**Universidade Federal da Bahia**
**Instituto de Matemática**
**Departamento de Estatística**

# ANÁLISE DE CLASSES LATENTES:

## Um Tutorial usando Software Estatístico

**Leila Denise Alves Ferreira Amorim**[1]
**Marcus Elias Silva Freire**[2]
**Pétala Gardênia da S. E. Tuy**[2]
**Juan Martin Leroux**[2]
**Nila Mara Smith Galvão Bahamonde**[3]

[1] Departamento de Estatística, Universidade Federal da Bahia, Salvador Bahia.
[2] Discentes da Universidade Federal da Bahia (Bolsistas de Iniciação Científica).
[3] Departamento de Ciências Exatas e da Terra, Universidade Estadual da Bahia, Salvador, Bahia.

**Setembro de 2015**

**Agradecimentos**

Este trabalho consiste em um dos produtos de projetos de pesquisa financiados pela Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB), Termo de Outorga n$^{o}$.4895/2012, e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (Processo: 474832/2012-0). Pétala G.S.E. Tuy e Juan M. Leroux foram bolsistas FAPESB do PIBIC-UFBA, e Marcus E.S. Freire foi bolsista CNPq do PIBIC-UFBA.

**LISTA DE ILUSTRAÇÕES**

**SUMÁRIO**

Apêndices

# ANÁLISE DE CLASSES LATENTES:

# Um Tutorial usando Software Estatístico

## 1 Introdução

A análise de classes latentes (LCA, *Latent Class Analysis*, em inglês) é usada para identificar subgrupos, tipos ou categorias de indivíduos de uma população em estudo e permite identificar padrões de resposta com base em características observadas, relacionando-as a um conjunto de classes latentes. A variável latente, ou construto, não é observada diretamente, mas sim mensurada indiretamente através de duas ou mais variáveis observadas (Collins e Lanza, 2010). LCA oferece uma abordagem para lidar com tabelas de contingência grandes e complexas. Aplicações em diversas áreas podem ser encontradas na literatura. A LCA tem sido bastante utilizada para caracterizar fenômenos nas ciências sociais, do comportamento e saúde, que podem ser representados por modelos com distintos subgrupos, tipos, ou categorias de indivíduos.

Uma das maiores dificuldades encontradas para disseminação do uso dos métodos relacionados à análise de classes latentes é a limitada literatura sobre como pode ser feita sua implementação em *softwares* estatísticos de maneira didática. Dentre os *softwares* que oferecem suporte para a análise de classes latentes estão o **MPlus** (Muthén e Muthén,1998-2010), o **R** (R Development Core Team, 2014) e o **WinLTA** (The Methodology Center, Penn State, 2002).

O objetivo deste trabalho é apresentar, passo-a-passo, a implementação do modelo básico usando LCA em 3 diferentes softwares: **Mplus** (Version 5.21), **R** (Version 3.03) e **WinLTA**(Version 3.1). No *software* **R**, a LCA pode ser implementada usando quatro bibliotecas distintas, que são apresentadas neste tutorial. A implementação da metodologia é feita através de uma aplicação relacionada ao diagnóstico de carcinoma no colo do útero, cujos dados estão disponibilizados em uma das bibliotecas do **R**.

## 1.1 Aplicação: Estudo sobre Carcinoma de Colo de Útero

Os dados utilizados e analisados ao longo deste tutorial são provenientes da biblioteca ***poLCA*** do *softwar*e **R**. Estes dados referem-se a resultados sobre a presença ou ausência de carcinoma no colo do útero (classificação dicotômica) para 118 amostras, baseados na avaliação feita por sete patologistas. Patologistas são rotulados de A a G. Foram observados 20 diferentes padrões de resposta. O banco de dados contém 118 observações com 7 variáveis categóricas que representam os resultados fornecidos por cada patologista (código: 1="não"; 2="sim"). O modelo de classes latentes para estes dados pode ser representado esquematicamente em forma de diagrama (Figura 1.1).

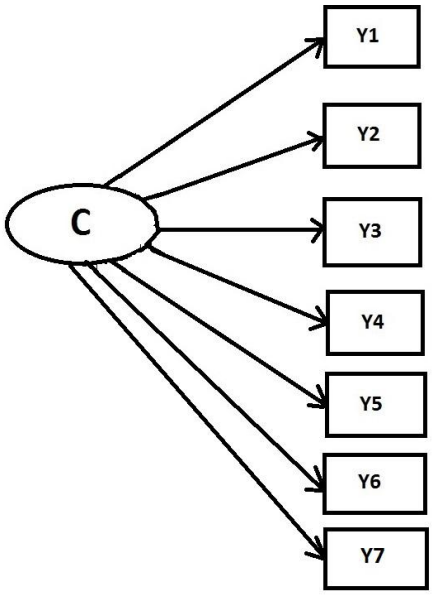


Figura1.1 Diagrama de caminhos para os dados de Carcinoma.

Neste exemplo o modelo de classes latentes (LCA) possui 7 indicadores binários (*Y*) e uma variável latente (*C*), também categórica, que referem-se, respectivamente, à avaliação de cada patologista e ao diagnóstico final de carcinoma.

## 2. Análise de Classes Latentes no Software MPlus

Este tutorial apresenta a implementação da análise de classes latentes (LCA, em inglês) no software **Mplus**, versão 5.21 (Muthén e Muthén,1998-2010). A versão demo mais recente pode ser encontrada no website *http://www.statmodel.com/demos.html*. O **Mplus** é um programa de modelagem estatística que fornece aos pesquisadores uma ferramenta flexível para analisar seus dados na abordagem com variáveis latentes. O Mplus incorpora uma ampla variedade de modelos, estimadores e algoritmos, além de apresentar uma interface fácil de usar e exibir uma variedade de gráficos e resultados para análise dos dados.

O **Mplus** é um *software* voltado especialmente para a modelagem de dados. O propósito da modelagem é descrever um conjunto de relações entre as variáveis de modo que seja compreensível e interpretável. Dentre os diversos tipos de modelagem implementados no **Mplus**, este tutorial apresenta a LCA, que tenta descrever as relações entre variáveis indicadoras e variáveis latentes categóricas.

A LCA pode ser definida como um caso especial da *Modelagem de Mistura*, referindo-se à modelagem com variáveis latentes categóricas. Estas representam subpopulações nas quais a associação entre os indicadores populacionais não é conhecida, mas é inferida a partir dos dados, sendo que as classes latentes tentam explicar as relações entre as variáveis, de maneira semelhante à análise fatorial.

### 2.1 Interface do *software*

O **Mplus** apresenta uma interface amigável (Figura 2.1), com uma caixa de texto onde pode ser digitada a sintaxe apropriada, que denominaremos *script*. Esta sintaxe pode ser também definida através do gerador de linguagem do *software*, que é encontrado na barra de ferramentas, como uma opção do item *Mplus*.

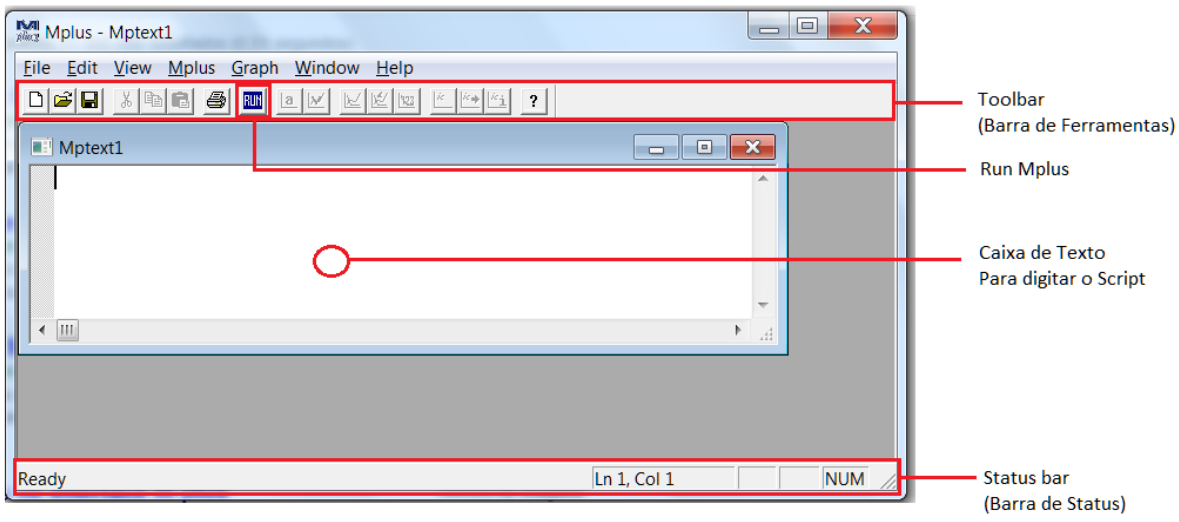


Figura 2.1 Interface do *software* MPLUS

O gerador de linguagem (*Language Generator*) é uma ferramenta do **Mplus** que pode auxiliar na definição do *script*, resultando em um arquivo com extensão *.INP*. Como mencionado anteriormente, o tipo de modelagem em que a LCA se encaixa nas definições do **MPlus** é o *Mixture*. Então, se utilizarmos o gerador de linguagem escolhemos esta opção para organizar o *script* para implementação de LCA (Figura 2.2).

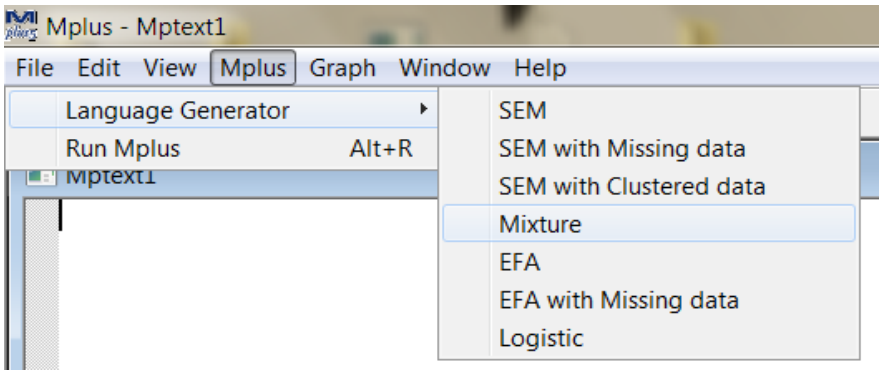


Figura 2.2 Seleção do tipo de modelagem para iniciar o gerador de linguagem.

Após esta etapa, uma janela é aberta para a definição do titulo do *script* (*Title*, maiores informações são apresentadas posteriormente neste tutorial) e seleção da base de dados (*Data File*), com a indicação do diretório onde o mesmo se encontra (Figura 2.3). Ao completar estas telas, o usuário deve escolher a opção "*Seguinte*" na parte inferior da tela.

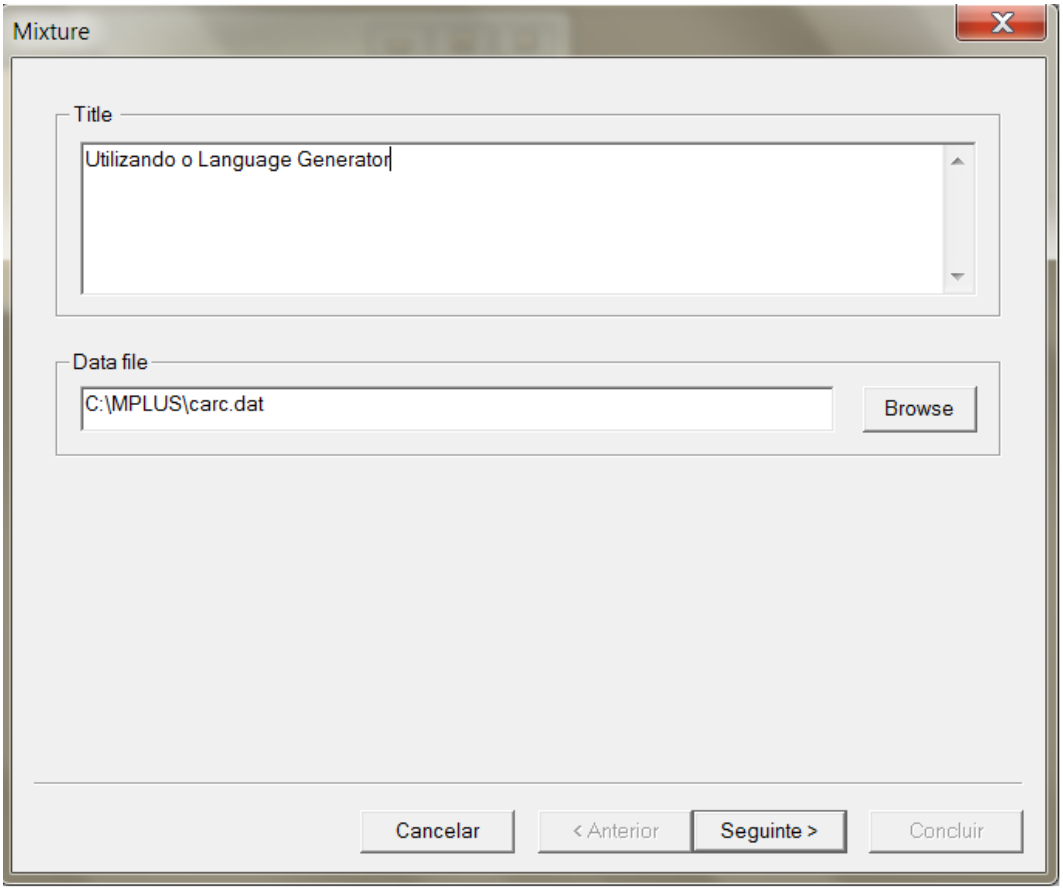


Figura 2.3 Definindo um título para o *script* e indicando o diretório do banco de dados.

As opções para seleção do formato do banco de dados estão disponíveis na Figura 2.4. Por exemplo, o formato livre (*Free*) indica que os dados estão separados livremente por espaços, vírgulas ou tabulação. Outro formato é o fixo (*Fixed*) que indica que os dados estão em um formato que requer a descrição da sua estrutura de separação das colunas. Pode-se ainda identificar a presença de dados faltantes (*Missing data*), se necessário. O arquivo contendo o banco de dados tem preferencialmente extensão *.txt* ou *.dat*.

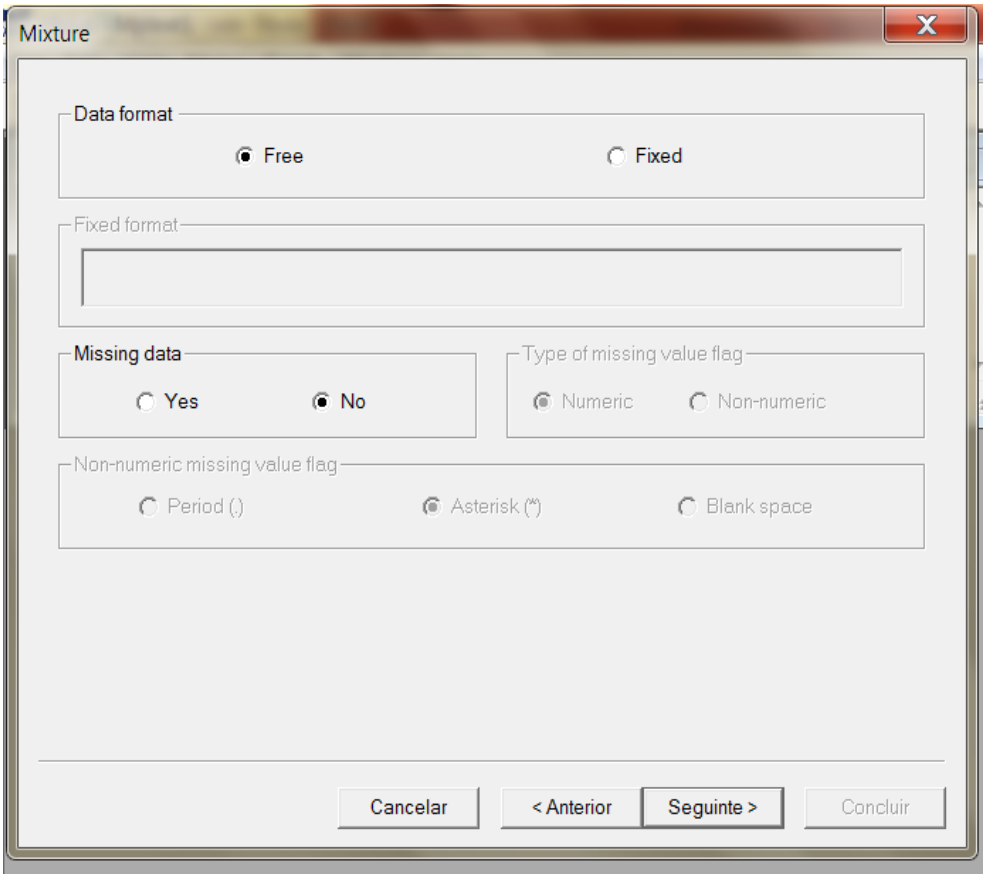


Figura 2.4 Especificando o formato do banco de dados.

É importante notar que o banco de dados não deve conter o nome das variáveis na sua primeira linha e os dados devem estar codificados como números inteiros não negativos sequenciais. Dados faltantes (*missing)* podem ser codificados de diferentes formas como, por exemplo, " " ou "999".

O próximo passo é especificar as variáveis observadas (*Variable list*), o nome e a quantidade de classes da variável latente (Figura 2.5). Para identificação de todas as variáveis contidas no banco de dados, deve-se digitar o nome de cada variável (*Variable names*) e depois clicar na opção *Add*. À medida que estas variáveis são adicionadas, seus correspondentes nomes são listados na janela *Variable list*. Caso seja necessária a exclusão de alguma variável, deve-se selecioná-la na *Variable list* e depois clicar em *Remove*. Se há interesse em modificar a ordem das variáveis nesta listagem pode-se usar as opções *Move up* (para movê-la para cima da lista) ou *Move down* (para movê-la para baixo da lista). Na janela *Latent class variable*, especifica-se o nome da variável latente (*Variable name*) e o número de classes latentes (*Number of classes*) (Figura 2.5).

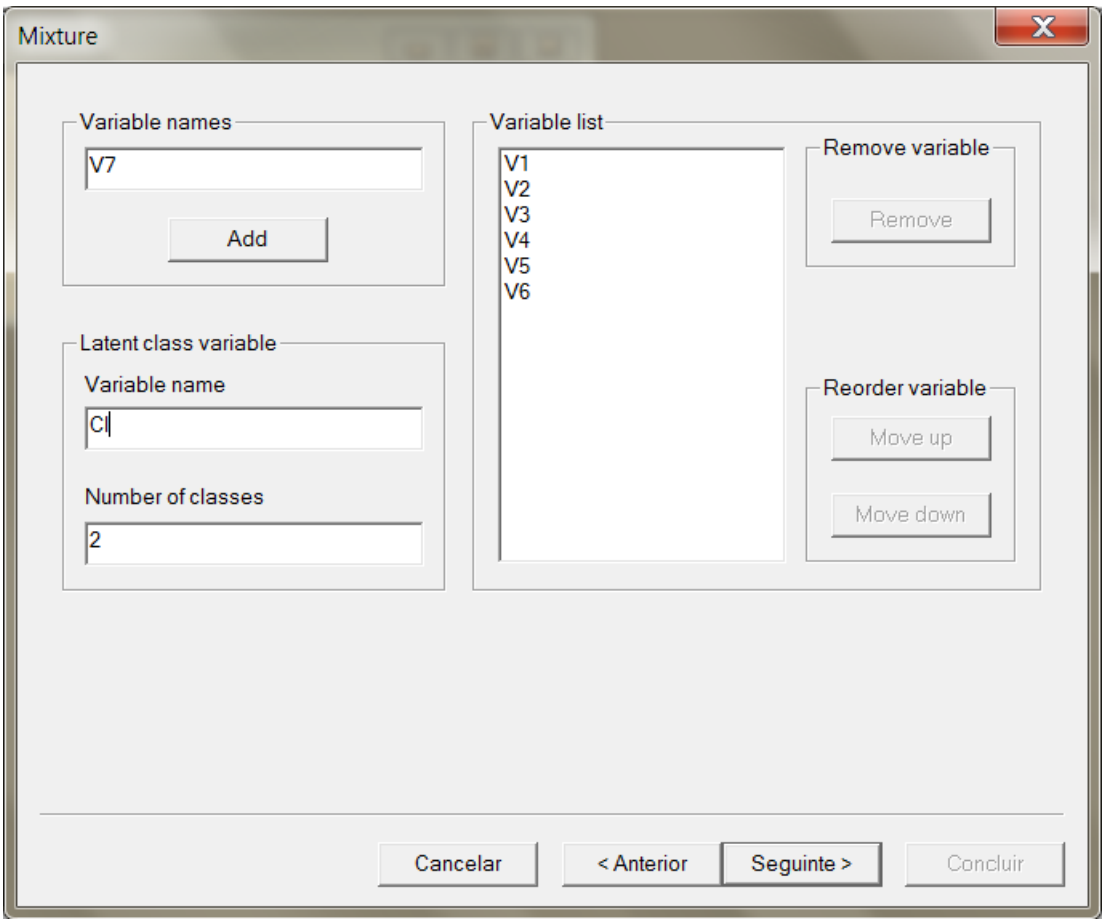


Figura 2.5 Especificando o número de classes latentes e as variáveis indicadoras.

