1,rep(0,7),1,rep(0,9),2,0,2,0))
e123.agr1<-with(e123.raw,aggregate(rep(1,nrow(e123.raw)),by=list(
  cons00=cons00,cons24=cons24,cons48=cons48,cons72=cons72,cons96=cons96,paridade=paridade),sum))
e123.agr2<-with(e123.raw,aggregate(rep(1,nrow(e123.raw)),by=list(
  cons00=cons00,cons24=cons24,cons48=cons48,cons72=cons72,paridade=paridade),sum))
i1<-0;e123.perfil1<-numeric(20);e123.TF1<-matrix(0,2,20)
i2<-0;e123.perfil2<-numeric(14);e123.TF2<-matrix(0,2,14)
for(i00 in 2:1) {

```



```

for(i24 in i00:0) {
  for(i48 in i24:0) {
    for(i72 in i48:0) {
      for(i96 in i72:0) {
        i1<-i1+1
        e123.perfil1[i1]<-10000*i00+1000*i24+100*i48+10*i72+i96
        acess<-with(e123.agr1,x[cons00==i00 & cons24==i24 & cons48==i48 & cons72==i72
          & cons96==i96 & paridade=="M"])
        e123.TF1[1,i1]<-ifelse(length(acess)==0,0,acess)
        acess<-with(e123.agr1,x[cons00==i00 & cons24==i24 & cons48==i48 & cons72==i72
          & cons96==i96 & paridade=="N"])
        e123.TF1[2,i1]<-ifelse(length(acess)==0,0,acess)
      }
      i2<-i2+1
      e123.perfil2[i2]<-1000*i00+100*i24+10*i48+i72
      acess<-with(e123.agr1,x[cons00==i00 & cons24==i24 & cons48==i48 & cons72==i72
        & paridade=="M"])
      e123.TF2[1,i2]<-ifelse(length(acess)==0,0,acess)
      acess<-with(e123.agr1,x[cons00==i00 & cons24==i24 & cons48==i48 & cons72==i72
        & paridade=="N"])
      e123.TF2[2,i2]<-ifelse(length(acess)==0,0,acess)
    }
  }
}
}
}
e123.catdata<-readCatdata(TF=e123.TF2)
e123.perfil2
# 2222 2221 2220 2211 2210 2200 2111 2110 2100 2000 1111 1110 1100 1000
e123.A0<-diag(2)%x%rbind(
#psi_{i12}
  c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
  c(0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0),
  c(0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0),
  c(0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0),
  c(1,1,1,1,1,1,0,0,0,0,0,0,0,0,0),
#psi_{i23}
  c(0,0,0,0,0,0,0,0,0,0,1,0,0,0,1),
  c(0,0,0,0,0,0,0,0,0,1,0,0,0,1,0),
  c(0,0,0,0,0,0,0,1,1,0,0,1,1,0,0),
  c(0,0,0,0,0,1,0,0,0,0,0,0,0,0,0),
  c(0,0,0,1,1,0,0,0,0,0,0,0,0,0,0),
  c(1,1,1,0,0,0,0,0,0,0,0,0,0,0,0),
#psi_{i34}
  c(0,0,0,0,0,1,0,0,1,1,0,0,1,1),
  c(0,0,0,0,1,0,0,1,0,0,0,1,0,0),
  c(0,0,0,1,0,0,1,0,0,0,1,0,0,0),
  c(0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),
  c(0,1,0,0,0,0,0,0,0,0,0,0,0,0,0),
  c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0))
e123.A1<-diag(2)%x%(diag(3)%x%rbind(c(0,1,0,1,0,0),c(0,1,1,1,1,1)))[-1]
e123.A2<-diag(6)%x%t(c(1,-1))
e123.probtrans<-funlinWLS(model=c("exp","lin","log","lin","lin"),obj=e123.catdata,A1=e123.A0,
  A2=e123.A1,A3=e123.A2,X=diag(6))
waldTest(e123.probtrans,cbind(diag(3),-diag(3)))
e123.probtrans2<-funlinWLS(model=c("exp","lin","log","lin","lin"),obj=e123.catdata,A1=e123.A0,
  A2=e123.A1,A3=e123.A2,X=rep(1,2)%x%diag(3))

```

Exemplo 13.2 (p.466)