A seguir as variáveis indicadoras a serem utilizadas na LCA (*Usevariable list*) devem ser selecionadas para definir a variável latente (Figura 2.6).

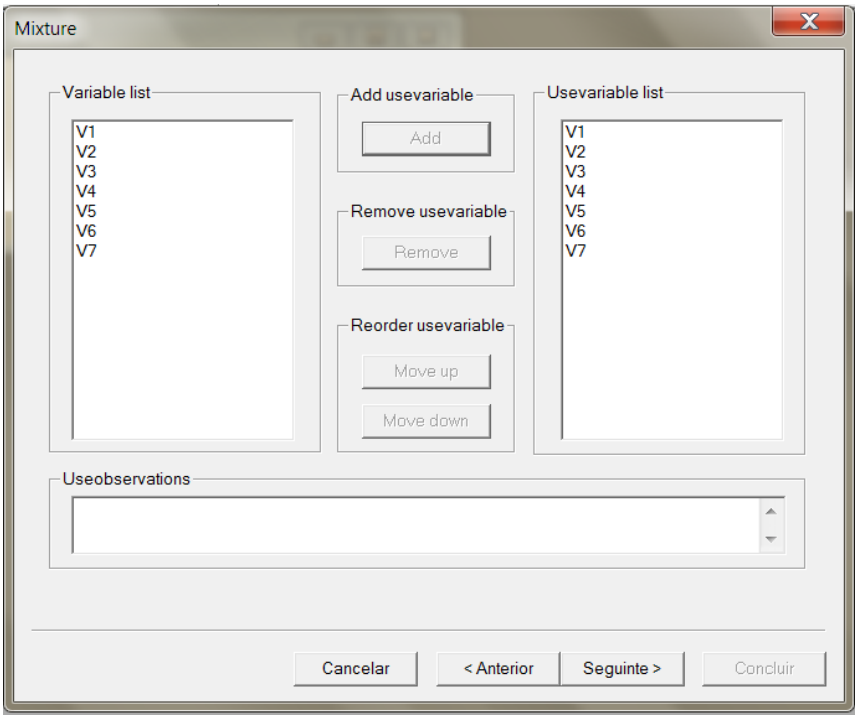


Figura 2.6 Especificando as variáveis observadas a serem utilizadas no modelo.

Na Figura 2.7 as variáveis indicadoras selecionadas são identificadas como variáveis categóricas, através de seleção individual ou de todas as variáveis, e, posteriormente, deve-se clicar no botão *Select/Remove* da janela *Latent class indicator*.

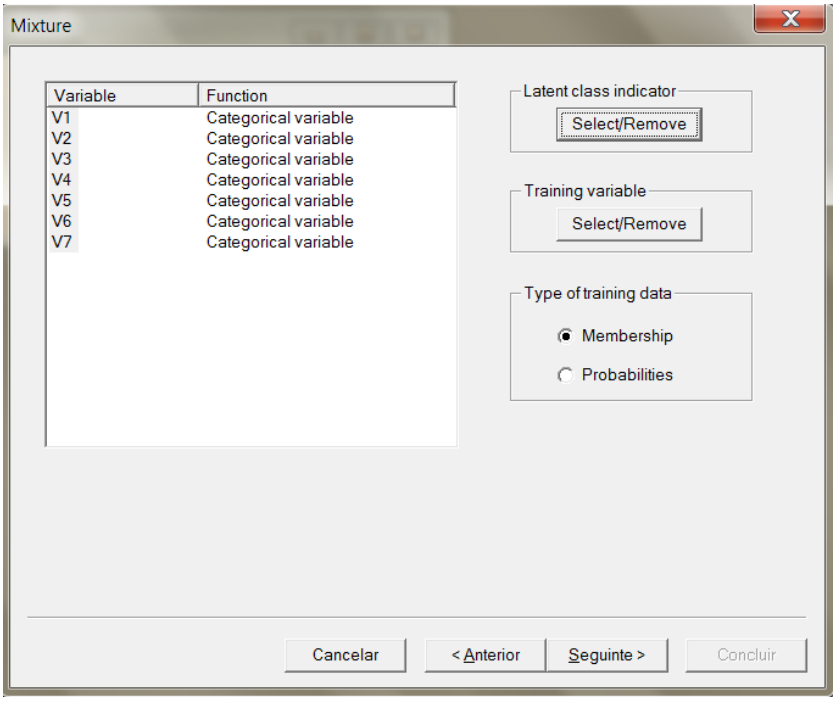


Figura 2.7 Informando quais dos indicadores são variáveis categóricas.

A opção "*training data*" é de interesse em algumas aplicações em especial, como a de avaliação de efeito de tratamento que envolve questões relacionadas à aderência ao tratamento. Este tipo de opção deve ser usado quando o pertencimento às classes latentes é conhecido para alguns dos indivíduos da amostra. Neste caso, é necessário incluir uma variável binária (0 ou 1) para definir o pertencimento ou não dos indivíduos em cada classe latente. Estas opções não se aplicam a nosso exemplo e por este motivo não foram selecionadas na Figura 2.7.

O próximo passo é definir o processo de estimação da LCA. As opções *default* estão apresentadas na Figura 2.8.

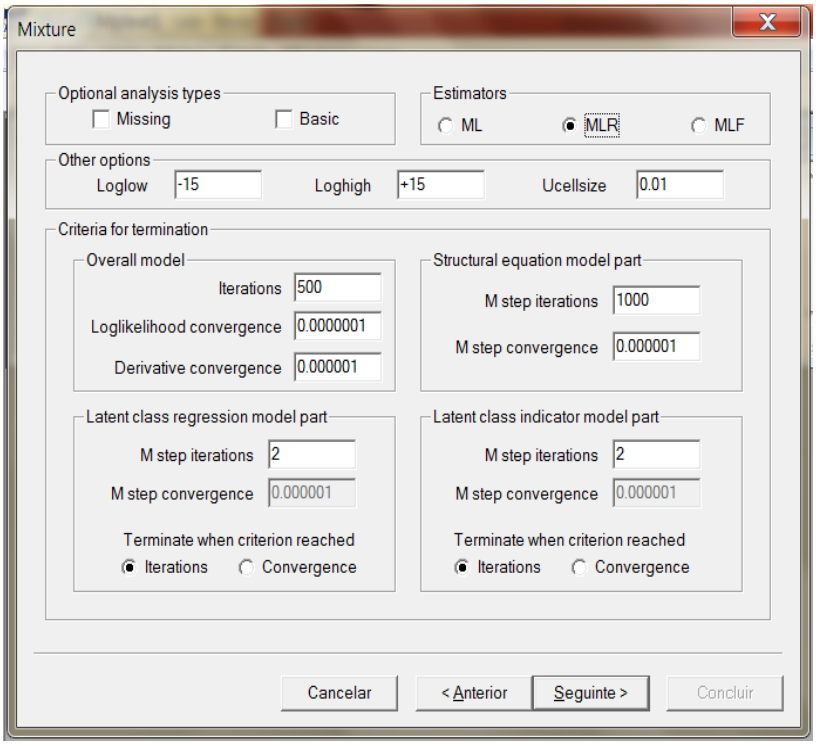


Figura 2.8 Opções *default* para o processo de estimação da LCA.

Alguns resultados específicos podem ser selecionados para serem apresentados no *OUTPUT* do **MPlus** (Figura 2.9). Em geral, estas não são opções comumente usadas em LCA. Por isto não faremos seleções adicionais neste tutorial.

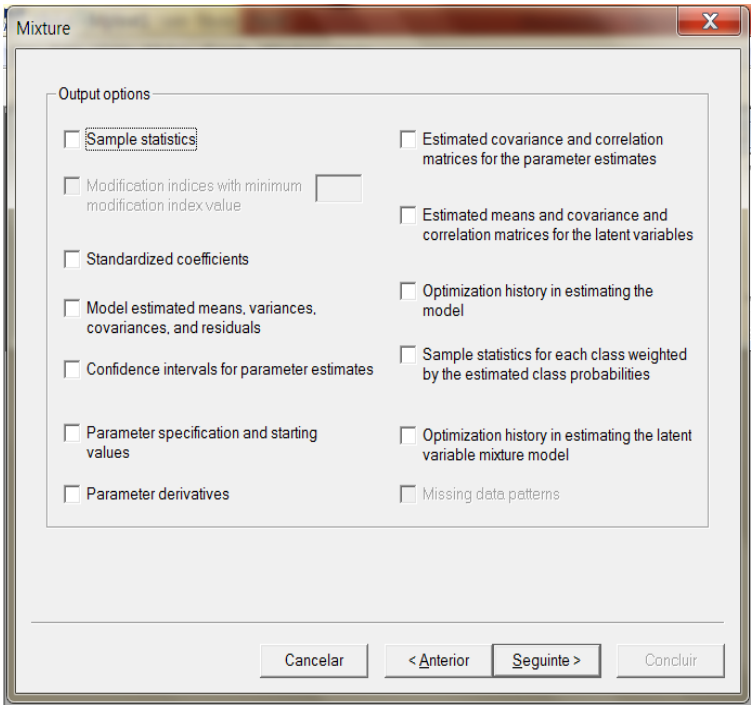


Figura 2.9 Resultados específicos a serem solicitados no **MPlus**.

O próximo passo é especificar quais dos resultados das análises deseja-se salvar em um arquivo de texto (*.txt*) (Figura 2.10). No nosso exemplo, não solicitamos a criação deste arquivo.

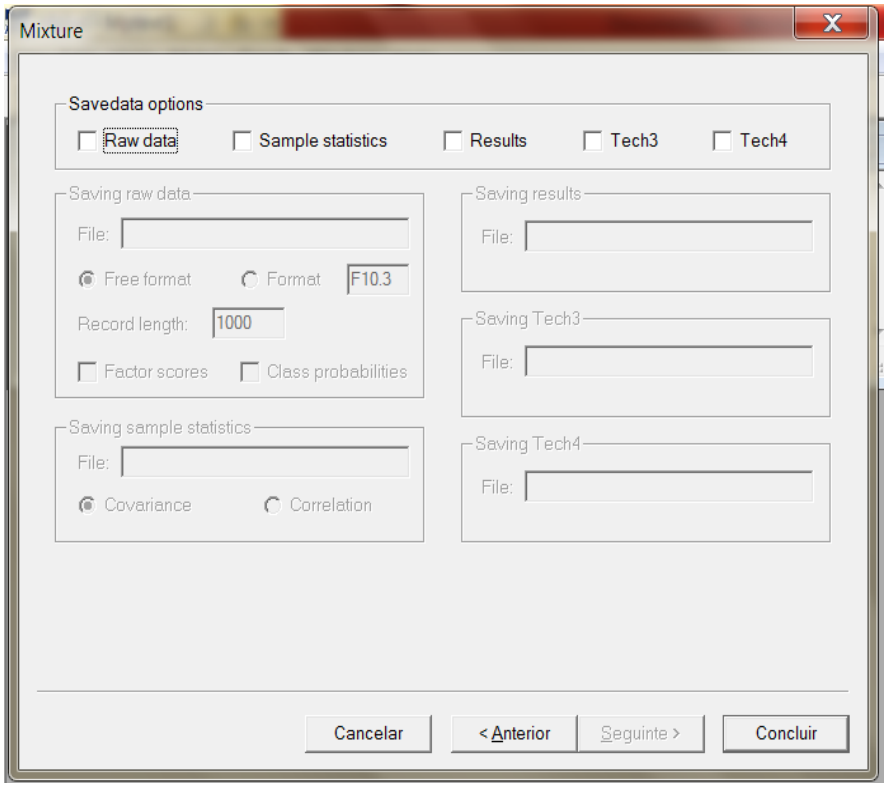


Figura 2.10 Quadro de opções para dados a serem salvos em arquivo txt.

Ao selecionar "*Concluir*" na Figura 2.10 um arquivo com extensão *.INP* é gerado contendo uma parte das especificações necessárias para implementação da LCA. O *script* produzido pelo *Language Generator*, conforme descrito anteriormente, é apresentado na Figura 2.11 e pode ser usado para obter resultados básicos da LCA.

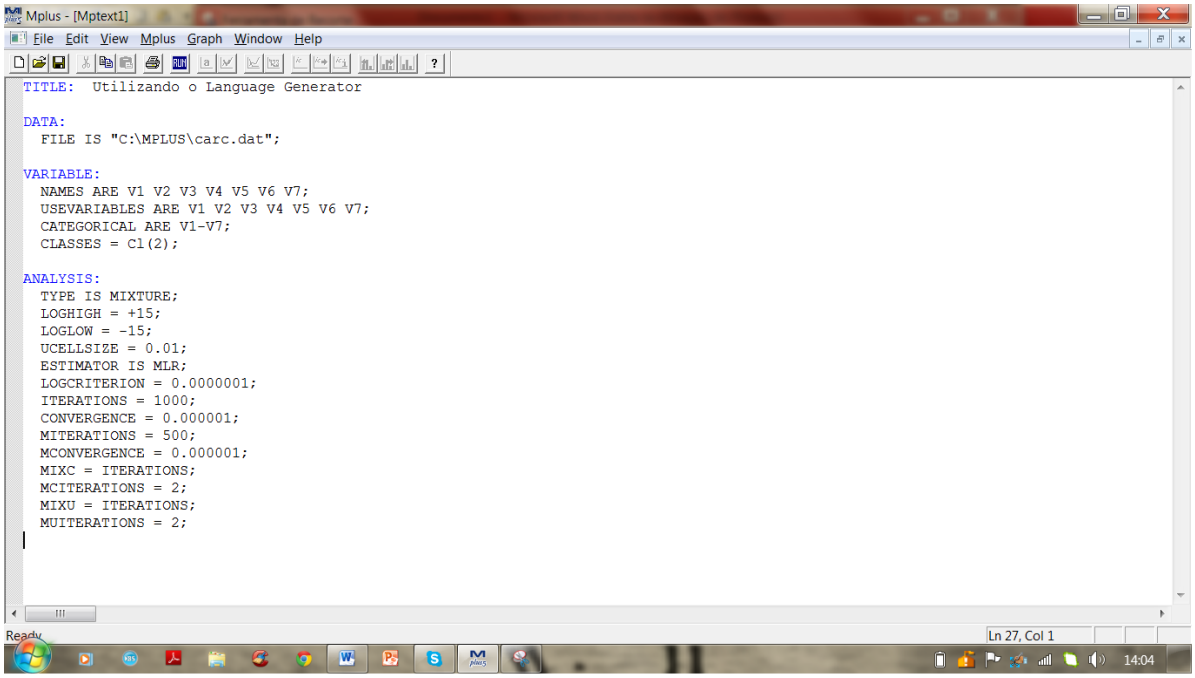


Figura 2.11 Script produzido pelo *Language Generator*.

A linguagem de programação do **MPlus** que pode ser usada para definição de modelos mais gerais e/ou para obtenção de resultados mais específicos, que não são fornecidos pelo *Language Generator*, são apresentados com mais detalhes a seguir. Nem todas as especificações no módulo *ANALYSIS*, que são apresentadas na Figura 2.11, são necessárias de serem especificadas na sintaxe usando linguagem de programação porque fazem parte das opções *default* do software. Há necessidade de especificá-las somente se houver interesse em alterar estas opções.

## 2.2 Linguagem de Programação

As etapas descritas anteriormente podem ser substituídas pela definição manual da linguagem de programação do *script*. A sintaxe do **Mplus** consiste em um conjunto de 10 comandos básicos, sendo que cada um deles tem variadas opções. Cada comando e suas especificações principais são apresentados a seguir:

- TITLE (Opcional)
  - Especifica o título que o **Mplus** vai imprimir em cada página do arquivo de saída (*output*).
  - Pode conter letras e símbolos, exceto as palavras usadas como comandos.
  - Pode usar várias linhas contanto que tenha, no máximo, 80 caracteres por linha.
  - Exemplo:

```
TITLE: Modelagem com variáveis latentes;
```

- DATA (Obrigatório)
    - É usado para fornecer informações sobre o conjunto de dados.
    - Especifica o arquivo de dados no formato ASCII, que não deve conter mais do que 500 variáveis.
    - O banco de dados deve estar em formato *free* ou *fixed*, com opções *tab-delimited, space-delimited* ou *comma-delimited*.
        - Sugestão: colocar o banco de dados no mesmo diretório do arquivo com a sintaxe do Mplus (*input file*). Caso contrário, é necessário especificar o endereço completo do diretório onde se encontra o banco de dados.
    - Exemplo:

```
DATA: FILE=meusdados.dat;
```

- VARIABLE (Obrigatório)
    - Usado para fornecer informações sobre as variáveis do conjunto de dados;
    - É formado por uma série de comandos com especificações fundamentais para a análise de dados:
        - NAMES: usado para atribuir nomes às variáveis do conjunto de dados;
            - Variáveis devem ser listadas na mesma ordem em que estão nos bancos de dados originais (STATA, R, SAS, SPSS).
            - Os nomes das variáveis não podem ter mais que 8 caracteres.
        - USEVARIABLES: indica quais variáveis serão utilizadas na análise.
        - CLASSES: usado para atribuir nomes às variáveis latentes categóricas e também para especificar o número de classes para cada variável latente categórica;
        - CATEGORICAL = para especificar quais variáveis indicadoras devem ser tratadas como categóricas (nominais);
        - AUXILIARY (não é obrigatório): serve para especificar as variáveis que não fazem parte da análise para definição da variável latente, mas para as quais as igualdades de médias entre as classes latentes serão testadas e podem também ser usadas posteriormente como probabilidade-base da imputação múltipla. A letra *e*, em parênteses, é acrescentada para solicitar o teste de igualdade de médias entre as classes latentes.
    - Exemplo: a variável latente *c* tem duas classes latentes e as variáveis *x* são contínuas:

```
NAMES ARE u1-u4 x1-x10;
USEVARIABLES = u1-u4;
CLASSES = c (2);
CATEGORICAL = u1-u4;
AUXILIARY = x1-x10 (e);
```

- DEFINE (Usado em situações particulares)
    - Este comando é usado para transformar variáveis existentes e criar novas variáveis. Este comando não é usado neste tutorial.

- ANALYSIS (Requerido para algumas análises)
    - A opção TYPE é utilizada para identificar o tipo de análise a ser executada;

- Ao selecionar TYPE=MIXTURE, um modelo de mistura vai ser estimado.
- Exemplo:

```
ANALYSIS:
TYPE = MIXTURE;
```

- MODEL: (Requerido para algumas análises)
  - Descreve o modelo a ser estimado. Não é necessário para estimação da LCA básica. Este comando não é usado neste tutorial.

- OUTPUT: (Muitas opções disponíveis)
  - Controla as informações fornecidas pelo Mplus;
  - Criação de um arquivo com a extensão .*OUT* que contém todos os resultados das análises.
  - Exemplos:

```
OUTPUT:
TECH1   TECH8  TECH10  TECH11  TECH14;
```

No manual do MPlus há uma descrição detalhada das informações contidas em cada TECH. Alguns resultados de interesse na LCA estão disponíveis em TECH11 e TECH14. As opções de resultados fornecidos no Mplus podem variar de acordo com a versão do software.

- SAVEDATA: (Usado em algumas aplicações)
  - Cria um arquivo de dados, contendo algumas estatísticas adicionais da análise realizada no MPlus.
  - CPROB: Adiciona no arquivo a probabilidade de pertencimento a cada classe e a identificação da classe a que pertence cada indivíduo;
  - Exemplo:

```
SAVEDATA:
    file is lca_save.txt ;
    save is cprob;
    format is free;
```

- PLOT: (Usado em algumas aplicações)
  - É utilizado para solicitar exibições gráficas de dados observados e de resultados da análise. Estes *displays* gráficos podem ser visualizados quando a análise for concluída.
  - O comando PLOT oferece histogramas (*histograms*), gráficos de dispersão (*scatterplots*), gráficos de valores observados e estimados, gráficos de médias estimadas e proporções/probabilidades, probabilidades estimadas para uma variável latente categórica em função de seus indicadores.
  - Os *displays* gráficos podem ser editados e exportados com extensões DIB, EMF, ou JPEG. Além disso, os dados de cada *display* gráfico podem ser salvos em um arquivo externo para uso por outro programa gráfico.
  - Exemplo:

```
PLOT:
TYPE=PLOT3;
SERIES IS v1(1) v2(2) v3(3) v4(4) v5(5) v6(6) v7(7);
```

- MONTECARLO: (Usado em simulações)
    - Usado para conduzir simulações de Monte Carlo, tanto para a geração de dados como para análise de dados. Este comando não é usado neste tutorial.