```
e132.TF<-c(7,11,2,3,9,5,0,10,4, 8,7,3,0, 0,7,14,7)
e132.Zp<-cbind(rbind( cbind(kronecker(rep(1,2),diag(3)),rep(0,6)),
                    cbind(matrix(0,3,3),rep(1,3)) ),
              rbind( cbind(rep(1,3),matrix(0,3,3)),
                    cbind(rep(0,6),kronecker(rep(1,2),diag(3))) ) )
e132.Rp<-c(4,4)
e132.catdata<-readCatdata(TF=e132.TF,Zp=e132.Zp,Rp=e132.Rp) #p_{ij} e \hat{\sigma}(p_{ij})
e132.satmarm1<-satMarML(e132.catdata) #\hat{\theta}_{ij}, \hat{\sigma}(...), Q_V e Q_P(M_2|M1)
e132.satmarm1$alphast #Tabela 13.3 - EMV das prob.condicionais de omissão
e132.TF2<-c(7,11,2,3,9,5,1e-5,10,4, 8,7,3,0, 0,7,14,7) #subst.zero por valor peq.
e132.catdata2<-readCatdata(TF=e132.TF2,Zp=e132.Zp,Rp=e132.Rp)
e132.satmarm12<-satMarML(e132.catdata2)
e132.U<-rbind(c(0, 1,1,-1,0,0,-1, 0),
              c(0,-1,0, 1,0,1, 0,-1) )
e132.linml<-linML(e132.satmarm12,U=e132.U) #\hat{\theta}_{ij}(H), \hat{\sigma}, Q_V(H|M1)
e132.linwls<-funlinWLS(model="lin",obj=e132.satmarm12,U=e132.U) #abordagem híbrida
e132.kA1<-rbind(
  c(rep(c(1,0,0,0),2),1),
  rep(1,9),
  kronecker(diag(3),t(rep(1,3))),
  kronecker(t(rep(1,3)),diag(3)) )
e132.kA2<-rbind(
  cbind(diag(2),matrix(0,2,6)),
  cbind(matrix(0,3,2),kronecker(t(rep(1,2)),diag(3))) )
e132.kA3<-cbind( c(1,0),c(1,1),-c(2,1))%*%t(rep(1,3)) )
e132.kA4<-t(c(1,-1))
e132.kappa<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
  obj=e132.satmarm1,A1=e132.kA1,A2=e132.kA2,A3=e132.kA3,A4=e132.kA4,PI1=-1,X=1)
#Estes resultados não estão no livro, mas ilustram o cálculo do kappa ponderado
W1<-c(1,0.75,0,0.75,1,0.75,0,0.75,1) #pesos quadráticos Fleiss e Cohen (1973)
W2<-c(1,0.5,0,0.5,1,0.5,0,0.5,1) #pesos absolutos Agresti (2002)
e132.kw1A1<-rbind(
  t(W1),
  rep(1,9),
  kronecker(diag(3),t(rep(1,3))),
  kronecker(t(rep(1,3)),diag(3)) )
e132.kw2A1<-rbind(
  t(W2),
  rep(1,9),
  kronecker(diag(3),t(rep(1,3))),
  kronecker(t(rep(1,3)),diag(3)) )
e132.kwA2<-rbind(
  cbind(diag(2),matrix(0,2,6)),
  cbind(matrix(0,9,2),cbind(kronecker(diag(3),rep(1,3)),kronecker(rep(1,3),diag(3)))) )
e132.kw1A3<-cbind( c(1,0),c(1,1),kronecker(-c(2,1),t(W1)) )
e132.kw2A3<-cbind( c(1,0),c(1,1),kronecker(-c(2,1),t(W2)) )
e132.kappaw1<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
  obj=e132.satmarm1,A1=e132.kw1A1,A2=e132.kwA2,A3=e132.kw1A3,A4=e132.kA4,PI1=-1,X=1)
e132.kappaw2<-funlinWLS(model=c("add","exp","lin","log","lin","exp","lin","log","lin"),
  obj=e132.satmarm1,A1=e132.kw2A1,A2=e132.kwA2,A3=e132.kw2A3,A4=e132.kA4,PI1=-1,X=1)
```

Exemplo 13.3 (p.469)

```
e133.TF<-c(77,87,94,70,67,36,143,78, 14,8,3,9, 25,18,43,16, 14,12)
e133.Zp<-cbind(kronecker(diag(4),rep(1,2)),
```

```

      kronecker(diag(2),kronecker(rep(1,2),diag(2))),
      kronecker(diag(2),rep(1,4)) )
e133.Rp<-c(4,4,2)
e133.catdata<-readCatdata(TF=e133.TF,Zp=e133.Zp,Rp=e133.Rp) #p_{ij}, \hat{\sigma}(p_{ij})
e133.satmcarml<-satMarML(e133.catdata,missing="MCAR")#\hat{\theta}_{ij}, Q_V e Q_P(M_2|M1)
e133.satmcarwls<-satMcarWLS(e133.catdata) #\tilde{\theta}_{ij},\hat{\sigma}(...), QN_(H_0)
e133.X<-rbind(c( 1, 1, 1, 1, 1, 1),
             c( 1, 1,-1, 1,-1,-1),
             c( 1,-1, 1, -1, 1,-1),
             c( 1,-1,-1, -1,-1, 1),
             c(-1, 1, 1, -1,-1, 1),
             c(-1, 1,-1, -1, 1,-1),
             c(-1,-1, 1, 1,-1,-1),
             c(-1,-1,-1, 1, 1, 1))
e133.loglinml<-loglinML(obj=e133.satmcarml,X=e133.X)
e133.loglinwls<-funlinWLS(model=c("lin","log"),obj=e133.satmcarwls,X=e133.X)
e133.loglinhib<-funlinWLS(model=c("lin","log"),obj=e133.satmcarml,X=e133.X)
e133.loglinml2<-loglinML(obj=e133.satmcarml,X=e133.X[,-4])
e133.loglinml3<-loglinML(obj=e133.satmcarml,X=e133.X[,-5])
e133.loglinml4<-loglinML(obj=e133.satmcarml,X=e133.X[,-6])
e133.loglinml2$QvH-e133.loglinml$QvH
e133.loglinml3$QvH-e133.loglinml$QvH
e133.loglinml4$QvH-e133.loglinml$QvH
1-pchisq(e133.loglinml2$QvH-e133.loglinml$QvH,1)
1-pchisq(e133.loglinml3$QvH-e133.loglinml$QvH,1)
1-pchisq(e133.loglinml4$QvH-e133.loglinml$QvH,1)
#note delta^{A(1)}=delta^{A(2)} sob ausência de interação de 2a.ordem
c(1,-1,-1,1,0,0,0,0)%%log(e133.loglinml$thetaH)
c(0,0,0,0,1,-1,-1,1)%%log(e133.loglinml$thetaH)
c(1,-1,-1,1,0,0,0,0)%%e133.X
c(0,0,0,0,1,-1,-1,1)%%e133.X
round(4*e133.loglinml$beta[6:4],3)
round(4*sqrt(diag(e133.loglinml$Vbeta))[6:4],3)
round(4*e133.loglinwls$beta[6:4],3)
round(4*sqrt(diag(e133.loglinwls$Vbeta))[6:4],3)
round(4*e133.loglinhib$beta[6:4],3)
round(4*sqrt(diag(e133.loglinhib$Vbeta))[6:4],3)

```

Exemplo 13.4 (p.472) / 13.1 (p.454)

```

e134.TF<-c(12,4,5,2, 50,31, 27,12)
e134.Zp<-cbind(kronecker(diag(2),rep(1,2)),kronecker(rep(1,2),diag(2)))
e134.Rp<-c(2,2)
e134.catdata<-readCatdata(TF=e134.TF,Zp=e134.Zp,Rp=e134.Rp)
e134.satmcarml<-satMarML(e134.catdata,miss="MCAR")#MV MCAR Tabs.13.10 (Inf.Fisher),13.11,13.12
e134.satmarmml<-satMarML(e134.catdata)#Results.MV MAR_{sat} Tabs.13.11-13.12
e134.satmcarwls<-satMcarWLS(e134.catdata)
e134.A<-rbind(c(1,1,0,0),c(1,0,1,0))
e134.hmmcarml<-linML(e134.satmcarml,A=e134.A,X=rep(1,2)) #Tab.13.12/13.13
e134.hmmcarwls<-funlinWLS(model="lin",obj=e134.satmcarwls,A1=e134.A,X=rep(1,2)) #Tab.13.12
e134.hmmcarhib<-funlinWLS(model="lin",obj=e134.satmcarml,A1=e134.A,X=rep(1,2))
e134.TF2<-c(e134.TF,24) #para mecanismos MNAR, cenários de omissão total trazem inf.na estim.
mnarsat.mlv<-function(p,n11=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1,\alpha_2
t11<-p[1];t12<-p[2];t21<-p[3]

```

```

a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7];a2<-p[8]
value<- -(
  n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
  n112*log( t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))) )+
  n121*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
  n122*log( (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
    (exp(a20+a1+a2)/(1+exp(a20+a1+a2))) )+
  n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
    t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))) )+
  n22*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
    (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
    (1/(1+exp(a20+a1+a2))) )+
  n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
    t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
  n32*log( t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))) +
    (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*
    (exp(a30+a1+a2)/(1+exp(a30+a1+a2))) )+
  N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
    t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))) +
    t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
    (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2))) )
)
value
}
mnarsat.der<-deriv3(~-(
  n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
  n112*log( t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))) )+
  n121*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
  n122*log( (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
    (exp(a20+a1+a2)/(1+exp(a20+a1+a2))) )+
  n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
    t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))) )+
  n22*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
    (1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
    (1/(1+exp(a20+a1+a2))) )+
  n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
    t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
  n32*log( t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))) +
    (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*
    (exp(a30+a1+a2)/(1+exp(a30+a1+a2))) )+
  N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
    t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))) +
    t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
    (1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2))) )
),c("t11","t12","t21","a10","a20","a30","a1","a2"),
c("t11","t12","t21","a10","a20","a30","a1","a2",
  "n111","n112","n121","n122","n21","n22","n31","n32","N4")
) #obtem o gradiente e a hessiana analiticamente
mnarsat.esp<-function(p,N){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1,\alpha_2
t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7];a2<-p[8]
value<-N*c(
  t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))),
  t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),

```

```

(1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*(exp(a20+a1+a2)/(1+exp(a20+a1+a2))),
t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))),
t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
(1-t11-t12-t21)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*(1/(1+exp(a20+a1+a2))),
t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))),
t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
(1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(exp(a30+a1+a2)/(1+exp(a30+a1+a2))),
t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))),
t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))),
(1-t11-t12-t21)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2)))
)
value
}
mnarred.mlv<-function(p,n111=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7]
value<- -(
n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
n122*log( (1-t11-t12-t21)*(exp(a10+a1)/(1+exp(a10+a1)))*
(exp(a20+a1)/(1+exp(a20+a1))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
(1-t11-t12-t21)*(exp(a10+a1)/(1+exp(a10+a1)))*
(1/(1+exp(a20+a1))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10+a1)))*
(exp(a30+a1)/(1+exp(a30+a1))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
(1-t11-t12-t21)*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) )
)
value
}
mnarred.der<-deriv3(~-(
n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
n122*log( (1-t11-t12-t21)*(exp(a10+a1)/(1+exp(a10+a1)))*
(exp(a20+a1)/(1+exp(a20+a1))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
(1-t11-t12-t21)*(exp(a10+a1)/(1+exp(a10+a1)))*
(1/(1+exp(a20+a1))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +

```

```

t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10+a1)))*
(exp(a30+a1)/(1+exp(a30+a1))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
(1-t11-t12-t21)*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) )
),c("t11","t12","t21","a10","a20","a30","a1"),
c("t11","t12","t21","a10","a20","a30","a1",
"n111","n112","n121","n122","n21","n22","n31","n32","N4")
)
mnrred.esp<-function(p,N){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7]
value<-N*c(
t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t21*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),
(1-t11-t12-t21)*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),
t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t21*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
(1-t11-t12-t21)*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t21*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
(1-t11-t12-t21)*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t21*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))),
(1-t11-t12-t21)*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1)))
)
value
}
mnrred.mlv<-function(p,n111=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7]
value<- -(
n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
n122*log( (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))) +
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
t21*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +

```