Note a inclusão de ponto-e-vírgula ao final de cada linha de comando. A sintaxe para implementação de LCA no MPlus é apresentada na Figura 2.12. Para executar os comandos deve-se clicar no botão com o nome RUN (Figura 2.1). Outra opção é clicar na opção *RunMplus* (Figura 2.12) ou ainda usar o comando *ALT+R* no teclado do computador.

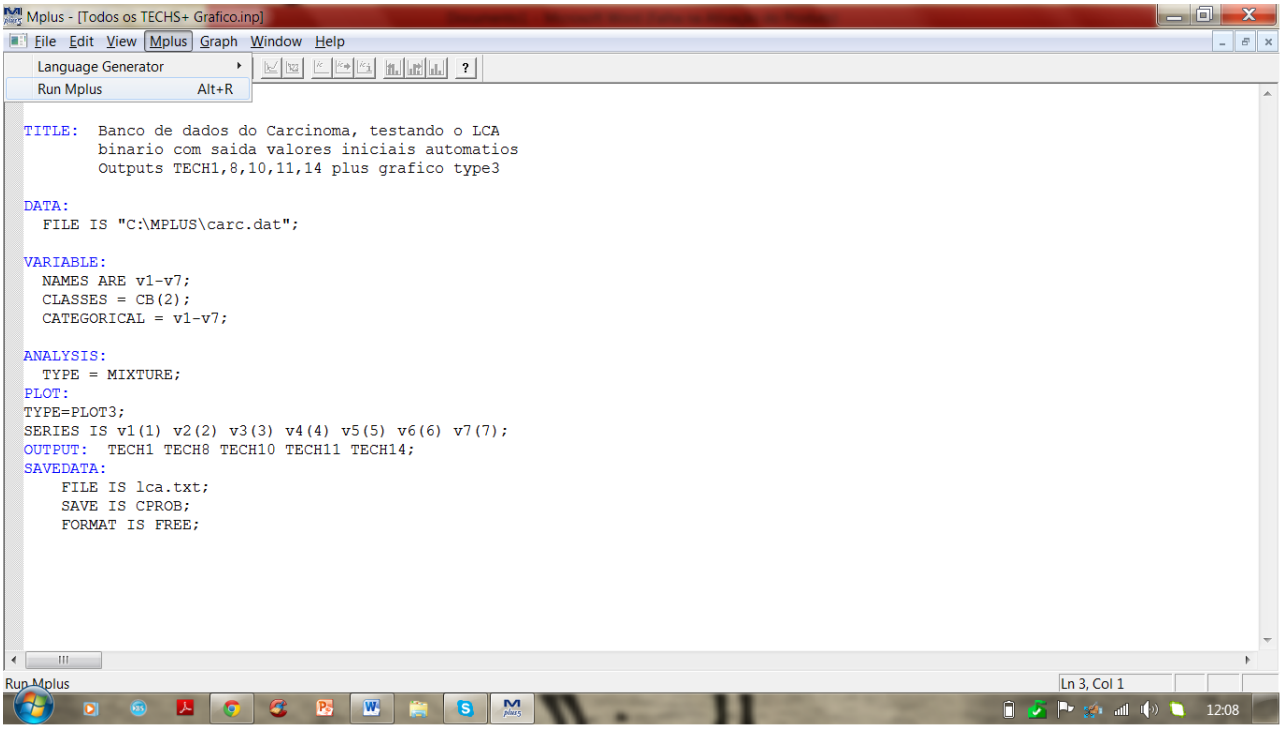


Figura 2.12 *Script* básico para ajuste de LCA no exemplo Carcinoma.

### 2.3 Resultados (*outputs*) principais da LCA

Após executar os comandos para implementação da LCA, os seguintes resultados (versão reduzida) são apresentados e salvos em um arquivo com extensão *.OUT*. Os resultados mais básicos incluem, nesta ordem, a repetição do *script* utilizado para análise dos dados, informações sobre os dados utilizados nas análises e opções utilizadas pelo *software*, distribuição de frequências para as variáveis indicadoras, estatísticas de bondade de ajuste do modelo, as probabilidades não condicionais estimadas por diferentes métodos, a entropia estimada no modelo e as probabilidades condicionais. Estes resultados para análise dos dados de carcinoma do colo de útero são listados a seguir para ilustração.

```
INPUT INSTRUCTIONS

  TITLE: Banco de dados do Carcinoma,
       valores iniciais automáticos com saídas padrões para LCA

DATA:
    FILE IS "C:\MPLUS\carc.dat";

  VARIABLE:
    NAMES ARE v1-v7;
    CLASSES = CB(2);
    CATEGORICAL = v1-v7;

  ANALYSIS:
    TYPE = MIXTURE;
  PLOT:
  TYPE=PLOT3;
  SERIES IS v1(1) v2(2) v3(3) v4(4) v5(5) v6(6) v7(7);
  OUTPUT:  TECH1 TECH8 TECH10 TECH11 TECH14;
  SAVEDATA:
      FILE IS lca.txt;
      SAVE IS CPROB;
      FORMAT IS FREE;



INPUT READING TERMINATED NORMALLY



Banco de dados do Carcinoma,
valores iniciais automáticos com saídas padrões para LCA

SUMMARY OF ANALYSIS

Number of groups                 1
Number of observations           118

Number of dependent variables    7
Number of independent variables  0
Number of continuous latent variables  0
Number of categorical latent variables  1
```

```
Observed dependent variables

  Binary and ordered categorical (ordinal)
   V1          V2          V3          V4          V5          V6          V7

Categorical latent variables
   CB


Estimator                                       MLR
Information matrix                              OBSERVED
Optimization Specifications for the Quasi-Newton Algorithm for
Continuous Outcomes
  Maximum number of iterations                  100
  Convergence criterion                         0.100D-05
Optimization Specifications for the EM Algorithm
  Maximum number of iterations                  500
  Convergence criteria
Loglikelihood change                            0.100D-06
    Relative loglikelihood change               0.100D-06
    Derivative                                  0.100D-05
Optimization Specifications for the M step of the EM Algorithm for
Categorical Latent variables
  Number of M step iterations                   1
  M step convergence criterion                  0.100D-05
  Basis for M step termination                  ITERATION
Optimization Specifications for the M step of the EM Algorithm for
Censored, Binary or Ordered Categorical (Ordinal), Unordered
Categorical (Nominal) and Count Outcomes
  Number of M step iterations                   1
  M step convergence criterion                  0.100D-05
  Basis for M step termination                  ITERATION
  Maximum value for logit thresholds            15
  Minimum value for logit thresholds            -15
  Minimum expected cell size for chi-square     0.100D-01
Optimization algorithm                          EMA
Random Starts Specifications
  Number of initial stage random starts         10
  Number of final stage optimizations           2
  Number of initial stage iterations            10
  Initial stage convergence criterion           0.100D+01
  Random starts scale                           0.500D+01
  Random seed for generating random starts      0
Link                                            LOGIT

Input data file(s)
  C:\MPLUS\carc.dat
Input data format  FREE

SUMMARY OF CATEGORICAL DATA PROPORTIONS

    V1
      Category 1    0.441
      Category 2    0.559
    V2
      Category 1    0.331
      Category 2    0.669
    V3
      Category 1    0.619
```

```
    Category 2    0.381
  V4
    Category 1    0.729
    Category 2    0.271
  V5
    Category 1    0.398
    Category 2    0.602
  V6
    Category 1    0.788
    Category 2    0.212
  V7
    Category 1    0.441
    Category 2    0.559

RANDOM STARTS RESULTS RANKED FROM THE BEST TO THE WORST LOGLIKELIHOOD VALUES

Final stage loglikelihood values at local maxima, seeds, and initial stage start numbers:

       -317.257          285380          1
       -317.257          195873          6

THE MODEL ESTIMATION TERMINATED NORMALLY

TESTS OF MODEL FIT

Loglikelihood

       H0 Value                              -317.257
       H0 Scaling Correction Factor      1.006
for MLR

Information Criteria

       Number of Free Parameters            15
Akaike (AIC)                       664.514
       Bayesian (BIC)                  706.074
       Sample-Size Adjusted BIC      658.655
         (n* = (n + 2) / 24)

Chi-Square Test of Model Fit for the Binary and Ordered Categorical
(Ordinal) Outcomes

       Pearson Chi-Square

       Value                          92.648
       Degrees of Freedom                112
       P-Value                        0.9084

       Likelihood Ratio Chi-Square

       Value                          62.366
       Degrees of Freedom                112
       P-Value                        1.0000

FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASSES
BASED ON THE ESTIMATED MODEL

  Latent
 Classes
```

| | | |
|---|---|---|
| 1 | 58.85692 | 0.49879 |
| 2 | 59.14308 | 0.50121 |

FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASS PATTERNS
BASED ON ESTIMATED POSTERIOR PROBABILITIES

| Latent Classes | | |
|---|---|---|
| 1 | 58.85692 | 0.49879 |
| 2 | 59.14308 | 0.50121 |

CLASSIFICATION QUALITY

| | |
|---|---|
| Entropy | 0.986 |

CLASSIFICATION OF INDIVIDUALS BASED ON THEIR MOST LIKELY LATENT CLASS MEMBERSHIP

Class Counts and Proportions

| Latent Classes | | |
|---|---|---|
| 1 | 59 | 0.50000 |
| 2 | 59 | 0.50000 |

Average Latent Class Probabilities for Most Likely Latent Class Membership (Row)
by Latent Class (Column)

| | 1 | 2 |
|---|---|---|
| 1 | 0.996 | 0.004 |
| 2 | 0.002 | 0.998 |

MODEL RESULTS

RESULTS IN PROBABILITY SCALE

Latent Class 1

| | | | | |
|---|---|---|---|---|
| V1 | | | | |
| Category 1 | 0.883 | 0.043 | 20.404 | 0.000 |
| Category 2 | 0.117 | 0.043 | 2.691 | 0.007 |
| V2 | | | | |
| Category 1 | 0.646 | 0.064 | 10.081 | 0.000 |
| Category 2 | 0.354 | 0.064 | 5.533 | 0.000 |
| V3 | | | | |
| Category 1 | 1.000 | 0.000 | 0.000 | 1.000 |
| Category 2 | 0.000 | 0.000 | 0.000 | 1.000 |
| V4 | | | | |
| Category 1 | 1.000 | 0.000 | 0.000 | 1.000 |
| Category 2 | 0.000 | 0.000 | 0.000 | 1.000 |
| V5 | | | | |
| Category 1 | 0.777 | 0.055 | 14.203 | 0.000 |
| Category 2 | 0.223 | 0.055 | 4.075 | 0.000 |
| V6 | | | | |

| | | | | |
|---|---|---|---|---|
| Category 1 | 1.000 | 0.000 | 0.000 | 1.000 |
| Category 2 | 0.000 | 0.000 | 0.000 | 1.000 |
| V7 | | | | |
| Category 1 | 0.883 | 0.044 | 20.260 | 0.000 |
| Category 2 | 0.117 | 0.044 | 2.672 | 0.008 |

Latent Class 2

| | | | | |
|---|---|---|---|---|
| V1 | | | | |
| Category 1 | 0.000 | 0.000 | 0.000 | 1.000 |
| Category 2 | 1.000 | 0.000 | 0.000 | 1.000 |
| V2 | | | | |
| Category 1 | 0.017 | 0.017 | 1.009 | 0.313 |
| Category 2 | 0.983 | 0.017 | 58.639 | 0.000 |
| V3 | | | | |
| Category 1 | 0.239 | 0.057 | 4.190 | 0.000 |
| Category 2 | 0.761 | 0.057 | 13.331 | 0.000 |
| V4 | | | | |
| Category 1 | 0.459 | 0.065 | 7.041 | 0.000 |
| Category 2 | 0.541 | 0.065 | 8.301 | 0.000 |
| V5 | | | | |
| Category 1 | 0.021 | 0.021 | 1.022 | 0.307 |
| Category 2 | 0.979 | 0.021 | 46.805 | 0.000 |
| V6 | | | | |
| Category 1 | 0.577 | 0.065 | 8.900 | 0.000 |
| Category 2 | 0.423 | 0.065 | 6.516 | 0.000 |
| V7 | | | | |
| Category 1 | 0.000 | 0.000 | 0.000 | 1.000 |
| Category 2 | 1.000 | 0.000 | 0.000 | 1.000 |

QUALITY OF NUMERICAL RESULTS

| | |
|---|---|
| Condition Number for the Information Matrix (ratio of smallest to largest eigenvalue) | 0.313E-01 |

Os resultados referentes às probabilidades não condicionais e condicionais estimadas pela LCA podem ainda ser sumarizados em forma de gráfico (Figura 2.13).

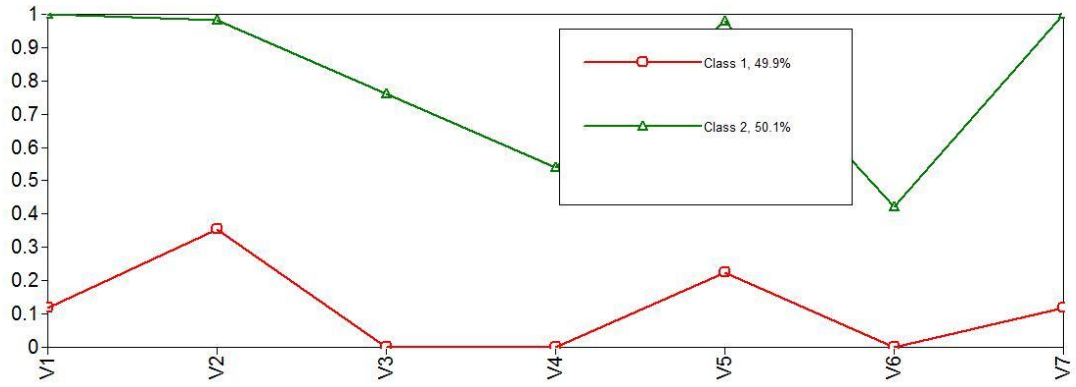


Figura 2.13 Gráfico das probabilidades condicionais e não condicionais estimadas por LCA nos dados de carcinoma.

O gráfico apresentado na Figura 2.13 pode ser construído de acordo com o seguinte procedimento, que é apresentado passo-a-passo:

1º) Na aba *Graph* escolha a opção *Viewgraphs* ou tecle ALT + V (Figura 2.14).

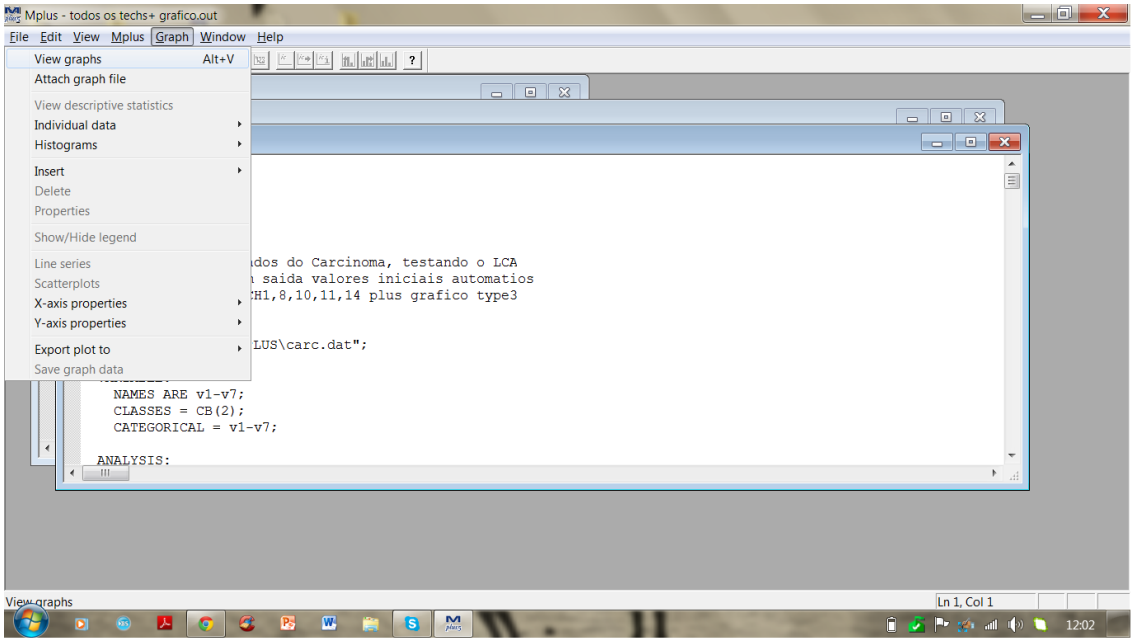


Figura 2.14 Solicitando a visualização de gráficos.

2º) Selecione a opção *Estimated Probabilities* (Figura 2.15). Pressione a opção *View*.

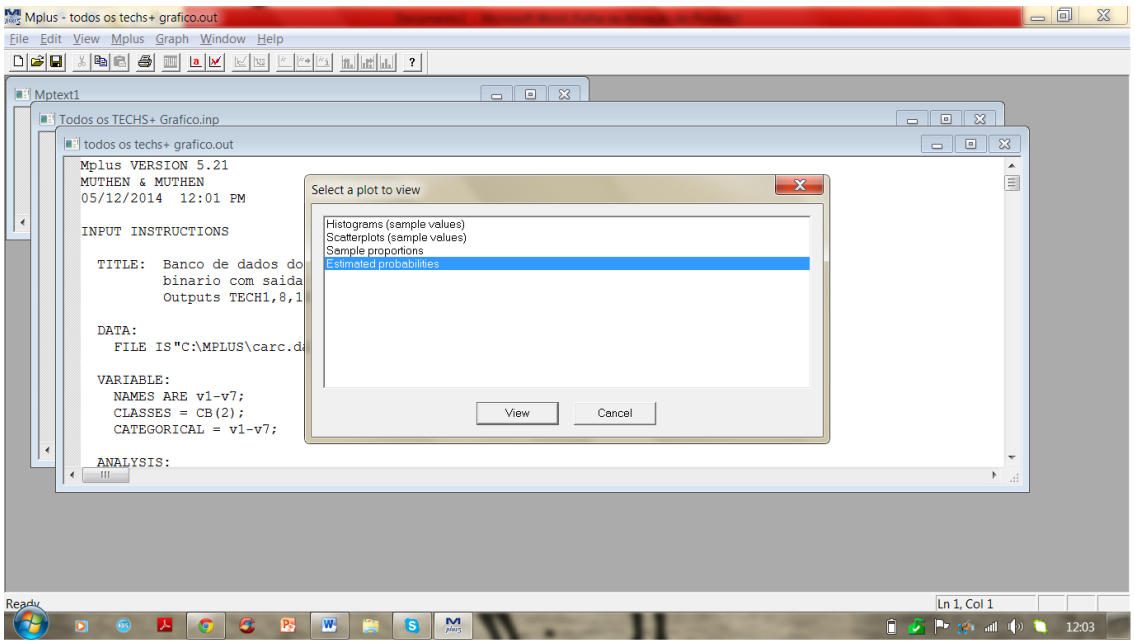


Figura 2.15 Selecionando o tipo de gráfico.

3º) Selecione a categoria de interesse para seus indicadores (*Probability of one category*). Neste exemplo, a categoria é 2 (sim) e pressione OK (Figura 2.16).

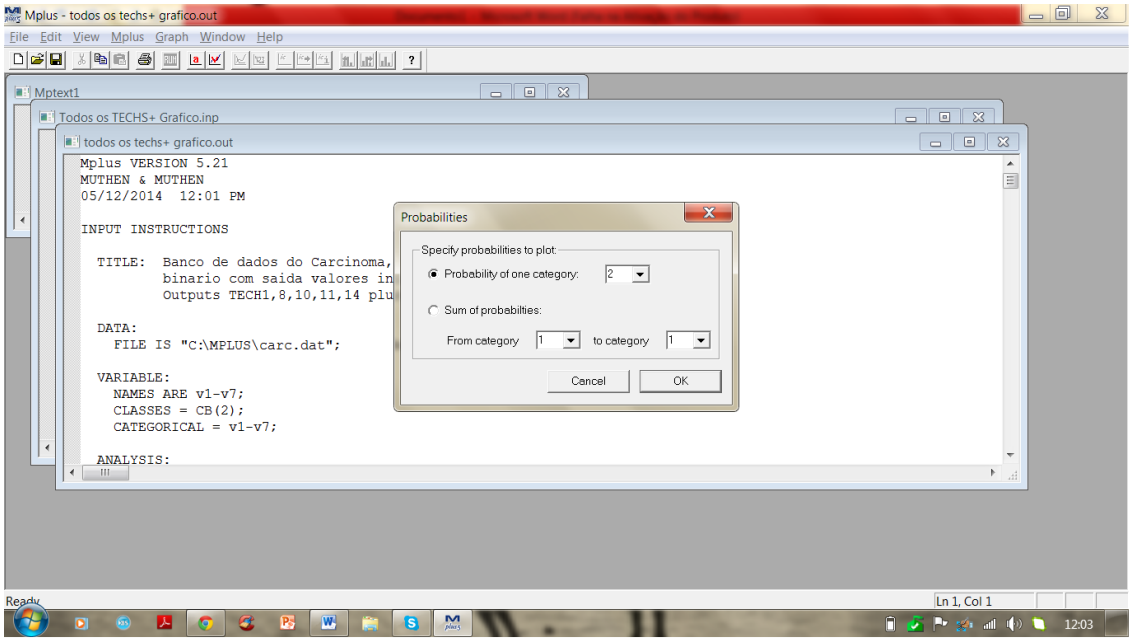


Figura 2.16 Especificação das probabilidades a serem apresentadas no gráfico (*default* é a categoria 2).

4º) Após o terceiro passo o gráfico será construído. No entanto, como as variáveis são categóricas o eixo X precisa ser reformatado com a opção *X-axis properties>Labels*, que é disponibilizada clicando o botão da direita do mouse sobre o gráfico (Figura 2.17).

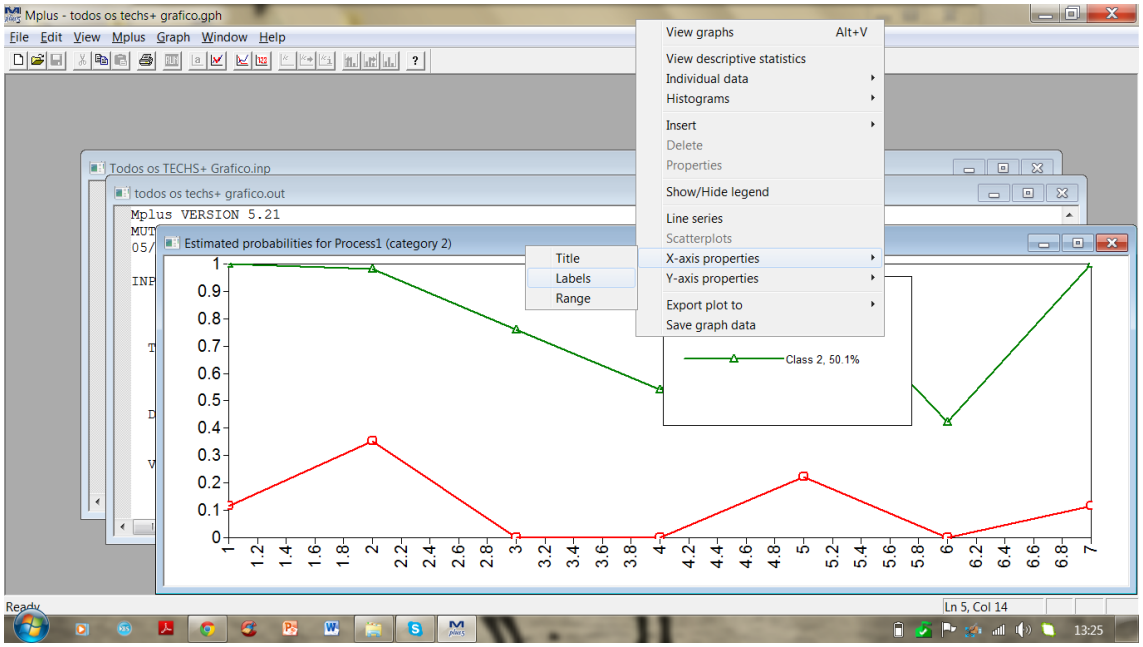


Figura 2.17 Modificando as abcissas do gráfico (parte 1).

5º) Na aba *Dialog*, clicar em *Customized* como opção para *Type of Labels*. Depois pressionar *OK* (Figura 2.18).

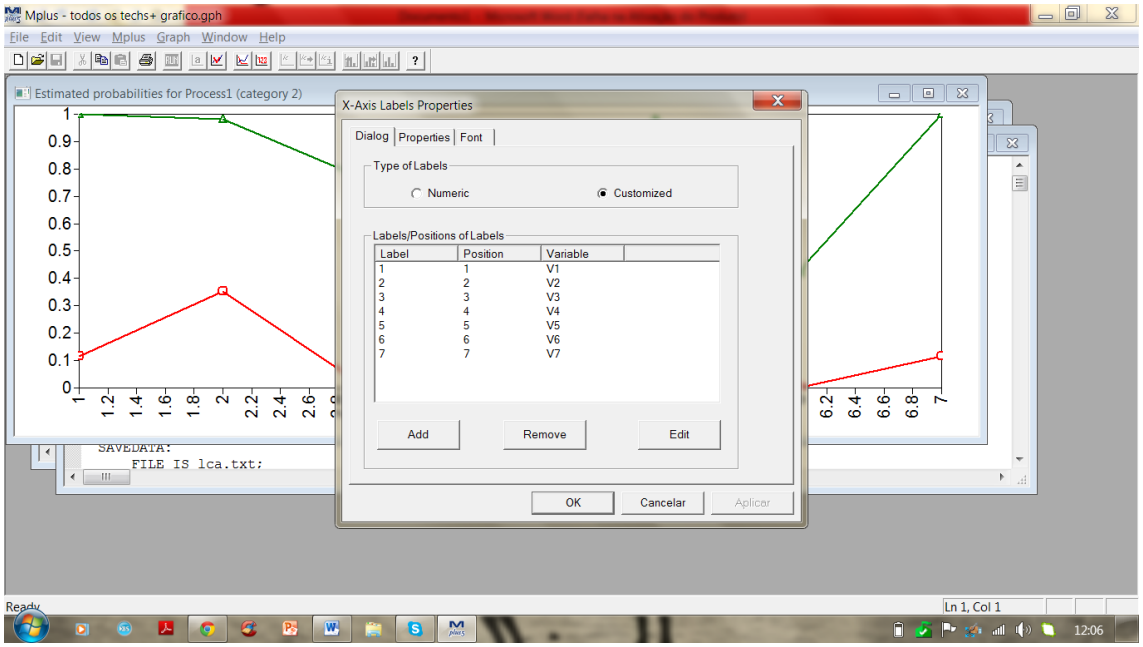


Figura 2.18 Modificando as abcissas do gráfico (parte 2).

Agora selecione o número 1 na coluna *Label*, conforme pode ser visto na Figura 2.18. Pressione o botão *Edit* para mudar o nome da variável (*Text*) (Figura 2.19).

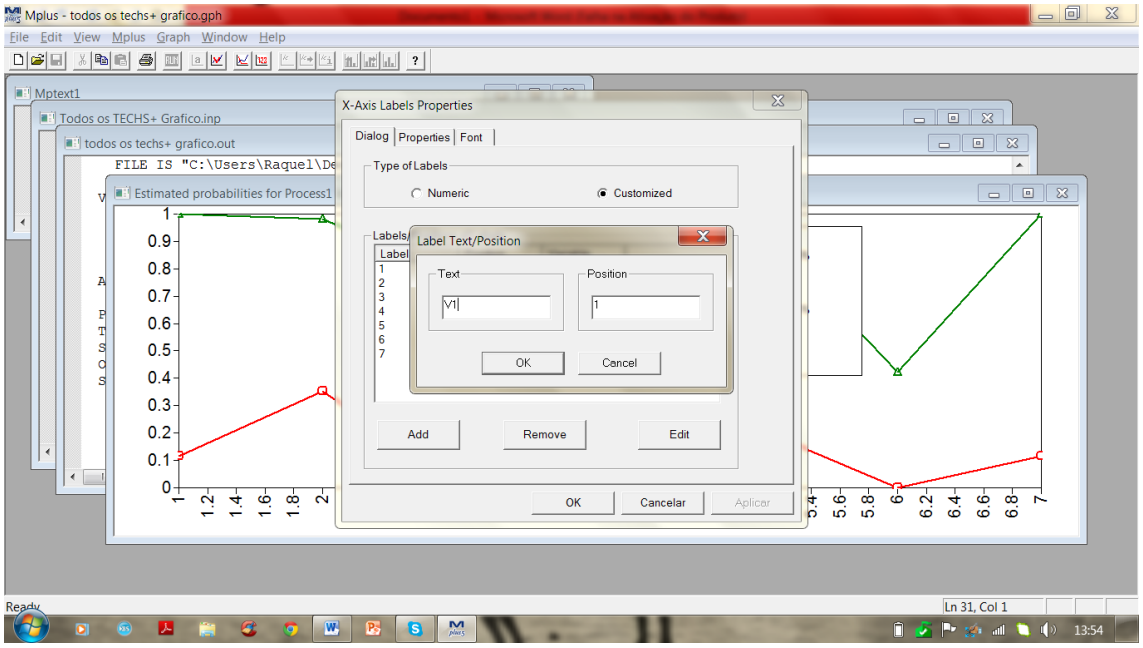


Figura 2.19 Modificando as abcissas do gráfico (parte 3).

Edite o nome de todas as variáveis, como mostra na Figura 2.20. Verifique que a primeira coluna denominada *Label* agora contém os nomes destas variáveis (V1, V2,..., V7).

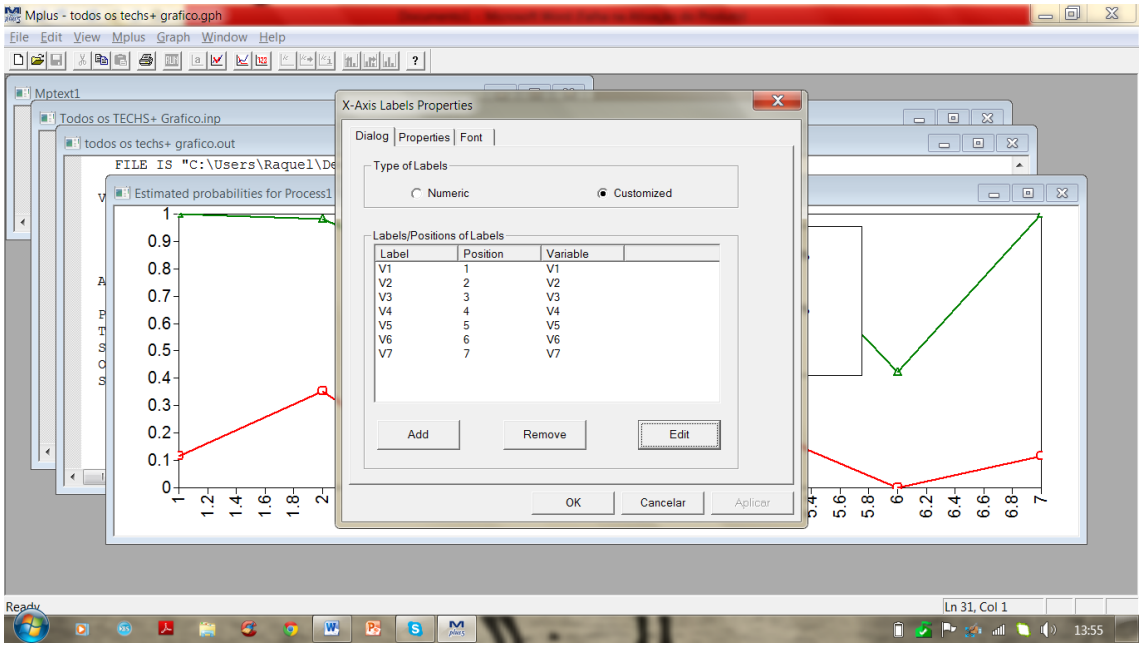


Figura 2.20 Modificando as abcissas do gráfico (parte 4).

O gráfico poderá ser exportado usando arquivos com extensão JPEG, EMF, DJB. Para isto, clique com o botão direito no gráfico e escolha a opção *Export plot to* para, em seguida, escolher o nome do gráfico (Figura 2.21).

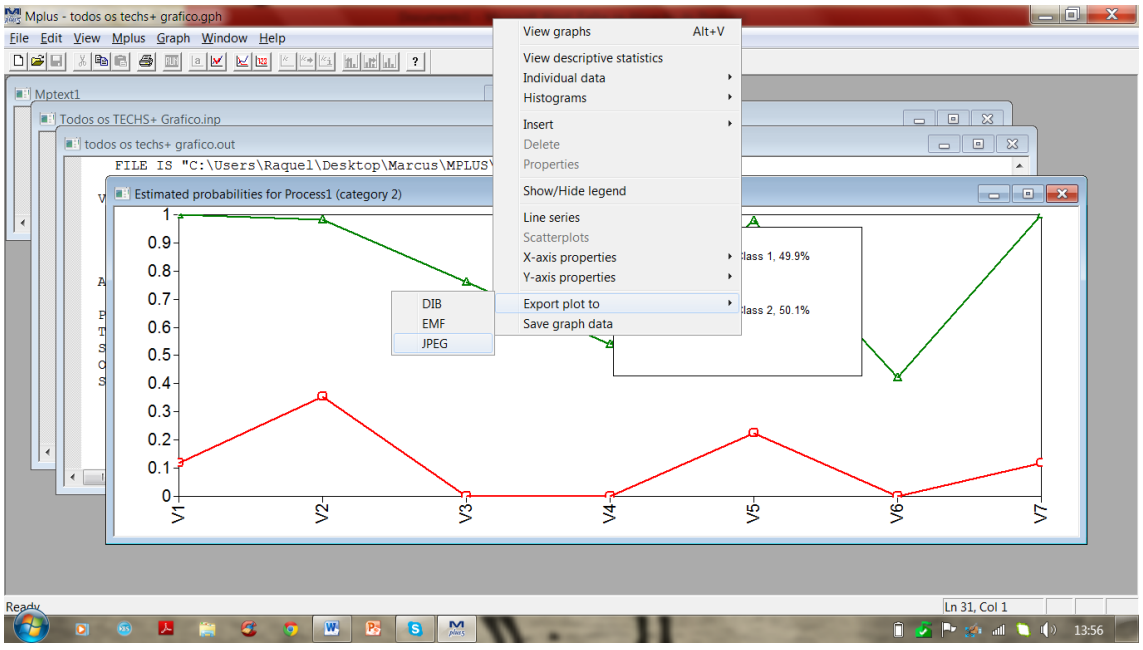


Figura 2.21 Opções para salvar o gráfico.

É importante mencionar que a caixa de texto com a legenda no gráfico da Figura 2.13 pode também ser editada, deslocada na área do gráfico e/ou ter seu tamanho alterado.

Outras opções de resultados a serem apresentados, denominadas TECH, podem também ser especificadas no *script* do Mplus. As mais utilizadas em LCA são TECH10,

TECH11 e TECH14, que são descritas a seguir. Um recorte com os resultados de maior interesse em cada TECH é também apresentado.

A opção TECH10 é usada para solicitar informações univariadas e bivariadas sobre o ajuste do modelo, bem como os padrões de respostas e suas frequências observadas e estimadas (esperadas), além de resíduos padronizados.

TECHNICAL 10 OUTPUT

MODEL FIT INFORMATION FOR THE LATENT CLASS INDICATOR MODEL PART

RESPONSE PATTERNS

| No. | Pattern | No. | Pattern | No. | Pattern | No. | Pattern |
|---|---|---|---|---|---|---|---|
| 1 | 0000000 | 2 | 0000100 | 3 | 0100000 | 4 | 0100001 |
| 5 | 0100100 | 6 | 0100101 | 7 | 1000000 | 8 | 1010101 |
| 9 | 1100000 | 10 | 1100001 | 11 | 1100100 | 12 | 1100101 |
| 13 | 1100111 | 14 | 1101001 | 15 | 1101101 | 16 | 1101111 |
| 17 | 1110101 | 18 | 1110111 | 19 | 1111101 | 20 | 1111111 |

RESPONSE PATTERN FREQUENCIES AND CHI-SQUARE CONTRIBUTIONS

| Response Pattern | Frequency Observed | Estimated | Standardized Residual (z-score) | Chi-square Contribution Pearson | Loglikelihood | Deleted |
|---|---|---|---|---|---|---|
| 1 | 34.00 | 23.05 | 2.54 | 5.20 | 26.43 | |
| 2 | 2.00 | 6.61 | -1.85 | 3.22 | -4.78 | |
| 3 | 6.00 | 12.65 | -1.98 | 3.50 | -8.95 | |
| 4 | 1.00 | 1.67 | -0.52 | 0.27 | -1.02 | |
| 5 | 4.00 | 3.63 | 0.20 | 0.04 | 0.78 | |
| 6 | 5.00 | 0.48 | 6.55 | 42.72 | 23.46 | |
| 7 | 2.00 | 3.04 | -0.60 | 0.36 | -1.67 | |
| 8 | 1.00 | 0.20 | 1.81 | 3.27 | 3.25 | |
| 9 | 2.00 | 1.67 | 0.26 | 0.07 | 0.73 | |
| 10 | 1.00 | 0.30 | 1.28 | 1.65 | 2.42 | |
| 11 | 2.00 | 0.48 | 2.20 | 4.84 | 5.72 | |
| 12 | 7.00 | 3.67 | 1.77 | 3.03 | 9.05 | |
| 13 | 1.00 | 2.64 | -1.02 | 1.02 | -1.94 | |
| 14 | 1.00 | 0.09 | 2.98 | 8.87 | 4.76 | |
| 15 | 2.00 | 4.25 | -1.11 | 1.19 | -3.02 | |
| 16 | 3.00 | 3.11 | -0.06 | 0.00 | -0.22 | |
| 17 | 13.00 | 11.47 | 0.48 | 0.20 | 3.25 | |
| 18 | 5.00 | 8.40 | -1.22 | 1.38 | -5.19 | |
| 19 | 10.00 | 13.52 | -1.02 | 0.92 | -6.04 | |
| 20 | 16.00 | 9.90 | 2.02 | 3.76 | 15.36 | |

THE TOTAL PEARSON CHI-SQUARE CONTRIBUTION FROM EMPTY CELLS IS 7.17

BIVARIATE MODEL FIT INFORMATION

| Variable | Variable | Estimated Probabilities H1 | H0 | Standardized Residual |
|---|---|---|---|---|

```
                                                    (z-score)
   V1            V2
     Category 1    Category 1       0.305      0.285      0.495
     Category 1    Category 2       0.136      0.156     -0.615
     Category 2    Category 1       0.025      0.046     -1.067
     Category 2    Category 2       0.534      0.513      0.447
     Bivariate Pearson Chi-Square                          1.678
     Bivariate Log-Likelihood Chi-Square                   1.899
   V1            V3
     Category 1    Category 1       0.441      0.441      0.000
     Category 1    Category 2       0.000      0.000     -0.005
     Category 2    Category 1       0.178      0.178      0.000
     Category 2    Category 2       0.381      0.381      0.000
     Bivariate Pearson Chi-Square                          0.000
     Bivariate Log-Likelihood Chi-Square                   0.000

  [ ..... ]

 V6            V7
     Category 1    Category 1       0.441      0.441      0.000
     Category 1    Category 2       0.347      0.347      0.000
     Category 2    Category 1       0.000      0.000     -0.005
     Category 2    Category 2       0.212      0.212      0.000
     Bivariate Pearson Chi-Square  0.000
     Bivariate Log-Likelihood Chi-Square                   0.000

     Overall Bivariate Pearson Chi-Square                 31.228
     Overall Bivariate Log-Likelihood Chi-Square          38.900
```

A opção TECH11 é usada em conjunto com TYPE = MIXTURE para solicitar avaliações do ajuste do modelo, como o teste da razão de verossimilhanças de *Lo-Mendell-Rubin*, que compara o modelo ajustado com um modelo com uma classe a menos do que a do modelo ajustado.

```
TECHNICAL 11 OUTPUT

   Random Starts Specifications for the k-1 Class Analysis Model
     Number of initial stage random starts                10
     Number of final stage optimizations                   2

   VUONG-LO-MENDELL-RUBIN LIKELIHOOD RATIO TEST FOR 1 (H0) VERSUS 2 CLASSES

       H0 Loglikelihood Value                         -524.465
       2 Times the Loglikelihood Difference            414.416
       Difference in the Number of Parameters            8
       Mean                                           -9.519
       Standard Deviation                             28.463
       P-Value                                         0.0000

   LO-MENDELL-RUBIN ADJUSTED LRT TEST

Value                403.835
       P-Value             0.0000
```

A opção TECH14 é usada para solicitar o teste de razão de verossimilhanças *bootstrap* paramétrico, comparando o modelo ajustado com um modelo que tenha uma classe a menos. Neste caso, Ho refere-se à hipótese de melhor ajuste para o modelo com k-1 classes, onde k denota o número de classes do modelo ajustado.

TECHNICAL 14 OUTPUT

PARAMETRIC BOOTSTRAPPED LIKELIHOOD RATIO TEST FOR 1 (H0) VERSUS 2 CLASSES

| | |
|---|---|
| H0 Loglikelihood Value | -524.465 |
| 2 Times the Loglikelihood Difference | 414.416 |
| Difference in the Number of Parameters | 8 |
| Approximate P-Value | 0.0000 |
| Successful Bootstrap Draws | 5 |

## 3. Análise de Classes Latentes no Software R

Neste tutorial utilizou-se o software R, v 3.0.3, que é um software livre e aberto a desenvolvimento metodológico estatístico. O R é composto por uma variedade de funções e pacotes/bibliotecas (*libraries*) para implementação de técnicas estatísticas, sendo um software estatístico com grande flexibilidade, o que o torna bastante popular no meio acadêmico. Além disso, possui facilidade, robustez e qualidade na produção das análises e gráficos. O software R pode ser instalado gratuitamente através do endereço eletrônico: www.r-project.org.