```

      t21*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
      (1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a30))) )
)
value
}
marred.der<-deriv3(~(
  n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
  n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
  n121*log( t21*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
  n122*log( (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
  n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
    t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
  n22*log( t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))) +
    (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))) )+
  n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
    t21*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
  n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
    (1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
  N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
    t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
    t21*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
    (1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a30))) )
  ),c("t11","t12","t21","a10","a20","a30","a1"),
  c("t11","t12","t21","a10","a20","a30","a1",
    "n111","n112","n121","n122","n21","n22","n31","n32","N4")
)
marred.esp<-function(p,N){
  #p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
  t11<-p[1];t12<-p[2];t21<-p[3]
  a10<-p[4];a20<-p[5];a30<-p[6];a1<-p[7]
  value<-N*c(
    t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
    t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
    t21*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))),
    (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))),
    t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
    t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
    t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))),
    (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))),
    t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
    t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
    t21*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
    (1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
    t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
    t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
    t21*(1/(1+exp(a10)))*(1/(1+exp(a30))),
    (1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a30)))
  )
  value
}
mcar.mlv<-function(p,n111=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
  n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
  #p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30}
  t11<-p[1];t12<-p[2];t21<-p[3]
  a10<-p[4];a20<-p[5];a30<-p[6]
  value<- -(

```

```

n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n122*log( (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
t21*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t21*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a30))) )
)
value
}
mcar.der<-deriv3(~-(
n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n122*log( (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
t21*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t21*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a30))) )
),c("t11","t12","t21","a10","a20","a30"),
c("t11","t12","t21","a10","a20","a30",
"n111","n112","n121","n122","n21","n22","n31","n32","N4")
)
mcar.esp<-function(p,N){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30}
t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5];a30<-p[6]
value<-N*c(
t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t21*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t21*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),

```



```

(1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t21*(1/(1+exp(a10)))*(1/(1+exp(a30))),
(1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a30)))
)
value
}
mcarred.mlv<-function(p,n111=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10}=\alpha_{30},\alpha_{20}
t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5]
value<- -(
n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n122*log( (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) +
t21*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
t21*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a10))) )
)
value
}
mcarred.der<-deriv3(~-(
n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t21*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n122*log( (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
(1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) +
t21*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
t21*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
(1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a10))) )
),c("t11","t12","t21","a10","a20"),
c("t11","t12","t21","a10","a20",
"n111","n112","n121","n122","n21","n22","n31","n32","N4")
)
mcarred.esp<-function(p,N){
#p=\theta_{11},\theta_{12},\theta_{21},\alpha_{10}=\alpha_{30},\alpha_{20}

```

```

t11<-p[1];t12<-p[2];t21<-p[3]
a10<-p[4];a20<-p[5]
value<-N*c(
  t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t21*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
  t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
  t21*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
  (1-t11-t12-t21)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
  t11*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))), #acertar a ordem dos thetas
  t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))),
  t21*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))),
  (1-t11-t12-t21)*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))),
  t11*(1/(1+exp(a10)))*(1/(1+exp(a10))),
  t12*(1/(1+exp(a10)))*(1/(1+exp(a10))),
  t21*(1/(1+exp(a10)))*(1/(1+exp(a10))),
  (1-t11-t12-t21)*(1/(1+exp(a10)))*(1/(1+exp(a10)))
)
value
}
require(geoR) #.nlmP adapta nlm p/restringir o espaço paramétrico. Isso é importante,
inipars<-c(0.25,0.25,0.25,0,0,0,0,0) #pois mecanismos MNAR resultam facilmente em estims.
minpars<-c(0,0,0,-Inf,-Inf,-Inf,-Inf,-Inf) #p/probs. >1 ou <0 quando não se usa o EM ou
maxpars<-c(1,1,1,Inf,Inf,Inf,Inf,Inf) #funções ligações próprias para probs.(e.g.,logito)
mnarsat<- .nlmP(objfunc=mnarsat.mlv,params=inipars
  ,
  lower=minpars
  ,upper=maxpars
  ,hessian=T)
mnrred<- .nlmP(objfunc=mnrred.mlv,params=inipars[-8]
  ,
  lower=minpars[-8]
  ,upper=maxpars[-8]
  ,hessian=T)
mrred <- .nlmP(objfunc=mrred.mlv ,params=inipars[-8]
  ,
  lower=minpars[-8]
  ,upper=maxpars[-8]
  ,hessian=T)
mcar <- .nlmP(objfunc=mcar.mlv ,params=inipars[-(7:8)]
  ,
  lower=minpars[-(7:8)]
  ,upper=maxpars[-(7:8)]
  ,hessian=T)
mcrred<- .nlmP(objfunc=mcrred.mlv,params=inipars[-(6:8)]
  ,
  lower=minpars[-(6:8)]
  ,upper=maxpars[-(6:8)]
  ,hessian=T)
p<-mnarsat$est
mnarsat.infobs<-attr(mnarsat.der(p[1],p[2],p[3],p[4],p[5],p[6],p[7],p[8],e134.TF2[1],e134.TF2[2],
e134.TF2[3],e134.TF2[4],e134.TF2[5],e134.TF2[6],e134.TF2[7],e134.TF2[8],e134.TF2[9]),
  "hessian")[1,,] #informação observada obtida analiticamente
p<-mnrred$est
mnrred.infobs<-attr(mnrred.der(p[1],p[2],p[3],p[4],p[5],p[6],p[7],e134.TF2[1],e134.TF2[2],
e134.TF2[3],e134.TF2[4],e134.TF2[5],e134.TF2[6],e134.TF2[7],e134.TF2[8],e134.TF2[9]),
  "hessian")[1,,]
p<-mrred$est
mrred.infobs<-attr(mrred.der(p[1],p[2],p[3],p[4],p[5],p[6],p[7],e134.TF2[1],e134.TF2[2],
e134.TF2[3],e134.TF2[4],e134.TF2[5],e134.TF2[6],e134.TF2[7],e134.TF2[8],e134.TF2[9]),
  "hessian")[1,,]
p<-mcar$est
mcar.infobs<-attr(mcar.der(p[1],p[2],p[3],p[4],p[5],p[6],e134.TF2[1],e134.TF2[2],
e134.TF2[3],e134.TF2[4],e134.TF2[5],e134.TF2[6],e134.TF2[7],e134.TF2[8],e134.TF2[9]),
  "hessian")[1,,]
p<-mcrred$est
mcrred.infobs<-attr(mcrred.der(p[1],p[2],p[3],p[4],p[5],e134.TF2[1],e134.TF2[2],
e134.TF2[3],e134.TF2[4],e134.TF2[5],e134.TF2[6],e134.TF2[7],e134.TF2[8],e134.TF2[9]),
  "hessian")[1,,]

```

```

mnarsat.infobs2<-mnarsat$hess #informação observada obtida numericamente
mnarred.infobs2<-mnarred$hess
marred.infobs2<-marred$hess
mcar.infobs2<-mcar$hess
mcarred.infobs2<-mcarred$hess
round(mnarsat.infobs-mnarsat.infobs2,3) #note as diferenças
round(mnarred.infobs-mnarred.infobs2,3)
round(marred.infobs-marred.infobs2,3)
round(mcarred.infobs-mcarred.infobs2,3)
round(mcar.infobs-mcar.infobs2,3)
mnarsat.cov<-solve(mnarsat.infobs)
mnarred.cov<-solve(mnarred.infobs)
marred.cov<-solve(marred.infobs)
mcar.cov<-solve(mcar.infobs)
mcarred.cov<-solve(mcarred.infobs)
mnarsat.cov2<-solve(mnarsat.infobs2)
mnarred.cov2<-solve(mnarred.infobs2)
marred.cov2<-solve(marred.infobs2)
mcar.cov2<-solve(mcar.infobs2)
mcarred.cov2<-solve(mcarred.infobs2)
round(mnarsat.cov-mnarsat.cov2,6) #as diferenças são maiores justamente
round(mnarred.cov-mnarred.cov2,6) #para os parâmetros de interesse
round(marred.cov-marred.cov2,6)
round(mcarred.cov-mcarred.cov2,6)
round(mcar.cov-mcar.cov2,6)
b<-c(rep(0,3),1)
B<-rbind(diag(3),rep(-1,3))
mnarsat.estp<-c(b+B%*mnarsat$est[1:3])
mnarred.estp<-c(b+B%*mnarred$est[1:3])
marred.estp<-c(b+B%*marred$est[1:3])
mcar.estp<-c(b+B%*mcar$est[1:3])
mcarred.estp<-c(b+B%*mcarred$est[1:3])
mnarsat.covp<-B%*mnarsat.cov[1:3,1:3]%%t(B)
mnarred.covp<-B%*mnarred.cov[1:3,1:3]%%t(B)
marred.covp<-B%*marred.cov[1:3,1:3]%%t(B)
mcar.covp<-B%*mcar.cov[1:3,1:3]%%t(B)
mcarred.covp<-B%*mcarred.cov[1:3,1:3]%%t(B)
#Recorde, no Exercício~13.5, item (c), que a estimativa da matriz de informação de Fisher
#relativa a \theta sob o mecanismo MAR é igual à estimativa da matriz de inf.observada
#relativa a \theta sob os mecanismos MAR e MCAR
#Como a rotina satMarML utiliza a informação de Fisher, devemos comparar a estimativa
#da matriz de covariâncias obtida sob o mecanismo MAR com as dos mecanismos MAR_red,
#MCAR, MCAR_red baseadas na matriz de informação observada
round(e134.satmarml$Vtheta-marred.covp,6)
round(e134.satmarml$Vtheta-mcar.covp,6)
round(e134.satmarml$Vtheta-mcarred.covp,6)
round(e134.satmarml$Vtheta-B%*marred.cov2[1:3,1:3]%%t(B),6)
round(e134.satmarml$Vtheta-B%*mcar.cov2[1:3,1:3]%%t(B),6)
round(e134.satmarml$Vtheta-B%*mcarred.cov2[1:3,1:3]%%t(B),6)
#Portanto, sugere-se calcular sempre a inf.obs.analítica!
e134.satmarml$theta-marred.estp
e134.satmarml$theta-mcar.estp
e134.satmarml$theta-mcarred.estp
#Recorde tb que as estimativas de MV dos \theta são sempre iguais para todos os
#mecanismo MAR, MCAR e estruturas mais reduzidas destes.
cbind(mnarsat.estp,mnarred.estp,marred.estp,mcar.estp,mcarred.estp) #Tab.13.10

```