Selecionou-se neste tutorial 4(quatro) bibliotecas/pacotes do R que fazem a implementação de LCA. Os pacotes escolhidos foram ***poLCA*** (Linzer e Lewis, 2011), ***e1071*** (Dimitriadou, Hornik, Lisch e Weingessel, 2011), ***lcca*** (Schafer e Kang, 2013) e ***randomLCA*** (Beath, 2014). O pacote ***poLCA*** permite a análise de classes latentes com variáveis politômicas, enquanto o ***e1071*** é um pacote que só trabalha com variáveis dicotômicas e foi desenvolvido por pesquisadores da Universidade Técnica de Viena. O ***randomLCA*** permite realizar análises de classes latentes com inclusão de efeitos aleatórios, bem como os modelos de classes latentes padrões. O ***lcca*** permite o ajuste de um modelo LCA convencional com ou sem covariáveis. A sintaxe e o *output* de cada um destes pacotes para implementação da LCA básica são apresentados a seguir.

Os dados sobre carcinoma de colo de útero são reanalisados utilizando-se estas bibliotecas para ilustração do procedimento de implementação da LCA no software *R*.

### 3.1 Instalação de bibliotecas/pacotes

Nesta secção apresentamos o processo de *download* de um pacote do *software R* através da sua rede de distribuição (em inglês **CRAN**). Para ilustrar como executar este procedimento vamos utilizar o pacote ***poLCA***, que deve ser instalado após a instalação padrão do *R*. Para isto, conectado à internet, clique na opção ***Pacotes*** do menu principal do *software R* e, em seguida, clique em ***Instalar pacote(s)...***, seguindo as instruções e confirmando os *downloads* necessários (Figuras 3.1 e 3.2) (R Development Core Team, 2014).

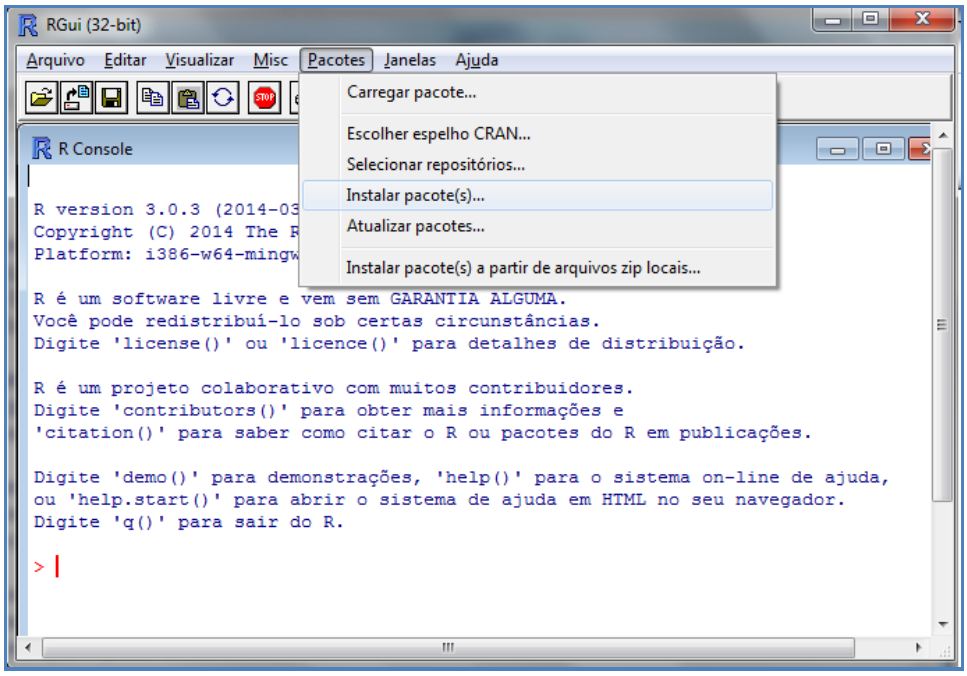


*Figura 3.1 Instalando pacotes no R*

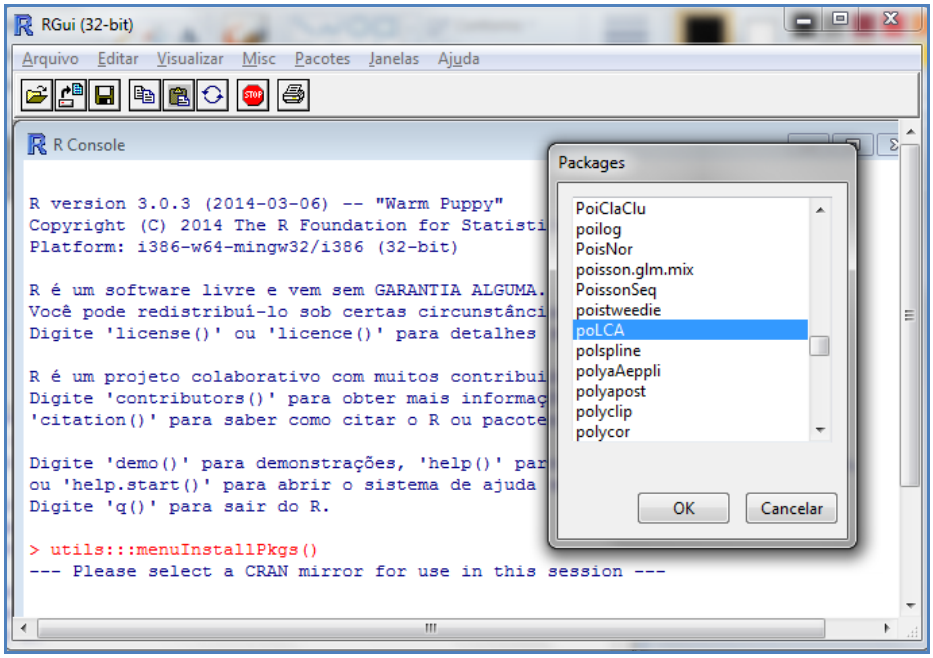


*Figura 3.2 Selecionando o pacote a ser instalado*

Entretanto nem todos os pacotes para LCA estão disponíveis para *download* no **CRAN**. Como exemplo podemos citar o pacote ***lcca***, que foi desenvolvido por um grupo de pesquisadores da Universidade Estadual da Pensilvânia. Neste caso o usuário deve se registrar no site (http://methodology.psu.edu) e realizar o *download* do pacote no link

http://methodology.psu.edu/downloads/lcca, onde se encontra o arquivo com a extensão *.zip*. Após o *download* deve-se proceder com a instalação como apresentado nas Figuras 3.3 e 3.4.

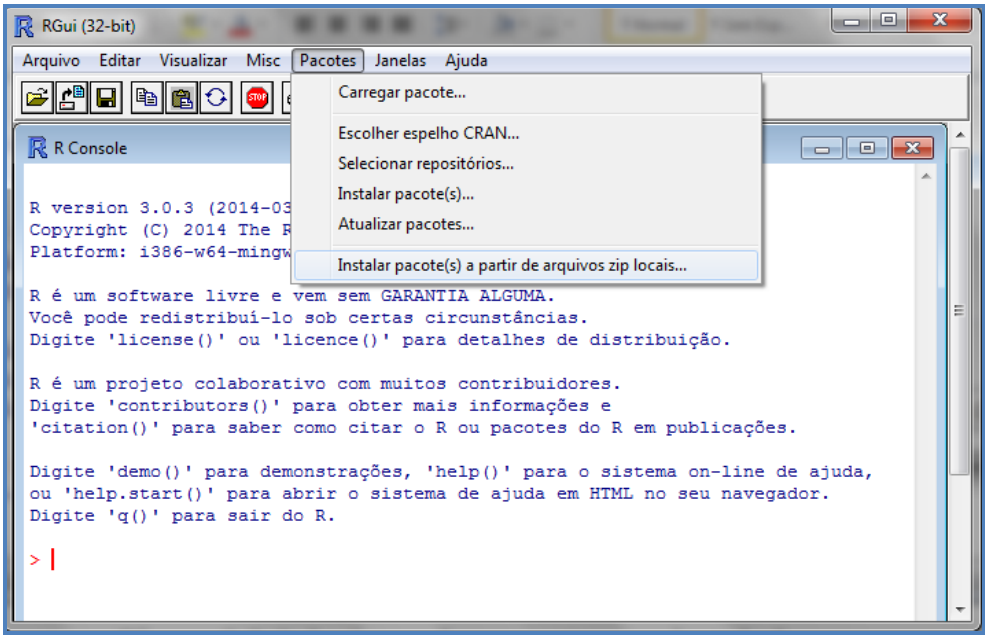


*Figura 3.3 Instalando pacotes em arquivos zip*

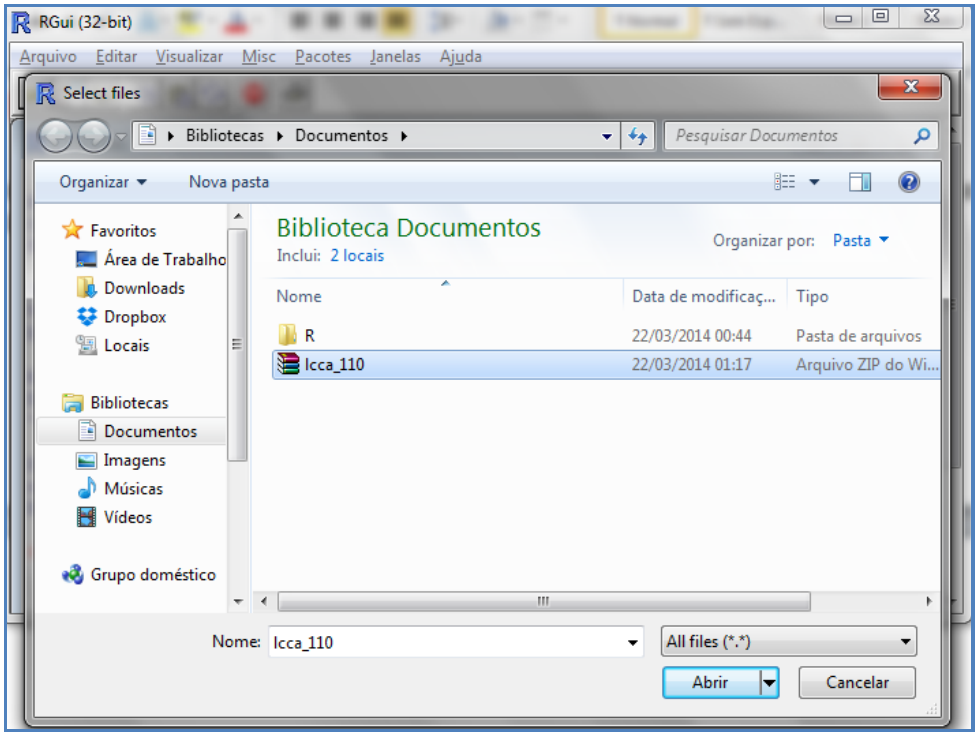


*Figura 3.4 Selecionando o arquivo para instalar o pacote lcca*

O uso do *software* R requer a digitação de sintaxe específica, que também pode ser copiada e colada na linha de comando ou no *prompt* de comando (símbolo >), onde o cursor fica posicionado. A forma geral dos comandos do R é:

**Nomedoobjeto = nome-do-comando(argumento1, argumento2, ...)**

Note que dentro dos parêntesis *( )* os argumentos ficam separados por vírgula. O sinal " = " indica a atribuição de um nome ao objeto criado pelo comando. Para ver a sintaxe completa de um comando tecle **help(nome-do-comando)**, como, por exemplo, **help(*poLCA*).**

### 3.2 Descrição das bibliotecas para LCA

As seguintes bibliotecas podem ser utilizadas para análise de classes latentes no software *R*:

**poLCA** (*Polytomous Variable Latent Class Analysis*) (Versão 1.4.1)

Essa biblioteca permite a análise de classes latentes para variáveis tanto dicotômicas quanto politômicas. A idéia central é ajustar um modelo em que qualquer relação entre as variáveis manifestas/observadas possa ser explicada por uma única variável latente categórica, que não é observada. Esta versão do pacote foi publicada no dia 10/01/2014 pelos autores Drew Linzer e Jeffrey Lewis. O link para acesso a este pacote é: http://cran.r-project.org/web/packages/poLCA/index.html.

**e1071** (Versão 1.6-3)

A ***e1071*** é uma biblioteca mais geral que apresenta suporte para diversos tipos de análises, entre elas a análise de classes latentes. Ela contém várias funções desenvolvidas por pesquisadores do Departamento de Estatística da Universidade Técnica de Viena, mas neste tutorial estamos descrevendo apenas a sintaxe referente à implementação da LCA. Esta versão do pacote foi publicada no dia 17/03/2014. O link para acesso a este pacote é: http://cran.r-project.org/web/packages/e1071/index.html.

**lcca** (Versão 1.1.0)

A biblioteca ***lcca*** permite a modelagem causal com classes latentes, bem como o ajuste de um modelo LCA convencional com ou sem covariáveis. Ela permite ainda a inclusão de variáveis politômicas. Esta versão foi publicada no ano de 2013 pelos autores Schafer e Kang. O link para acesso a este pacote é: http://methodology.psu.edu/downloads/lcca.

**randomLCA** (Versão 0.8-7)

A biblioteca ***randomLCA*** permite realizar análise de classes latentes incluindo efeitos aleatórios, que incorporam uma possível dependência entre observações, bem como

LCA padrão. A leitura dos dados nesta biblioteca é feita no formato que inclui os padrões de resposta e sua correspondente frequência. Esta versão do pacote foi publicada no dia 14/12/2013 pelo autor Ken Beath. O link para acesso a este pacote é: http://cran.r-project.org/web/packages/randomLCA/index.html.

## 3.3 Implementação da LCA

### 1º Passo: *Carregando pacotes para implementação do LCA*

Após instalar as bibliotecas no computador, deve-se carregar o pacote a ser utilizado para a análise de dados na memória do software. Este procedimento deve ser repetido toda vez que uma biblioteca for necessária para uma determinada análise de dados. O pacote pode ser carregado utilizando o comando *library(nome_do_pacote)*, como mostrado na Figura 3.5**.** Para verificar quais pacotes encontram-se ativados no R basta digitar *search()* no cursor do software.

```
> library(poLCA)
Carregando pacotes exigidos: scatterplot3d
Carregando pacotes exigidos: MASS
> # Verificando quais são os pacotes que estão carregados
> search()
 [1] ".GlobalEnv"            "package:poLCA"         "package:MASS"
 [4] "package:scatterplot3d" "package:lcca"          "package:stats"
 [7] "package:graphics"      "package:grDevices"     "package:utils"
[10] "package:datasets"      "package:methods"       "Autoloads"
[13] "package:base"
>
```

*Figura 3.5 Carregando o pacote poLCA e verificando quais estão ativos.*

O símbolo # é usado no *software R* para fazer comentários, conforme ilustrado na Figura 3.5.

### 2º Passo: *Leitura do banco de dados*

Neste tutorial estamos usando o banco de dados *carcinoma*, que já foi descrito anteriormente, e que está disponível na biblioteca ***poLCA***. A descrição da leitura do banco de dados para duas situações específicas é feita a seguir.

*Situação 1* (Quando ainda não se tem o banco de dados no computador, mas o mesmo está disponível em algum pacote do *R*):

1. *Carregando o banco de dados:*

Caso ainda não se tenha o banco de dados *carcinoma* no computador, utiliza-se a função *data( )* depois de incluir a biblioteca ***poLCA*** no ***R*** (Figura 3.6). O banco de dados vai ser denominado *carcinoma_positivo*. Vale relembrar que existe diferença entre letras maiúsculas e minúsculas no software ***R***. É importante frisar também que o software ***R***

permite a leitura de bases de dados com diferentes extensões. Para simplificar a ilustração dos métodos dispostos neste tutorial, apenas o formato ASCII é utilizado.

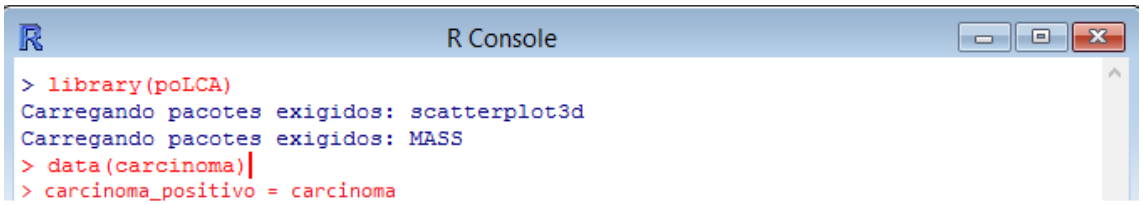


*Figura 3.6 Carregando o banco de dados carcinoma.*

*2. Salvando o banco de dados no computador:*

Para salvar o banco de dados *carcinoma* no computador utiliza-se a função *write.table( )* (Figura 3.7), conforme definido a seguir:

**write.table(nome do objeto, "nome do arquivo.txt", sep = "\t")**

onde:

**Nome do objeto** – Nome do objeto a ser salvo no arquivo.

**Nome do arquivo.txt** – Nome do arquivo com extensão *.txt* a ser criado contendo o banco de dados, salvo em diretório especificado pelo analista de dados.

**sep = "\t"** – indica qual é o separador de colunas para a base de dados. Neste caso particular informa que as colunas estão separadas por tabulação. Existem também outras formas de separar os dados, sendo as principais por espaço em branco ou por vírgula, respectivamente definidas por *sep = " "* e *sep = ","*.

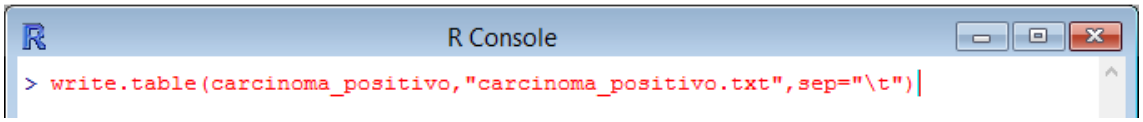


*Figura 3.7 Salvando o banco de dados carcinoma_positivo.*

*Situação 2* (Quando o banco de dados já está disponível no computador):

Para a leitura do banco de dados em formato ASCII (arquivos de texto) usa-se a função *read.table( )*, conforme mostrado a seguir e na Figura 3.8:

**Nome_dados=read.table("endereço completo do arquivo de dados", sep = "\t", h=T)**

onde:

**Nome_dados** – Nome do objeto que vai denotar o banco de dados.

**"endereço..."** – Diretório completo onde se encontra seu arquivo. Por exemplo: ("c:/arquivo.txt").

**h= T** – Indica que a primeira linha da base de dados é o cabeçalho, contendo o nome das variáveis de cada coluna. Caso a base de dados não tenha cabeçalho muda-se T(*true*) por F(*false*).

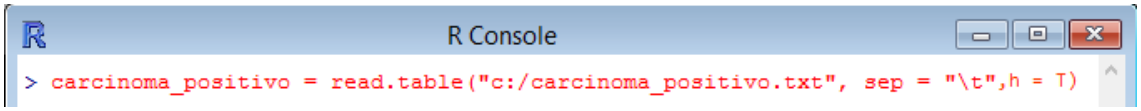


*Figura 3.8 Usando a função read.table*

**3º Passo** – ***Implementando LCA em distintas bibliotecas do R***

***(A) Utilizando a biblioteca poLCA.***

Após ativar o pacote ***poLCA***, vamos implementar um modelo com 2 classes latentes usando a função ***poLCA*** desta biblioteca. O comando geral para a construção de um modelo de classes latentes nesta biblioteca é definido como:

***poLCA(formula, data, nclass = 2, maxiter = 1000, graphs = FALSE, tol = 1e-10, na.rm = TRUE, probs.start = NULL, nrep = 1, verbose = TRUE, calc.se = TRUE)***

As especificações para cada componente do comando acima são listadas a seguir:

***formula:*** A expressão define a relação entre os indicadores da LCA e covariáveis na forma: indicadores ~ covariáveis. Os modelos de classes latentes têm mais de uma variável manifesta ou indicadores, de modo que são definidas por cbind (dv1, dv2, dv3,...), onde dv# refere-se ao nome dos indicadores para a variável latente. Para os modelos sem covariáveis e com 3 indicadores, por exemplo, a fórmula é dada por cbind (dv1, dv2, dv3) ~ 1. Para os modelos com covariáveis, deve-se substituir o ~ 1 pelas covariáveis iv1, iv2, iv3, como, por exemplo, cbind (dv1, dv2, dv3) ~ iv1 + iv2 + iv3.
***data:*** Base de dados contendo variáveis incluídas na análise. Variáveis manifestas devem conter apenas valores inteiros, e devem ser codificadas com valores consecutivos a partir do número 1 até o número máximo de categorias para cada variável. Todos os valores faltantes devem ser inseridos como NA.

***nclass:*** Número de classes latentes definidas para o modelo. Definir *nclass = 1* resulta na estimação do modelo de independência loglinear. O *default* é 2.
***maxiter:*** Número máximo de iterações permitido no processo de estimação, que pode ser alterado pelo usuário.
***graphs:*** Opção para que o pacote possa exibir graficamente as estimativas dos parâmetros após a conclusão do algoritmo de estimação. O *default* é FALSE.
***tol:*** Um valor de tolerância para julgar quando a convergência foi alcançada. Quando a mudança de uma iteração no logaritmo da verossimilhança estimada é inferior a *tol*, o algoritmo de estimação conclui o processo de iteração. Esta opção pode ser alterada pelo usuário.
***na.rm:*** Maneira com que a biblioteca lida em situações com valores faltantes nas variáveis indicadoras. Se opção for TRUE, estas observações são removidas antes da estimação do modelo. Se opção for FALSE, as observações com valores faltantes são mantidas nas análises. As observações com valores faltantes nas covariáveis são sempre removidas. O *default* é TRUE.
***probs.start:*** Uma lista de probabilidades de resposta condicionais a serem utilizadas como valores iniciais para o algoritmo de estimação. O *default* é NULL, produzindo valores iniciais aleatórios.
***nRep:*** Número de vezes que se deve estimar o modelo usando diferentes valores de *probs.start*. O *default* é 1. Definir *nRep* > 1 automatiza a busca pela máxima verossimilhança global ao invés de apenas um máximo local. ***poLCA*** retorna as estimativas dos parâmetros correspondentes ao modelo com a maior verossimilhança.
***verbose:*** Indica se as saídas/*outputs* dos resultados do modelo devem ser apresentadas. Se a opção for FALSE, nenhuma saída é produzida. O *default* é TRUE.
***calc.se:***Indica se os erros-padrão das probabilidades de resposta condicionais devem ser calculados. O padrão é TRUE; só pode ser definido como FALSE para estimação de um modelo básico sem covariáveis especificadas na fórmula.

Quando se utiliza o termo *default* está-se referindo às opções padrão do argumento. Dos argumentos descritos anteriormente para o pacote ***poLCA***, os únicos que devem ser obrigatoriamente informados são: *formula* e *data*. Se os demais não forem referidos, o *software* utiliza as correspondentes opções *default*. A biblioteca ***poLCA*** usa os algoritmos EM e Newton-Raphson para maximização da função de verossimilhança do modelo LCA.

A maneira geral utilizada para definir um modelo LCA para análise dos dados de carcinoma na biblioteca ***poLCA*** é apresentada abaixo:

```
f=cbind(A,B,C,D,E,F,G)~1
analiseLCA=poLCA(f,carcinoma_positivo)
```

Ou equivalentemente pode-se usar diretamente:

```
analiseLCA=poLCA(cbind(A,B,C,D,E,F,G)~1,carcinoma_positivo)
```

Para os demais argumentos que não foram incluídos nesta sintaxe se considerou as opções *default.* A implementação e resultados destas análises são apresentadas nas Figuras 3.9 e 3.10.

```
R Console

> library(poLCA)
> data(carcinoma)
> carcinoma_positivo = carcinoma
> f = cbind(A,B,C,D,E,F,G)~1
> analiseLCA = poLCA(f, carcinoma_positivo, nclass = 2)
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$A
           Pr(1)  Pr(2)
class 1:  0.8835 0.1165
class 2:  0.0000 1.0000

$B
           Pr(1)  Pr(2)
class 1:  0.6456 0.3544
class 2:  0.0169 0.9831

$C
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  0.2391 0.7609

$D
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  0.4589 0.5411

$E
           Pr(1)  Pr(2)
class 1:  0.7771 0.2229
class 2:  0.0214 0.9786

$F
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  0.5773 0.4227

$G
           Pr(1)  Pr(2)
class 1:  0.8835 0.1165
class 2:  0.0000 1.0000

Estimated class population shares
 0.4988 0.5012

Predicted class memberships (by modal posterior prob.)
 0.5 0.5
```

*Figura 3.9 Probabilidades condicionais e não condicionais estimadas no pacote poLCA.*

O *output* contendo as estatísticas de bondade de ajuste para este modelo é apresentado na Figura 3.10.

```
============================================================
Fit for 2 latent classes:
============================================================
number of observations: 118
number of estimated parameters: 15
residual degrees of freedom: 103
maximum log-likelihood: -317.2568

AIC(2): 664.5137
BIC(2): 706.074
G^2(2): 62.36543 (Likelihood ratio/deviance statistic)
X^2(2): 92.64814 (Chi-square goodness of fit)
```

*Figura 3.10 Estatísticas de bondade do ajuste apresentadas pelo pacote poLCA.*

O objeto *analiseLCA* definido na biblioteca ***poLCA*** pode fornecer informações em formato de uma tabela contendo os padrões de respostas e as correspondentes frequências observadas e esperadas, podendo ser obtida com o comando *analiseLCA$predcell* (Figura 3.11). Os resultados referentes à frequência observada deste objeto podem ser usados como banco de dados para a biblioteca ***randomLCA*** e para o software ***WinLTA***, pois ambos aceitam esse padrão de formato para o banco de dados. O procedimento para extração dos padrões de respostas e correspondentes frequências é apresentado na Figura 3.20.

R Console

```
> analiseLCA$predcell
   A B C D E F G observed expected
1  1 1 1 1 1 1 1       34   23.049
2  1 1 1 1 2 1 1        2    6.612
3  1 2 1 1 1 1 1        6   12.651
4  1 2 1 1 1 1 2        1    1.668
5  1 2 1 1 2 1 1        4    3.629
6  1 2 1 1 2 1 2        5    0.479
7  2 1 1 1 1 1 1        2    3.039
8  2 1 2 1 2 1 2        1    0.197
9  2 2 1 1 1 1 1        2    1.668
10 2 2 1 1 1 1 2        1    0.299
11 2 2 1 1 2 1 1        2    0.479
12 2 2 1 1 2 1 2        7    3.668
13 2 2 1 1 2 2 2        1    2.640
14 2 2 1 2 1 1 2        1    0.093
15 2 2 1 2 2 1 2        2    4.250
16 2 2 1 2 2 2 2        3    3.112
17 2 2 2 1 2 1 2       13   11.470
18 2 2 2 1 2 2 2        5    8.399
19 2 2 2 2 2 1 2       10   13.523
20 2 2 2 2 2 2 2       16    9.902
```

*Figura 3.11 Tabela com os padrões de resposta e as frequências observadas e esperadas obtidos no pacote poLCA*

No pacote ***poLCA*** pode-se ainda visualizar as probabilidades não condicionais e condicionais graficamente. Para isto, usa-se o argumento *graph* incluído na função *poLCA* da seguinte forma:

```
> analiseLCA = poLCA(f, carcinoma_positivo, nclass = 2, graph = T)
```

Verifique ainda que se incluiu na sintaxe anterior a opção *nclass = 2*, que havia sido omitida anteriormente porque o número de classes *default* da função *poLCA* é 2. No entanto, se for de interesse alterar o número de classes, esta opção precisa ser utilizada. Na Figura 3.12 tem-se o gráfico obtido via função *poLCA*, que apresenta as estimativas das probabilidades não condicionais, e representa as probabilidades condicionais através de colunas associadas a cada uma das classes e correspondentes variáveis manifestas ou indicadoras.

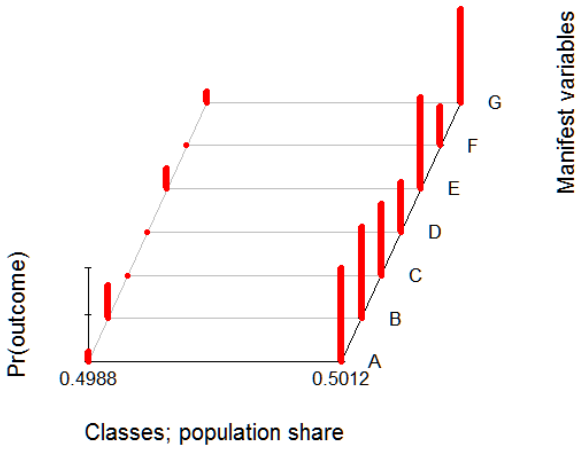


*Figura 3.12 Grafico gerado pela função poLCA.*

***(B) Utilizando a biblioteca e1071.***

Para utilizar a biblioteca ***e1071***, os dados devem ser binários e codificados como 0 e 1. Uma vez que a biblioteca e o banco de dados "carcinoma_positivo" já estão disponíveis no computador, ver-se-á na Figura 3.13 uma maneira de recodificar os dados originais (códigos 1 e 2) para o novo objeto - carcinoma_binário (códigos 0 e 1) - a ser analisado na biblioteca ***e1071***.

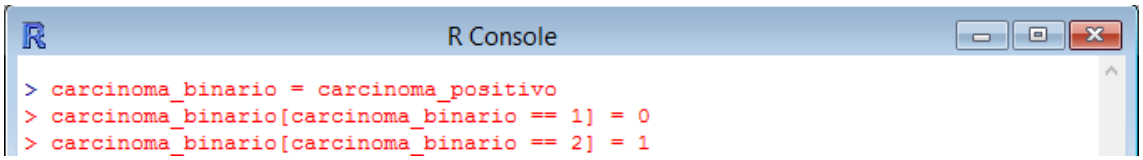


*Figura 3.13 Recodificando a base com dados de carcinoma.*

Para a implementação de um modelo de classes latentes na biblioteca ***e1071*** usa-se o seguinte comando:

**Nome_do_objeto = lca(padrões de resposta, número de classes latentes, número de interações, matchdata=TRUE)**

Com o argumento *matchdata= TRUE*, pode-se identificar a classe latente à qual cada indivíduo pertence. Esta é a opção *default* do pacote. A biblioteca ***e1071*** permite que o ajuste do modelo seja feito utilizando-se os padrões de respostas ou os dados originais, da maneira convencional. Ambos os casos são ilustrados a seguir.

Na Figura 3.14 apresenta-se a utilização do comando *lca* da biblioteca ***e1071*** do R considerando dados em formato de padrões de resposta. Na biblioteca ***e1071*** as probabilidades de resposta ao item (PRI's) relacionadas a somente uma das categorias de resposta são fornecidas - denominadas *Itemprobabilities* no *output* - uma vez que as outras são complementares (Figura 3.14). As PRI's apresentadas pelo *output* são relacionadas às categorias de respostas iguais a 1 e podem ser obtidas pelo comando analise*LCA$p*, sendo este o objeto definido na Figura 3.14.

```
> library(e1071)
> contador_padrao=countpattern(carcinoma_binario)
> analiseLCA = lca(contador_padrao, 2, niter = 10, matchdata = T)
> print(analiseLCA)
LCA-Result
----------

Datapoints: 118
Classes:    2
Probability of classes
[1] 0.522 0.478
Itemprobabilities
     1    2    3    4    5    6    7
1 0.04 0.02 0.27 0.48 0.02 0.59 0.00
2 0.88 0.67 1.00 1.00 0.81 1.00 0.92
```

*Figura 3.14 Output da biblioteca e1071 com dados de carcinoma.*

Nesta biblioteca se a leitura dos dados for baseada nos padrões de resposta, estes e suas correspondentes frequências podem ser obtidos através do comando *countpattern* (Figura 3.15). Verifique que são apresentados todos os padrões de resposta possíveis de serem observados com 7 variáveis indicadoras ($2^7$=128), mas que não foram necessariamente observados no estudo. Nos nossos dados foram observados apenas 20 padrões.

```
> contador_padrao=countpattern(carcinoma_binario)
> contador_padrao
0000000 0000001 0000010 0000011 0000100 0000101 0000110 0000111 0001000 0001001
     16       0      10       0       0       0       0       0       5       0
0001010 0001011 0001100 0001101 0001110 0001111 0010000 0010001 0010010 0010011
     13       0       0       0       0       0       3       0       2       0
0010100 0010101 0010110 0010111 0011000 0011001 0011010 0011011 0011100 0011101
      0       0       1       0       1       0       7       2       0       0
0011110 0011111 0100000 0100001 0100010 0100011 0100100 0100101 0100110 0100111
      1       2       0       0       0       0       0       0       0       0
0101000 0101001 0101010 0101011 0101100 0101101 0101110 0101111 0110000 0110001
      0       0       1       0       0       0       0       0       0       0
0110010 0110011 0110100 0110101 0110110 0110111 0111000 0111001 0111010 0111011
      0       0       0       0       0       0       0       0       0       0
0111100 0111101 0111110 0111111 1000000 1000001 1000010 1000011 1000100 1000101
      0       0       0       2       0       0       0       0       0       0
1000110 1000111 1001000 1001001 1001010 1001011 1001100 1001101 1001110 1001111
      0       0       0       0       0       0       0       0       0       0
1010000 1010001 1010010 1010011 1010100 1010101 1010110 1010111 1011000 1011001
      0       0       0       0       0       0       0       0       0       0
1011010 1011011 1011100 1011101 1011110 1011111 1100000 1100001 1100010 1100011
      5       4       0       0       1       6       0       0       0       0
1100100 1100101 1100110 1100111 1101000 1101001 1101010 1101011 1101100 1101101
      0       0       0       0       0       0       0       0       0       0
1101110 1101111 1110000 1110001 1110010 1110011 1110100 1110101 1110110 1110111
      0       0       0       0       0       0       0       0       0       0
1111000 1111001 1111010 1111011 1111100 1111101 1111110 1111111
      0       0       0       2       0       0       0      34
```

*Figura 3.15 Padrões de resposta obtidos pela função countpattern da biblioteca e1071.*

A classe latente à qual pertence cada padrão de respostas também pode ser obtida através desta biblioteca. Para isso utiliza-se o comando *analiseLCA$matching* (*F*igura 3.16).

```
> analiseLCA$matching
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1
[112] 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2
```

*Figura 3.16 Identificando a que classe pertence cada padrão de resposta na biblioteca e1071*

Para obter maiores detalhes sobre o ajuste do modelo de classes latentes pode-se digitar o comando *summary(nome_objeto)*, conforme apresentado na Figura 3.17. O Critério de Informação de Akaike (AIC) não é fornecido por esta biblioteca como opção para avaliar a bondade do ajuste do modelo.

```
> summary(analiseLCA)
LCA-Result
----------

Datapoints: 118
Classes:    2

Goodness of fit statistics:

Number of parameters, estimated model: 15
Number of parameters, saturated model: 127
Log-Likelihood, estimated model:       -317.257
Log-Likelihood, saturated model:       -286.0741

Information Criteria:

BIC, estimated model: 706.0742
BIC, saturated model: 1178.025

TestStatistics:

Likelihood ratio:   62.36569   p-val: 0.9999603
Pearson Chi^2:      92.64471   p-val: 0.9084114
Degress of freedom: 112
```

*Figura 3.17 Estatísticas de bondade do ajuste usando LCA com a biblioteca e1071.*

A segunda forma de leitura das bases de dados na biblioteca ***e1071*** é através da definição de uma matriz. Para isso pode-se utilizar o comando:

***dados_matriz = as.matrix(banco_de_dados)***

E para ajuste da LCA a sintaxe é dada por:

**Nome_do_objeto = lca(dados_matriz, número de classes latentes, número de interações)**

Neste caso não se faz necessária a inclusão do argumento *matchdata* na função *lca*, conforme apresentado na Figura 3.14. Com este formato de dados, pode-se usar diretamente o comando analiseLCA$matching para obter a identificação das classes latentes às quais pertencem todos os possíveis padrões de resposta (Figura 3.18). Estes padrões de resposta referem-se a aqueles listados na Figura 3.15.

```
> dados_matriz = as.matrix(carcinoma_binario)
> analiseLCA = lca(dados_matriz, 2 niter = 10)
> analiseLCA$matching
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1
[112] 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2
```

*Figura 3.18 Leitura de dados em formato de matriz e identificação da classe latente associada à cada padrão de respostas.*

***(C) Utilizando a biblioteca lcca***

Para utilizar a biblioteca ***lcca***, as variáveis devem estar codificadas com números estritamente positivos (não pode conter o zero). Por conta disto, utilizou-se novamente o banco de dados *carcinoma_positivo*. O ajuste do modelo LCA nesta biblioteca é parecido ao feito na biblioteca ***poLCA***, sendo definido por:

***library(lcca)***
***f=cbind(A,B,C,D,E,F,G)~1***
***analiseLCA=lca(f, nome do banco de dados)***

Ou equivalentemente pode-se usar:

***analiseLCA=lca(cbind(A,B,C,D,E,F,G)~1,carcinoma_positivo)***

Os resultados para a análise dos dados sobre carcinoma são apresentados na Figura 3.19. Verifique na Figura 3.19 que foi adicionada a opção *se.method=NONE* para excluir os erros-padrão uma vez que os mesmos são desnecessários para as análises mais básicas usando LCA. Caso este argumento não seja especificado, a opção *se.method="STANDARD"* é utilizada pela biblioteca.

```
> f <- cbind(A,B,C,D,E,F,G)~1
> analiseLCA = lca(f, data = carcinoma_positivo, se.method = "NONE")
> summary(anliseLCA)

        Summary of Latent-Class Analysis

================================================
Fit statistics
================================================

The EM algorithm CONVERGED in: 26 iterations

Standard errors computed successfully.
Standard-error method:   NONE

Number of free parameters estimated:     15.0000000
Loglikelihood:                         -317.2568730
Loglikelihood + penalty:               -317.2568730
-2 * Loglikelihood:                     634.5137461
AIC (smaller is better):                664.5137461
BIC (smaller is better):                706.0740155

=======================================
Parameter estimates
=======================================

Class prevalences (gammas):
Class:         1         2
        0.4988    0.5012

Item-response probabilities (rhos):
   Response category 1
Class:         1         2
   A      0.8835    0.0000
   B      0.6456    0.0169
   C      1.0000    0.2391
   D      1.0000    0.4589
   E      0.7771    0.0214
   F      1.0000    0.5773
   G      0.8835    0.0000

   Response category 2
Class:         1         2
   A      0.1165    1.0000
   B      0.3544    0.9831
   C      0.0000    0.7609
   D      0.0000    0.5411
   E      0.2229    0.9786
   F      0.0000    0.4227
   G      0.1165    1.0000
```

*Figura 3.19 Resultados do ajuste do modelo LCA com duas classes latentes utilizando a biblioteca lcca.*

***(D) Utilizando a biblioteca randomLCA***

A execução do modelo LCA é feita exclusivamente através dos padrões de respostas e das frequências com que eles aparecem na biblioteca ***randomLCA***. O comando geral para o ajuste é dado como segue:

**randomLCA(padrões de respostas, frequência dos padrões)**

Para obter esses padrões e suas respectivas frequências pode-se utilizar outras bibliotecas, como a ***poLCA***. Como vimos anteriormente (Figura 3.11), após a execução da LCA utilizando a biblioteca **poLCA**, o comando *analiseLCA$predcell* fornece os padrões de resposta e suas respectivas frequências observadas e esperadas. Pode-se então salvar os padrões de resposta no objeto *padr* e as frequências observadas no objeto *freq*. Este processo é apresentado na Figura 3.20.

```
> library(poLCA)
> f = cbind(A, B, C, D, E, F, G)~1
> analiseLCA = poLCA(f, carcinoma_positivo, nclass = 2)
> freq = analiseLCA$predcell[,8]
> padr = analiseLCA$predcell[,1:7]
> padr[padr == 1] = 0
> padr[padr == 2] = 1
```

*Figura 3.20 Extraindo padrões de resposta e correspondentes frequências observadas usando a biblioteca poLCA.*

Na Figura 3.21 são apresentados os objetos *padr* e *freq* para os nossos dados, conforme definidos na Figura 3.20. O objeto *padr* é uma matriz com 7 colunas (número de variáveis indicadoras no nosso exemplo) e 20 linhas, que se refere ao número de padrões de frequências observados. O objeto *freq* é um vetor com dimensão correspondente ao número de padrões observados.

```
> padr
   A B C D E F G
1  0 0 0 0 0 0 0
2  0 0 0 0 1 0 0
3  0 1 0 0 0 0 0
4  0 1 0 0 0 0 1
5  0 1 0 0 1 0 0
6  0 1 0 0 1 0 1
7  1 0 0 0 0 0 0
8  1 0 1 0 1 0 1
9  1 1 0 0 0 0 0
10 1 1 0 0 0 0 1
11 1 1 0 0 1 0 0
12 1 1 0 0 1 0 1
13 1 1 0 0 1 1 1
14 1 1 0 1 0 0 1
15 1 1 0 1 1 0 1
16 1 1 0 1 1 1 1
17 1 1 1 0 1 0 1
18 1 1 1 0 1 1 1
19 1 1 1 1 1 0 1
20 1 1 1 1 1 1 1
```

```
> freq
 [1] 34  2  6  1  4  5  2  1  2  1  2  7  1  1  2  3 13  5 10 16
```

*Figura 3.21 Objetos padr e freq obtidos com uso da biblioteca poLCA.*

O *output* obtido através do *summary* do objeto gerado com o comando *randomLCA* é apresentado na Figura 3.22. Verifique que a biblioteca *randomLCA* fornece as probabilidades condicionais associadas à resposta 2 (sim) originalmente no banco *carcinoma_positivo*, enquanto a biblioteca ***e1071*** fornece estas probabilidades para a resposta para a categoria 1 (não), e as bibliotecas ***poLCA*** e ***lcca*** fornecem a informação para ambas categorias de resposta. Resultados mais completos são obtidos através do comando *print(analiseLCA)*, que não são apresentados aqui devido à sua extensão.

```
> library(randomLCA)
> analiseLCA = randomLCA(padr, freq)
> summary(analiseLCA)
  Classes      AIC      BIC    logLik penlogLik
        2 664.5147 706.0749 -317.2573 -317.2631
Class probabilities
Class  1 Class  2
  0.4988   0.5012
Outcome probabilities
              A      B      C      D      E      F      G
Class  1 0.1165 0.3544 0.0000 0.0000 0.2229 0.0000 0.1165
Class  2 1.0000 0.9831 0.7609 0.5411 0.9786 0.4227 1.0000
```

*Figura 3.22 Resultados da biblioteca randomLCA para os dados de carcinoma.*

Outro formato de dados que é reconhecido pela biblioteca ***randomLCA*** é quando temos os padrões de respostas definidos por colunas que indicam os resultados para cada variável indicadora e existe uma coluna adicional que contém as correspondentes frequências observadas (Figura 3.23). Verifique que esta é a mesma informação que havia sido apresentada na Figura 3.11, mas seu formato é de banco de dados e não como objetos em formato de matrizes e vetores. Diferentemente da Figura 3.11 as categorias precisam estar codificadas como 0 e 1 para uso da biblioteca ***randomLCA***.