```

mnarsat$est[-(1:3)];mnrred$est[-(1:3)] #EMV dos \alphas da Tabela 13.10
marred$est[-(1:3)];mcar$est[-(1:3)];mcarred$est[-(1:3)]
cbind(sqrt(diag(mnarsat.covp)),sqrt(diag(mnrred.covp)),sqrt(diag(marred.covp)),
  sqrt(diag(mcar.covp)),sqrt(diag(mcarred.covp))) #erros padrões dos \thetas Tab.13.10
sqrt(diag(mnarsat.cov)) [4:8];sqrt(diag(mnrred.cov)) [4:7];sqrt(diag(marred.cov)) [4:7]
sqrt(diag(mcar.cov)) [4:6];sqrt(diag(mcarred.cov)) [4:5] #e.p.s dos \alphas Tab.13.10
mnarsat.wls<-funlinWLS(model="lin",theta=mnarsat.estp,Vtheta=mnarsat.covp,A1=t(c(0,1,-1,0)),X=1)
mnrred.wls<-funlinWLS(model="lin",theta=mnrred.estp,Vtheta=mnrred.covp,A1=t(c(0,1,-1,0)),X=1)
marred.wls<-funlinWLS(model="lin",theta=marred.estp,Vtheta=marred.covp,A1=t(c(0,1,-1,0)),X=1)
mcar.wls<-funlinWLS(model="lin",theta=mcar.estp,Vtheta=mcar.covp,A1=t(c(0,1,-1,0)),X=1)
mcarred.wls<-funlinWLS(model="lin",theta=mcarred.estp,Vtheta=mcarred.covp,A1=t(c(0,1,-1,0)),X=1)
rbind(c(mnarsat.wls$beta,mnrred.wls$beta,marred.wls$beta,mcar.wls$beta,mcarred.wls$beta),
  c(mnarsat.wls$Vbeta,mnrred.wls$Vbeta,marred.wls$Vbeta,mcar.wls$Vbeta,mcarred.wls$Vbeta))
-c(mnarsat$min,mnrred$min,marred$min,mcar$min,mcarred$min)#log-veros. Tab.13.10
sat.lv<-sum(e134.TF2*log(e134.TF2/sum(e134.TF2))) #vlr.máx.da log-veros.de um mod.sat.
#Modelos MNAR saturados podem não ter um ajuste perfeito, veja Baker e Laird (1988) ou
#Poletto (2006, pp.21-26). Além disso, modelos MNAR podem ter problemas de
#identificabilidade dos parâmetros, veja Poletto (2006, pp.27-30).
#Poletto (2006, pp.31-40) realiza um estudo de simulação para avaliar estas 2 patologias.
-2*(-c(mnarsat$min,mnrred$min,marred$min,mcar$min,mcarred$min)-sat.lv)
1-pchisq(2*(c(mnrred$min,marred$min,mcar$min,mcarred$min)+sat.lv),c(1,1,2,3)) #Valor-P
c(mnarsat$code,mnrred$code,marred$code,mcar$code,mcarred$code)
c(mnarsat$it,mnrred$it,marred$it,mcar$it,mcarred$it)
#Tabela 13.11
matrix(mnarsat.esp(p=mnarsat$est,sum(e134.TF2)) [rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2)
matrix(mnrred.esp(p=mnrred$est,sum(e134.TF2)) [rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2)
matrix(c(e134.satmarm1$yst$st1.1,e134.satmarm1$yst$st1.2,e134.satmarm1$yst$st1.3,
  24*e134.satmarm1$theta) [rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2) #MAR saturado
matrix(marred.esp(p=marred$est,sum(e134.TF2)) [rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2)
matrix(mcar.esp(p=mcar$est,sum(e134.TF2)) [rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2)
matrix(mcarred.esp(p=mcarred$est,sum(e134.TF2)) [rep(c(1,3,2,4),4)+rep(seq(0,12,4),rep(4,4))],2)
#Em tabelas 2x2, homogeneidade marginal <=> simetria
mnarsatHM.mlv<-function(p,n11=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
  n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
  #p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_{1},\alpha_{2}
  t11<-p[1];t12<-p[2]
  a10<-p[3];a20<-p[4];a30<-p[5];a1<-p[6];a2<-p[7]
  value<- -(
    n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
    n112*log( t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))) )+
    n121*log( t12*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
    n122*log( (1-t11-t12-t12)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
      (exp(a20+a1+a2)/(1+exp(a20+a1+a2))) )+
    n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
      t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))) )+
    n22*log( t12*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
      (1-t11-t12-t12)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*
      (1/(1+exp(a20+a1+a2))) )+
    n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
      t12*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
    n32*log( t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))) +
      (1-t11-t12-t12)*(1/(1+exp(a10+a1+a2)))*
      (exp(a30+a1+a2)/(1+exp(a30+a1+a2))) )+
    N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
      t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))) +
      t12*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +

```

```

      (1-t11-t12-t12)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2))) )
    )
  value
}
mnarsatHM.esp<-function(p,N){
  #p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1,\alpha_2
  t11<-p[1];t12<-p[2]
  a10<-p[3];a20<-p[4];a30<-p[5];a1<-p[6];a2<-p[7]
  value<-N*c(
    t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
    t12*(exp(a10+a2)/(1+exp(a10+a2)))*(exp(a20+a2)/(1+exp(a20+a2))),
    t12*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),
    (1-t11-t12-t12)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*(exp(a20+a1+a2)/(1+exp(a20+a1+a2))),
    t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
    t12*(exp(a10+a2)/(1+exp(a10+a2)))*(1/(1+exp(a20+a2))),
    t12*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
    (1-t11-t12-t12)*(exp(a10+a1+a2)/(1+exp(a10+a1+a2)))*(1/(1+exp(a20+a1+a2))),
    t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
    t12*(1/(1+exp(a10+a2)))*(exp(a30+a2)/(1+exp(a30+a2))),
    t12*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
    (1-t11-t12-t12)*(1/(1+exp(a10+a1+a2)))*(exp(a30+a1+a2)/(1+exp(a30+a1+a2))),
    t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
    t12*(1/(1+exp(a10+a2)))*(1/(1+exp(a30+a2))),
    t12*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))),
    (1-t11-t12-t12)*(1/(1+exp(a10+a1+a2)))*(1/(1+exp(a30+a1+a2)))
  )
  value
}
mnarredHM.mlv<-function(p,n111=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
  #p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
  t11<-p[1];t12<-p[2]
  a10<-p[3];a20<-p[4];a30<-p[5];a1<-p[6]
  value<- -(
    n111*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
    n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
    n121*log( t12*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
    n122*log( (1-t11-t12-t12)*(exp(a10+a1)/(1+exp(a10+a1)))*
              (exp(a20+a1)/(1+exp(a20+a1))) )+
    n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
              t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
    n22*log( t12*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))) +
              (1-t11-t12-t12)*(exp(a10+a1)/(1+exp(a10+a1)))*
              (1/(1+exp(a20+a1))) )+
    n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
              t12*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))) )+
    n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
              (1-t11-t12-t12)*(1/(1+exp(a10+a1)))*
              (exp(a30+a1)/(1+exp(a30+a1))) )+
    N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
              t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
              t12*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) +
              (1-t11-t12-t12)*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))) )
  )
  value
}

```