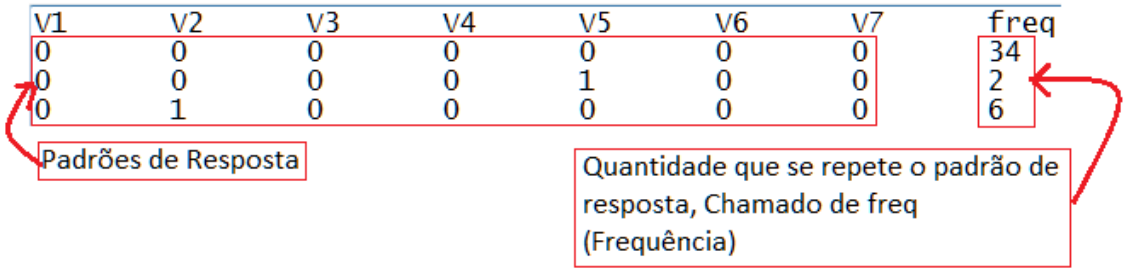


*Figura 3.23 As primeiras linhas do banco de dados agregado*

Neste caso, o comando para ajuste para esta estrutura de dados seria:

**summary(*randomLCA(dados[,1:7], dados[,8])*),**

onde, neste exemplo, *dados[,1:7]* indicam as colunas que contém os padrões de resposta e *dados[,8]* indica sua correspondente frequência.

A biblioteca ***randomLCA*** permite o ajuste de modelos de classes latentes convencionais bem como LCA com efeitos aleatórios, que não é discutido neste tutorial. As variáveis devem ser binárias, não sendo possível a inclusão de variáveis politômicas no modelo. Nesta biblioteca podem ainda ser obtidas informações mais detalhadas sobre o ajuste do modelo, como o gradiente hessiano e a estimação de parâmetros livres.

### 3.4 Novas funções propostas

É importante informar que as funções descritas a seguir requerem o uso dos pacotes **poLCA**, **randomLCA** e **psych** do software R, que precisam ter sido instalados no computador antes do seu uso.

#### 3.4.1 Cálculo da Entropia

Como a entropia não é fornecida pelas versões dos pacotes do *R* descritas neste tutorial, definiu-se uma função para estimação desta medida de bondade de ajuste da LCA. A definição matemática desta função pode ser encontrada na equação (3.6) na pág. 75 de Collins e Lanza (2010). Esta função é descrita a seguir.

**Sintaxe associada à função:**

```
entropy(formula, data, n_class, n_decimal)
```

**Argumentos:**

**formula**: Expressão que se refere à relação entre os indicadores da LCA e covariáveis na forma: indicadores ~ covariáveis.

**data**: Base de dados contendo variáveis indicadoras usadas na análise. Como a biblioteca ***poLCA*** é utilizada nesta função, então as variáveis devem estar codificadas como 1 e 2.

**n_class**: O número de classes do modelo LCA.

**n_decimal:** O número de casas decimais a serem utilizadas. Não é um argumento obrigatório. Quando este argumento não for utilizado, o *default* é: *n_decimal = 4.*

A função ***entropy*** é listada no Apêndice 1 deste tutorial.

A execução da função *entropy* utilizando o banco de dados *carcinoma_positivo* pode ser feita, passo-a-passo, usando:

```
f = cbind(A,B,C,D,E,F,G)~1
n_class = 2
data = carcinoma_positivo
E = entropy(f, data, n_class)
print(E)
```

Equivalentemente, esta função pode ser executada através de:

```
E = entropy(cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 4)
```

O resultado da função *entropy* para os dados sobre carcinoma é listado na Figura 3.24, que retorna o valor da entropia. De maneira similar, o valor da entropia pode ser obtido ao se usar o comando *print( )* no objeto que contém o resultado desta função.

```
> E = entropy(cbind(A,B,C,D,E,F,G)~1,carcinoma_positivo,2)
Entropia =  0.9855
> print(E)
[1] 0.9855
```

*Figura 3.24 Output da função entropy().*

### 3.4.2 Cálculo das probabilidades a posteriori de pertencimento às classes latentes condicionais ao padrão de resposta

Estas probabilidades foram definidas matematicamente conforme equação (3.5) na pág. 69 de Collins e Lanza (2010). Esta função foi denominada ***prob.class_padr*** e é descrita a seguir.

**Sintaxe associada à função:**

```
prob.class_padr(formula, data, n_class, n_var, n_decimal)
```

**Argumentos:**

**formula**: Expressão que se refere à relação entre os indicadores da LCA e covariáveis na forma: indicadores ~ covariáveis.

**data**: Base de dados contendo variáveis indicadoras usadas na análise. Como a biblioteca ***poLCA*** é utilizada nesta função, então as variáveis devem estar codificadas como 1 e 2.

**n_class**: O número de classes do modelo LCA.

**n_var**: O número de variáveis indicadoras.

**n_decimal:** O número de casas decimais a serem utilizadas. Não é um argumento obrigatório. Quando o argumento não for utilizado, o *default* é*: n_decimal = 4.*

A função ***prob.class_padr*** é listada no Apêndice 2 deste tutorial.

A execução da função ***prob.class_padr*** utilizando o banco de dados *carcinoma_positivo* pode ser feita, passo-a-passo, usando:

```
f = cbind(A,B,C,D,E,F,G)~1
n_class = 2
n_var = 7
data = carcinoma_positivo
prob.class = prob.class_padr(f, data, n_class, n_var)
print(prob.class)
```

Equivalentemente, esta função pode ser executada através de:

```
prob.class = prob.class_padr (cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 7)
print(prob.class)
```

O resultado da função ***prob.class_padr*** para os dados de carcinoma é listado na Figura 3.25, que retorna as probabilidades não condicionais (*Gamma Values*) e condicionais (*Rho Values*) para permitir que o analista possa identificar quais classes estão sendo descritas. Para obtenção das probabilidades *a posteriori* de pertencimento às classes latentes condicionais aos padrões de resposta é necessário usar o comando *print()* do objeto que contém o resultado desta função. As colunas *Prob.Class1* e *Prob.Class2* contêm as probabilidades correspondentes às duas classes do modelo LCA estimadas para os dados de carcinoma.

```
> prob.class = prob.class_padr(cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 7)

   Gamma Values
Class1 Class2
0.4988 0.5012

   Rho Values
           V1     V2     V3     V4     V5     V6     V7
Class1 0.1165 0.3544 0.0000 0.0000 0.2229 0.0000 0.1165
Class2 1.0000 0.9831 0.7609 0.5411 0.9786 0.4227 1.0000
> print(prob.class)
   V1 V2 V3 V4 V5 V6 V7 Prob.Class1 Prob.Class2
1   0  0  0  0  0  0  0      0.9999      0.0001
2   0  0  0  0  1  0  0      0.9907      0.0093
3   0  1  0  0  0  0  0      0.9938      0.0062
4   0  1  0  0  0  0  1      0.9549      0.0451
5   0  1  0  0  1  0  0      0.5017      0.4983
6   0  1  0  0  1  0  1      0.1172      0.8828
7   1  0  0  0  0  0  0      0.9996      0.0004
8   1  0  1  0  1  0  1      0.3683      0.6317
9   1  1  0  0  0  0  0      0.9549      0.0451
10  1  1  0  0  0  0  1      0.7362      0.2638
11  1  1  0  0  1  0  0      0.1172      0.8828
12  1  1  0  0  1  0  1      0.0172      0.9828
13  1  1  0  0  1  1  1      0.0234      0.9766
14  1  1  0  1  0  0  1      0.7030      0.2970
15  1  1  0  1  1  0  1      0.0146      0.9854
16  1  1  0  1  1  1  1      0.0199      0.9801
17  1  1  1  0  1  0  1      0.0055      0.9945
18  1  1  1  0  1  1  1      0.0075      0.9925
19  1  1  1  1  1  0  1      0.0046      0.9954
20  1  1  1  1  1  1  1      0.0063      0.9937
```

*Figura 3.25 Output da função prob.class_padr().*

### 3.4.3 Cálculo das probabilidades de se observar cada padrão de resposta (indepentemente da classe latente)

Estas probabilidades foram definidas matematicamente conforme equação (2.3) na pág 41 de Collins e Lanza (2010). Esta função foi denominada ***prob.padr*** e é descrita a seguir.

**Sintaxe associada à função:**

**prob.padr(formula, data, n_class, n_var, n_decimal)**

**Argumentos:**

**formula**: Expressão que se refere à relação entre os indicadores da LCA e covariáveis na forma: indicadores ~ covariáveis.

**data**: Base de dados contendo variáveis indicadoras usadas na análise. Como a biblioteca ***poLCA*** é utilizada nesta função, então as variáveis devem estar codificadas como 1 e 2.

**n_class**: O número de classes do modelo LCA.

**n_var**: O número de variáveis indicadoras.

**n_decimal:** O número de casas decimais a serem utilizadas. Não é um argumento obrigatório. Quando o argumento não for utilizado, o *default* é: *n_decimal = 4*.

A função ***prob.padr*** é listada no Apêndice 3 deste tutorial.

A execução da função ***prob. padr*** utilizando o banco de dados *carcinoma_positivo* pode ser feita, passo-a-passo, usando:

```
f = cbind(A,B,C,D,E,F,G)~1
n_class = 2
n_var = 7
data = carcinoma_positivo
prob_padr = prob.padr(f, data, n_class, n_var)
print(prob_padr)
```

Equivalentemente, esta função pode ser executada através de:

```
prob.padr = prob.padr (cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 7)
print(prob.padr)
```

O resultado da função ***prob.padr*** para os dados de carcinoma é listado na Figura3.26, que similarmente à função anterior retorna as probabilidades não condicionais (*Gamma Values*) e condicionais (*Rho Values*). Usando o comando *print( )* do objeto que contém o resultado desta função obtém-se as probabilidades estimadas (marginalmente) de cada padrão de resposta (*Prob.Padr*).

```
> prob_padr = prob.padr(cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 7)

   Gamma Values
Class1 Class2
0.5012 0.4988

   Rho Values
          V1     V2     V3     V4     V5     V6     V7
Class1 1.0000 0.9831 0.7609 0.5411 0.9786 0.4227 1.0000
Class2 0.1165 0.3544 0.0000 0.0000 0.2229 0.0000 0.1165
> print(prob_padr)
   V1 V2 V3 V4 V5 V6 V7 Prob.Padr
1   0  0  0  0  0  0  0    0.1953
2   0  0  0  0  1  0  0    0.0566
3   0  1  0  0  0  0  0    0.1079
4   0  1  0  0  0  0  1    0.0148
5   0  1  0  0  1  0  0    0.0613
6   0  1  0  0  1  0  1    0.0346
7   1  0  0  0  0  0  0    0.0258
8   1  0  1  0  1  0  1    0.0026
9   1  1  0  0  0  0  0    0.0148
10  1  1  0  0  0  0  1    0.0025
11  1  1  0  0  1  0  0    0.0346
12  1  1  0  0  1  0  1    0.0311
13  1  1  0  0  1  1  1    0.0229
14  1  1  0  1  0  0  1    0.0027
15  1  1  0  1  1  0  1    0.0365
16  1  1  0  1  1  1  1    0.0269
17  1  1  1  0  1  0  1    0.0977
18  1  1  1  0  1  1  1    0.0717
19  1  1  1  1  1  0  1    0.1151
20  1  1  1  1  1  1  1    0.0845
```

*Figura 3.26 Output da função prob.padr().*

### 3.4.4 Cálculo das probabilidades de se observar padrões de resposta condicionais às classes latentes

Estas probabilidades foram definidas matematicamente conforme equação (2.4) na pág 42 de Collins e Lanza (2010). Esta função foi denominada ***prob.padr_class*** e é descrita a seguir.

**Sintaxe associada à função:**

**prob.padr_class(formula, data, n_class, n_var, n_decimal)**

**Argumentos:**

**formula**: Expressão que se refere à relação entre os indicadores da LCA e covariáveis na forma: indicadores ~ covariáveis.

**data**: Base de dados contendo variáveis indicadoras usadas na análise. Como a biblioteca ***poLCA*** é utilizada nesta função, então as variáveis devem estar codificadas como 1 e 2.

**n_class**: O número de classes do modelo LCA.

**n_var**: O número de variáveis indicadoras.

**n_decimal:** O número de casas decimais a serem utilizadas. Não é um argumento obrigatório. Quando o argumento não for utilizado, o *default* é: *n_decimal = 4*.

A função ***prob.padr_class*** é listada no Apêndice 4 deste tutorial.

A execução da função ***prob.padr_class*** utilizando o banco de dados *carcinoma_positivo* pode ser feita, passo-a-passo, usando:

```
f = cbind(A,B,C,D,E,F,G)~1
n_class = 2
n_var = 7
data = carcinoma_positivo
prob_padr_class = prob.padr_class(f, data, n_class, n_var)
print(prob_padr_class)
```

Equivalentemente, esta função pode ser executada através de:

```
prob_padr_class = prob.padr_class(cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 7)
print(prob_padr_class)
```

O resultado da função ***prob.padr_class*** para os dados sobre carcinoma é listado na Figura 3.27, que similarmente às funções anteriores retorna as probabilidades não condicionais (*Gamma Values*) e condicionais (*Rho Values*). Usando o comando *print( )* do objeto que contém o resultado desta função obtém-se as probabilidades estimadas de se observar cada padrão de resposta condicionalmente às classes latentes. As colunas *Prob.Padr_Class1 e Prob.Padr_Class2* contém as probabilidades correspondentes às duas classes do modelo LCA estimadas para os dados de carcinoma.

```
> prob_padr_cond_class = prob.padr_class(cbind(A,B,C,D,E,F,G)~1, carcinoma_positivo, 2, 7)

   Gamma Values
Class1 Class2
0.5012 0.4988

   Rho Values
           V1     V2     V3     V4     V5     V6     V7
Class1 1.0000 0.9831 0.7609 0.5411 0.9786 0.4227 1.0000
Class2 0.1165 0.3544 0.0000 0.0000 0.2229 0.0000 0.1165
> print(prob_padr_cond_class)
   V1 V2 V3 V4 V5 V6 V7 Prob.Padr_Class1 Prob.Padr_Class2
1   0  0  0  0  0  0  0           0.0000           0.3916
2   0  0  0  0  1  0  0           0.0010           0.1123
3   0  1  0  0  0  0  0           0.0013           0.2150
4   0  1  0  0  0  0  1           0.0013           0.0283
5   0  1  0  0  1  0  0           0.0609           0.0617
6   0  1  0  0  1  0  1           0.0609           0.0081
7   1  0  0  0  0  0  0           0.0000           0.0516
8   1  0  1  0  1  0  1           0.0033           0.0020
9   1  1  0  0  0  0  0           0.0013           0.0283
10  1  1  0  0  0  0  1           0.0013           0.0037
11  1  1  0  0  1  0  0           0.0609           0.0081
12  1  1  0  0  1  0  1           0.0609           0.0011
13  1  1  0  0  1  1  1           0.0446           0.0011
14  1  1  0  1  0  0  1           0.0016           0.0037
15  1  1  0  1  1  0  1           0.0719           0.0011
16  1  1  0  1  1  1  1           0.0526           0.0011
17  1  1  1  0  1  0  1           0.1939           0.0011
18  1  1  1  0  1  1  1           0.1420           0.0011
19  1  1  1  1  1  0  1           0.2287           0.0011
20  1  1  1  1  1  1  1           0.1674           0.0011
```

*Figura 3.27 Output da função prob.padr_class().*

# 4. Análise de Classes Latentes no Software WinLTA

O WinLTA é um software gratuito que pode ser usado para ajustar análise de classes latentes (LCA) e análise de transição latente (LTA). Neste tutorial estamos apresentando o uso do WinLTA apenas para implementação do modelo básico LCA. Esse aplicativo foi criado para Windows e desenvolvido por um grupo de pesquisadores da Universidade Estadual da Pensilvânia. Para ter acesso ao software, faz-se necessário o registro no site (http://methodology.psu.edu), o que permitirá o seu *download* gratuitamente no *link* http://methodology.psu.edu/downloads/winlta. Embora o *download* seja gratuito, não há suporte para seu uso.

## 4.1 Estrutura do banco de dados

Para leitura dos dados no WinLTA faz-se necessário que os mesmos estejam na estrutura contendo padrões de resposta e correspondentes frequências observadas. Para que esta estrutura de banco de dados possa ser obtida através dos dados originais, referentes a respostas individuais a cada um dos indicadores, pode-se utilizar o *Data Aggregation Program (DataAgg)*, que é disponibilizado no mesmo site do WinLT*A*. O nome do arquivo disponível para *download* é *DataAgg.exe*.

Para usar o programa *DataAgg* (DOS ou UNIX), as respostas individuais aos indicadores devem estar codificadas como inteiros positivos, começando em "1". Dados faltantes (*missing)* devem estar codificados como "0". Os dados devem estar separados por espaços ou tabulação (formato ASCII). Tanto o arquivo de dados fornecido quanto o arquivo gerado podem estar no formato *.txt* ou *.dat*. O banco de dados com respostas individuais deve conter apenas as variaveis observadas a serem usadas nas análises com LCA, sem incluir seus nomes (*labels*) na primeira linha do arquivo. O arquivo de dados gerado pelo programa *DataAgg* (arquivo *output*/ASC II) poderá ser utilizado no WinLTA.

A Figura 4.1 apresenta a tela onde devem ser fornecidas as informações para agregação dos dados. Os passos para que o processo possa ser realizado são os seguintes:

*1º:* digita-se o diretório onde se encontra o arquivo a ser agregado, em seguida pressiona-se a tecla *Enter*.

*2º:* digita-se o nome do arquivo que vai ser gerado, seguido de *Enter*.

*3º:* informa-se o número de observações, seguido de *Enter*.

*4º:* informa-se o número de variáveis indicadoras, seguido de *Enter*.

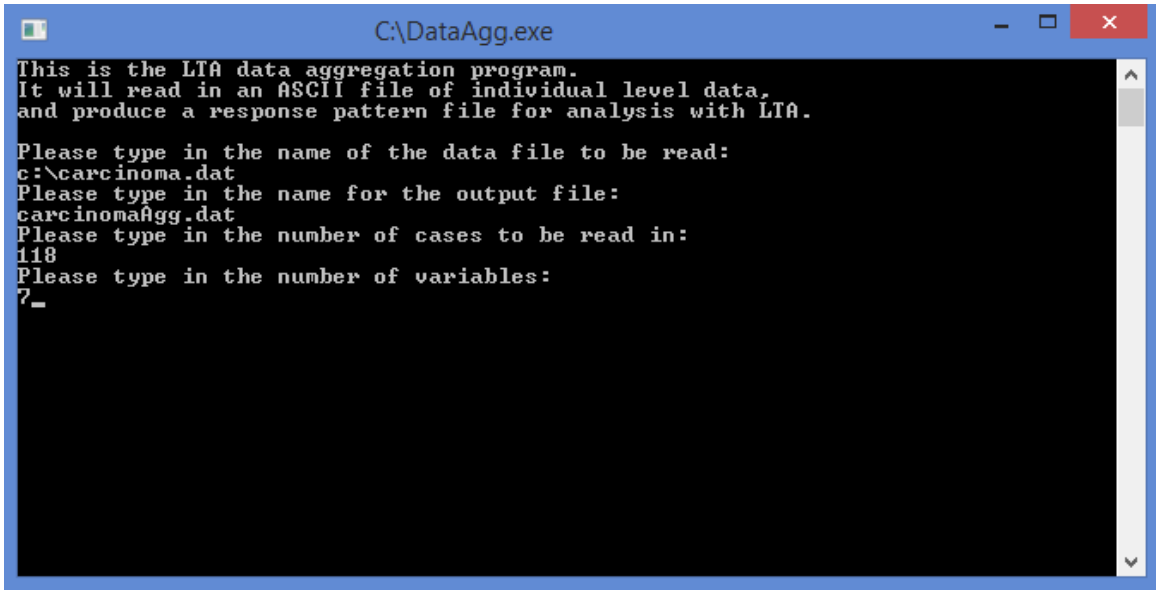


*Figura 4.1 Uso do Programa DataAgg para banco de dados carcinoma.*

Após fornecimento destas informações, a janela será fechada e o novo arquivo contendo os dados agregados estará disponível no diretório indicado (Figura 4.2).