```

mnrredHM.esp<-function(p,N){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4];a30<-p[5];a1<-p[6]
value<-N*c(
  t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t12*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),
  (1-t11-t12-t12)*(exp(a10+a1)/(1+exp(a10+a1)))*(exp(a20+a1)/(1+exp(a20+a1))),
  t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
  t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
  t12*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
  (1-t11-t12-t12)*(exp(a10+a1)/(1+exp(a10+a1)))*(1/(1+exp(a20+a1))),
  t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
  t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
  t12*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
  (1-t11-t12-t12)*(1/(1+exp(a10+a1)))*(exp(a30+a1)/(1+exp(a30+a1))),
  t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
  t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
  t12*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1))),
  (1-t11-t12-t12)*(1/(1+exp(a10+a1)))*(1/(1+exp(a30+a1)))
)
value
}
marredHM.mlv<-function(p,n11=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4];a30<-p[5];a1<-p[6]
value<- -(
  n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
  n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
  n121*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
  n122*log( (1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))) )+
  n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
  t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
  n22*log( t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))) +
  (1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))) )+
  n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
  t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
  n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
  (1-t11-t12-t12)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
  N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
  t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
  t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
  (1-t11-t12-t12)*(1/(1+exp(a10)))*(1/(1+exp(a30))) )
)
value
}
marredHM.esp<-function(p,N){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30},\alpha_1
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4];a30<-p[5];a1<-p[6]
value<-N*c(
  t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
  t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),

```

```

t12*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))),
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(exp(a20+a1)/(1+exp(a20+a1))),
t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))),
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20+a1))),
t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas
t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
(1-t11-t12-t12)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
(1-t11-t12-t12)*(1/(1+exp(a10)))*(1/(1+exp(a30)))
)
value
}
mcarHM.mlv<-function(p,n11=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30}
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4];a30<-p[5]
value<- -(
n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n122*log( (1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) +
(1-t11-t12-t12)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))) +
(1-t11-t12-t12)*(1/(1+exp(a10)))*(1/(1+exp(a30))) )
)
value
}
mcarHM.esp<-function(p,N){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10},\alpha_{20},\alpha_{30}
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4];a30<-p[5]
value<-N*c(
t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t11*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))), #acertar a ordem dos thetas

```

```

t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
(1-t11-t12-t12)*(1/(1+exp(a10)))*(exp(a30)/(1+exp(a30))),
t11*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a30))),
(1-t11-t12-t12)*(1/(1+exp(a10)))*(1/(1+exp(a30)))
)
value
}
mcarredHM.mlv<-function(p,n11=e134.TF2[1],n112=e134.TF2[2],n121=e134.TF2[3],n122=e134.TF2[4],
n21=e134.TF2[5],n22=e134.TF2[6],n31=e134.TF2[7],n32=e134.TF2[8],N4=e134.TF2[9]){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10}=\alpha_{30},\alpha_{20}
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4]
value<- -(
n11*log( t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n112*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n121*log( t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n122*log( (1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))) )+
n21*log( t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n22*log( t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) +
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))) )+
n31*log( t11*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) +
t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) )+
n32*log( t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) +
(1-t11-t12-t12)*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))) )+
N4*log( t11*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
t12*(1/(1+exp(a10)))*(1/(1+exp(a10))) +
(1-t11-t12-t12)*(1/(1+exp(a10)))*(1/(1+exp(a10))) )
)
value
}
mcarredHM.esv<-function(p,N){
#p=\theta_{11},\theta_{12}=\theta_{21},\alpha_{10}=\alpha_{30},\alpha_{20}
t11<-p[1];t12<-p[2]
a10<-p[3];a20<-p[4]
value<-N*c(
t11*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(exp(a20)/(1+exp(a20))),
t11*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t12*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
(1-t11-t12-t12)*(exp(a10)/(1+exp(a10)))*(1/(1+exp(a20))),
t11*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))), #acertar a ordem dos thetas
t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))),
t12*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))),
(1-t11-t12-t12)*(1/(1+exp(a10)))*(exp(a10)/(1+exp(a10))),
t11*(1/(1+exp(a10)))*(1/(1+exp(a10))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a10))),
t12*(1/(1+exp(a10)))*(1/(1+exp(a10))),
(1-t11-t12-t12)*(1/(1+exp(a10)))*(1/(1+exp(a10)))
)
}

```



```

)
value
}
mnarsatHM<- .nlmP(objfunc=mnarsatHM.mlv,params=inipars[-3]
, lower=minipars[-3] ,upper=maxpars[-3])
mnarredHM<- .nlmP(objfunc=mnarredHM.mlv,params=inipars[-c(3,8)]
, lower=minipars[-c(3,8)] ,upper=maxpars[-c(3,8)])
marredHM <- .nlmP(objfunc=marredHM.mlv ,params=inipars[-c(3,8)]
, lower=minipars[-c(3,8)] ,upper=maxpars[-c(3,8)])
mcarHM <- .nlmP(objfunc=mcarHM.mlv ,params=inipars[-c(3,7:8)],
, lower=minipars[-c(3,7:8)],upper=maxpars[-c(3,7:8)])
mcarredHM<- .nlmP(objfunc=mcarredHM.mlv,params=inipars[-c(3,6:8)],
, lower=minipars[-c(3,6:8)],upper=maxpars[-c(3,6:8)])
c(mnarsatHM$code,mnarredHM$code,marredHM$code,mcarHM$code,mcarredHM$code)
c(mnarsatHM$it,mnarredHM$it,marredHM$it,mcarHM$it,mcarredHM$it)
-c(mnarsatHM$min,mnarredHM$min,marredHM$min,mcarHM$min,mcarredHM$min)
2*(c(mnarsatHM$min,mnarredHM$min,marredHM$min,mcarHM$min,mcarredHM$min)+sat.lv) #Q_V(M,H)
1-pchisq(2*(c(mnarsatHM$min,mnarredHM$min,marredHM$min,mcarHM$min,
mcarredHM$min)+sat.lv),c(1,2,2,3,4)) #Valor-P
2*(c(mnarsatHM$min,mnarredHM$min,marredHM$min,mcarHM$min,mcarredHM$min)-
c(mnarsat$min,mnarred$min,marred$min,mcar$min,mcarred$min)) #Q_V(H|M)
1-pchisq(2*(c(mnarsatHM$min,mnarredHM$min,marredHM$min,mcarHM$min,mcarredHM$min)-
c(mnarsat$min,mnarred$min,marred$min,mcar$min,mcarred$min)),1) #Valor-P
QP<-function(esp){
esp2<-c(esp[1:4],sum(esp[5:6]),sum(esp[7:8]),sum(esp[c(9,11)]),sum(esp[c(10,12)]),
sum(esp[13:16]))
c(t(e134.TF2-esp2)%*%solve(diag(esp2))%*%(e134.TF2-esp2))
}
QPs<-c(QP(mnarsatHM.esp(p=mnarsatHM$est,sum(e134.TF2))),
QP(mnarredHM.esp(p=mnarredHM$est,sum(e134.TF2))),
QP(marredHM.esp(p=marredHM$est,sum(e134.TF2))),
QP(mcarHM.esp(p=mcarHM$est,sum(e134.TF2))),
QP(mcarredHM.esp(p=mcarredHM$est,sum(e134.TF2))))
rbind(QPs,1-pchisq(QPs,c(1,2,2,3,4))) #Q_P(M,H) / Valor-P

```

Agradecimentos

O autor agradece aos professores Julio da Motta Singer e Carlos Daniel Paulino pela atenciosa e dedicada orientação durante todo o longo período de desenvolvimento do mestrado, escrita da dissertação, implementação das rotinas e redação deste texto, que com certeza elevou excepcionalmente a qualidade de todos os trabalhos, obviamente encadeados. O mestrado e, conseqüentemente, este trabalho receberam apoio financeiro do CNPq, permitindo um maior empenho do autor para a sua realização. O autor é grato aos professores Carlos Daniel Paulino, Giovani Loiola Silva, Julio da Motta Singer, Paulo Soares e Viviana Giampaoli por terem divulgado as rotinas em cursos de análise de dados categorizados que ministraram na graduação e pós-graduação do Instituto de Matemática e Estatística da Universidade de São Paulo e do Instituto Superior Técnico da Universidade Técnica de Lisboa nos anos de 2005, 2006 e 2007, possibilitando que as rotinas fossem mais testadas e, por conseguinte, aprimoradas. Agradece-se aos alunos destes cursos pelas sugestões, críticas e comentários. O autor também é grato ao professor Paulo Soares por sua cautelosa revisão de uma versão anterior deste texto e pelas sugestões de aperfeiçoamento das rotinas. Agradece-se ao Elias Teixeira Krainski, Fernando Henrique Ferraz Pereira da Rosa e aos

professores Francisco José de Azevêdo Cysneiros, Laura A. Thompson e Paulo Justiniano Ribeiro Júnior por gentilmente terem auxiliado o autor com esclarecimentos de dúvidas sobre o R. Quaisquer erros e imprecisões que eventualmente permaneçam neste texto e/ou nas rotinas são de inteira responsabilidade do autor. Agradece-se previamente a todos aqueles que contribuirão para melhorar o resultado deste trabalho expressando suas críticas, comentários e sugestões.

Referências

- CHAMBERS, J.M. (1998). *Programming with data: a guide to the S language*. New York: Springer-Verlag.
- CHAMBERS, J.M. e HASTIE, T.J. (1991). *Statistical models in S*. London: Chapman & Hall / CRC.
- PAULA, G.A. (2000). *Introdução ao S-Plus*. Disponível em <http://www.ime.usp.br/~giapaula/apostila.html> (05/03/2007).
- PAULINO, C.D. e SINGER, J.M. (2006). *Análise de dados categorizados*. São Paulo: Edgard Blücher.
- POLETO, F.Z. (2006). *Análise de dados categorizados com omissão*. Dissertação de mestrado. IME-USP. Disponível em <http://www.poleto.com/missing.html> (05/03/2007)
- POLETO, F.Z., SINGER, J.M. e PAULINO, C.D. (2007a). *A product-multinomial framework for categorical data analysis with missing responses*. Relatório técnico RT-MAE-2007-07. IME-USP. Disponível em <http://www.poleto.com/missing.html> (05/03/2007).
- POLETO, F.Z., SINGER, J.M. e PAULINO, C.D. (2007b). *Analyzing categorical data with complete or missing responses using the Catdata package*. Vinheta para o R. Disponível em <http://www.poleto.com/missing.html> (05/03/2007).
- R DEVELOPMENT CORE TEAM (2006). *R: a language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.
- RIBEIRO JR., P.J. (sem data). *Tutorial de introdução ao R*. Disponível em <http://www.est.ufpr.br/Rtutorial> (05/03/2007).
- THOMPSON, L.A. (2007). *S-Plus (and R) manual to accompany Agresti's "Categorical Data Analysis" (2002) 2nd edition*. Disponível em <https://home.comcast.net/~lthompson221/Splusdiscrete2.pdf> (05/03/2007).
- VENABLES, W.N. e RIPLEY, B.D. (2000). *S programming*. New York: Springer-Verlag.
- VENABLES, W.N. e RIPLEY, B.D. (2002). *Modern applied statistics with S-Plus*. 4^a ed. New York: Springer-Verlag.