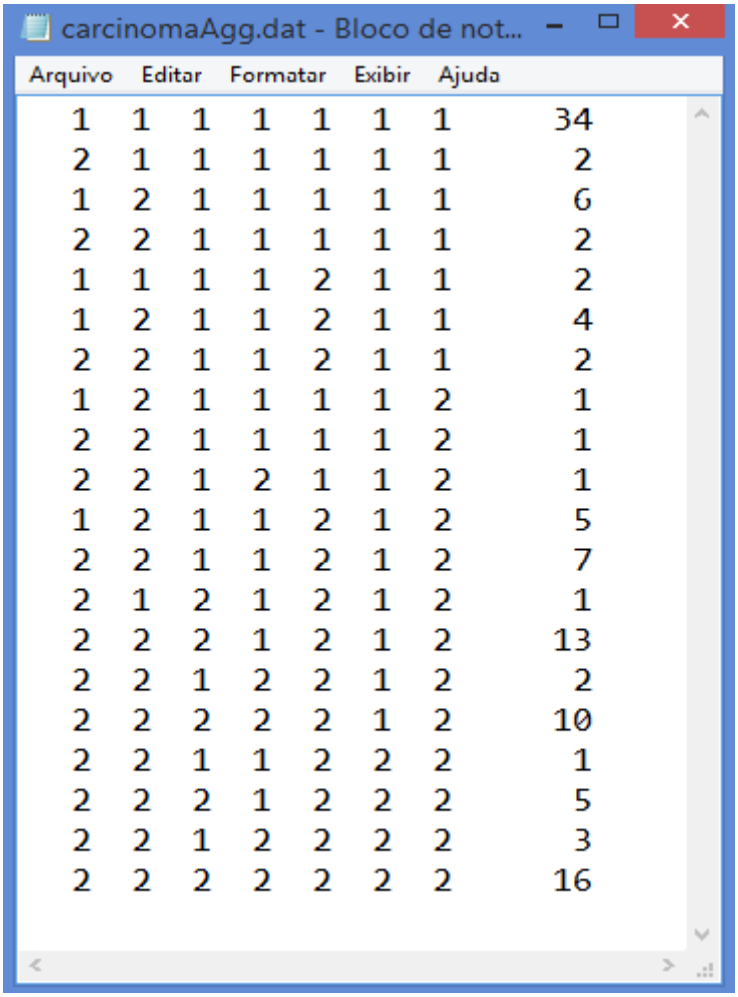


*Figura 4.2 Arquivo de dados gerado pelo programa DataAgg.*

## 4.2 Criação de um arquivo controle

Para executar o WinLTA deve-se criar um arquivo controle, que contém a definição do modelo e todas as especificações para a análise desejada. Para criação de um arquivo controle deve-se iniciar o WinLTA e clicar em *File e New Control File* (Figura 4.3).

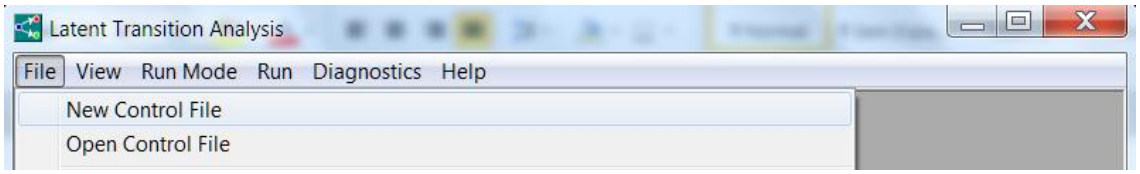


*Figura 4.3 Iniciando um arquivo de controle*

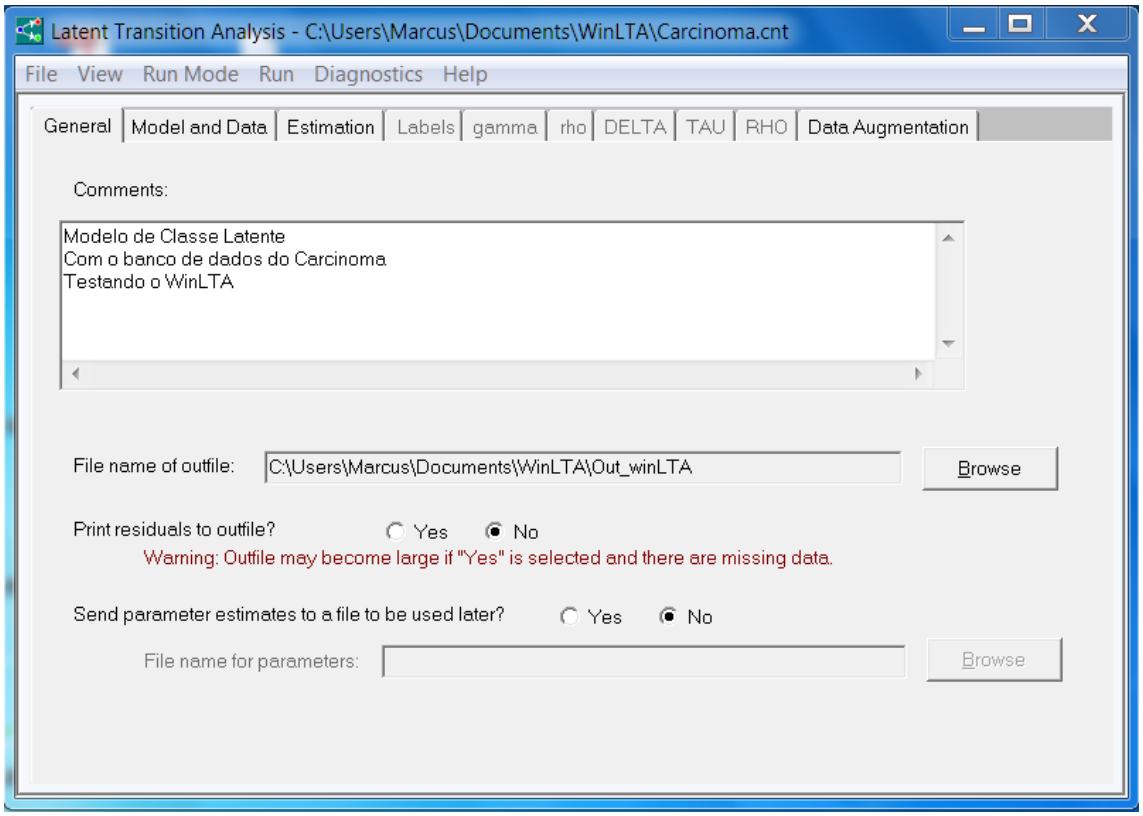


*Figura 4.4 Descrição da aba geral do arquivo de controle*

Com o arquivo de controle aberto, é necessário preencher cada uma das abas disponíveis no programa. Inicialmente deve-se selecionar a aba *General* (Geral) (Figura 4.4). Algumas informações que precisam ser acrescentadas nesta janela são:

- *Comments* : adiciona um título ou comentário no seu arquivo de saída (*output*);
- *File name of outfile*: especifica o local e o nome do arquivo que será gerado com *output*;
- *Print residual to outfile*? : escolha pela inclusão dos resíduos no *output.*

- *Send parameter estimates to a file be used later*? : Se a opção for *yes* os parâmetros estimados são salvos no arquivo que foi especificado para ser usado posteriormente (*File name for parameters*).

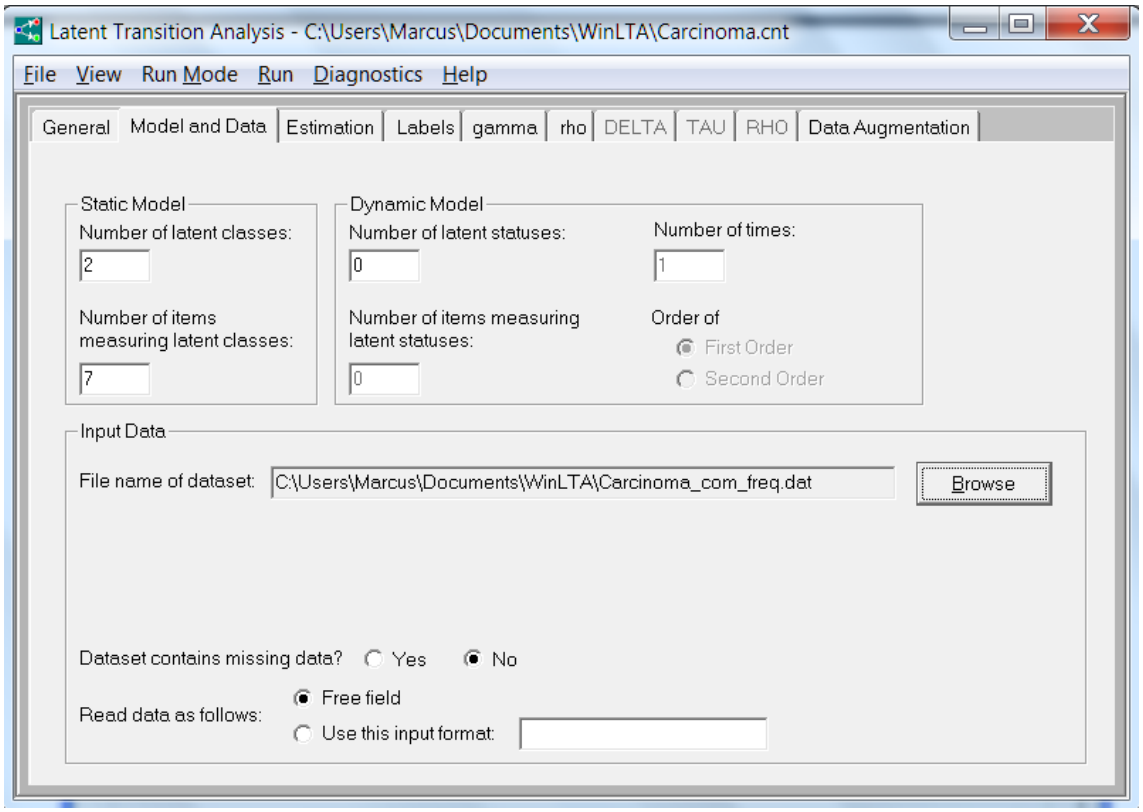


*Figura 4.5 Segunda aba do arquivo de controle: Model and Data*

A aba *Model and Data* (Modelo e dados) é a segunda a ser preenchida (Figura 4.5). Informações que precisam ser acrescentadas nesta janela são:

- *Static Model* – contém informações sobre as variáveis a serem acrescentadas no modelo LCA:
  - *Number of Latent Classes* (Número de classes latentes): Neste exemplo são 2 classes latentes.
  - *Number of items measuring latent classes* (Número de indicadores): Neste exemplo são 7 indicadores.
- *Dynamic Model* – contém informações sobre as variáveis a serem acrescentadas no modelo LTA (que não está sendo discutido neste tutorial):
  - *Number of latent statuses*: "0" indica que esta opção está desativada.
  - *Number of times*: "1" para indicar LCA.
- *Input Data*
  - *File name of Dataset*: especifica o local onde se encontra o arquivo de dados contendo os padrões de resposta (dados agregados).

- *Dataset contains missing data:* se o conjunto de dados contiver dados faltantes, então deve-se marcar a opção *yes*. Vale lembrar que o código para dados faltantes é 0.
- *Read data as follows*: identifica o formato do conjunto de dados.

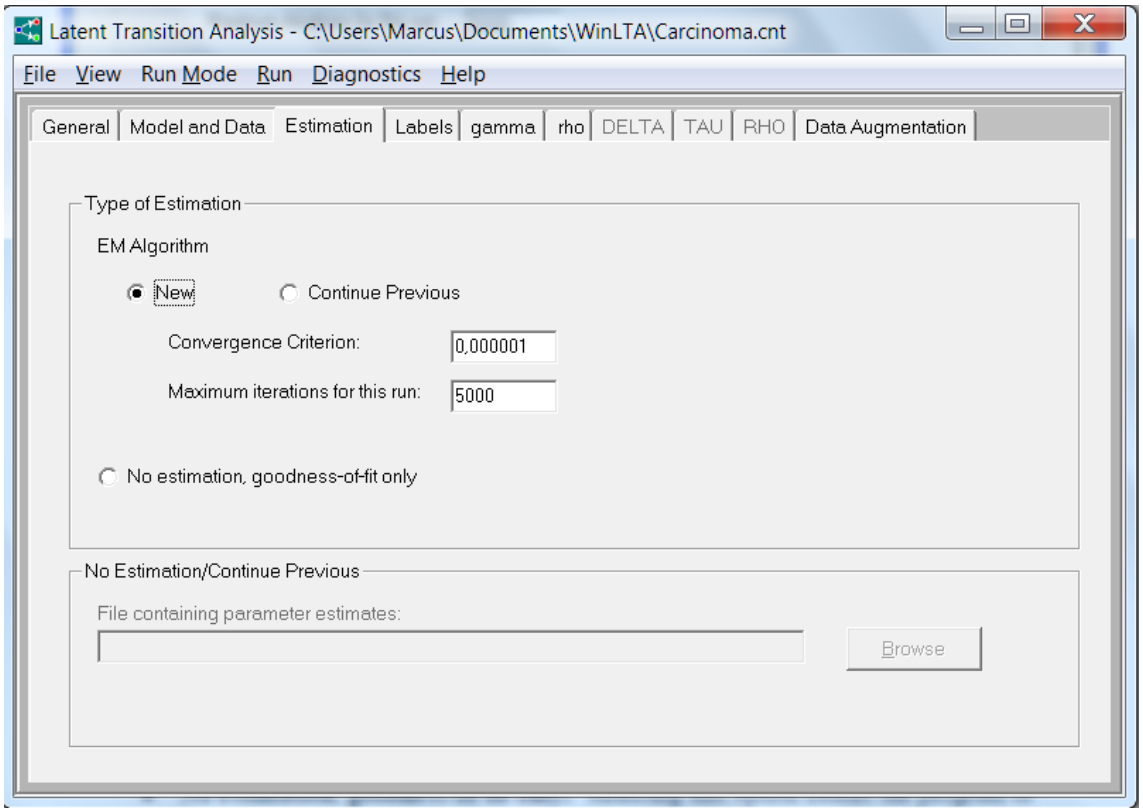


*Figura 4.6 Terceira aba do arquivo de controle: Estimation*

A terceira aba - *Estimation* - inclui informações sobre o processo de estimação a ser considerado (Figura 4.6) na LCA, incluindo as seguintes opções:

- *Type of Estimator - EM Algorithm*:
  - *New* – define o uso de novos valores iniciais dos parâmetros;
  - *Continue previous* – define valores iniciais dos parâmetros obtidos em análise anterior;
  - *Convergence Criterion*: o *default* é desvio médio absoluto menor ou igual a $10^{-6}$
  - *Maximum iterations for this run*: o número *default* de iterações é 5000.
- *No estimation, goodness-of-fit only*: – não apresenta resultados das estimativas dos parâmetros, mas fornece somente estatísticas de bondade do ajuste do modelo.
- *No Estimation/Continue Previous*: neste caso precisa-se fornecer o arquivo que contenha estimativas dos parâmetros já obtidas em processo anterior.

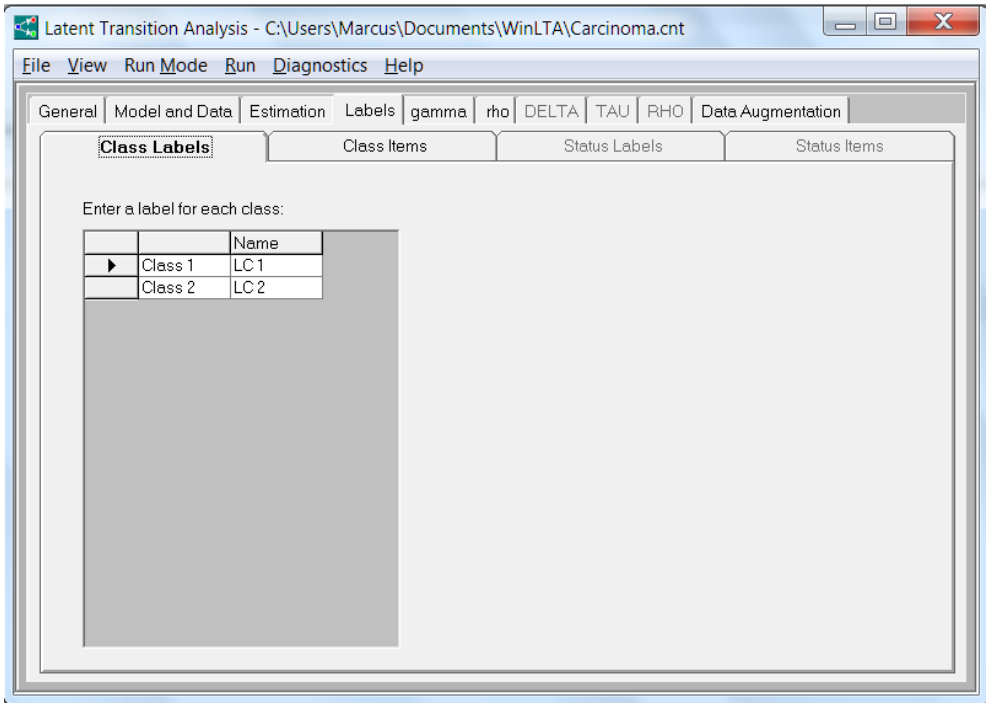


*Figura 4.7 Quarta aba do arquivo de controle: Labels.*

Na aba *Labels* precisa-se preencher informações sobre o nome das classes das variáveis latentes (*Class Labels*) e nome dos indicadores (*Class Items*). Os nomes das variáveis devem conter, no máximo, 8 caracteres (Figura 4.7).

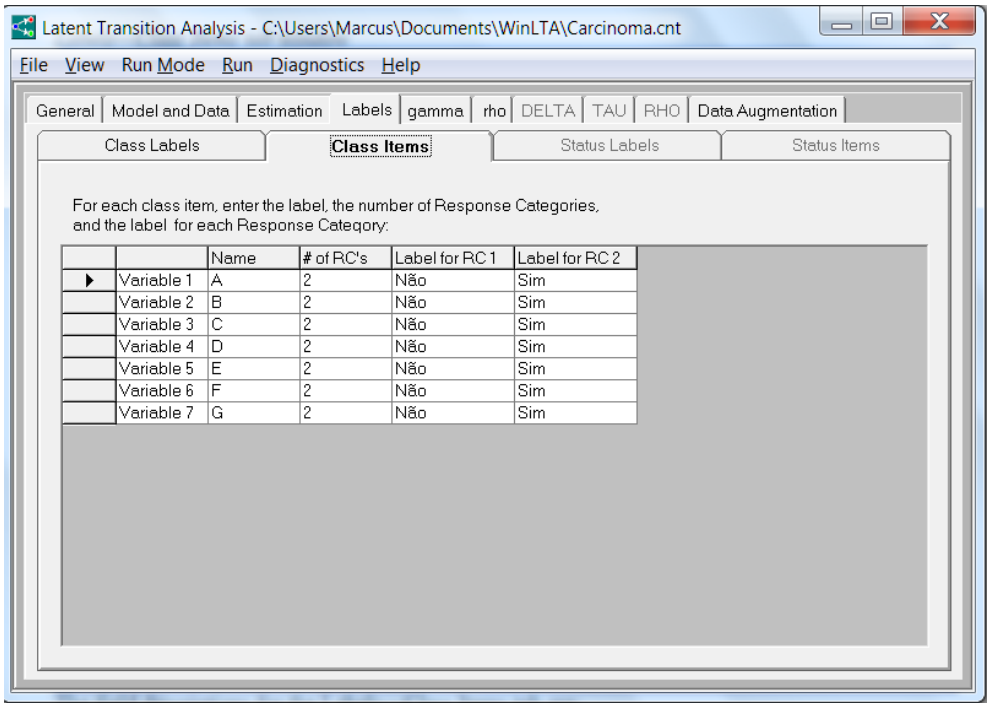


*Figura 4.8 Rotulando as variáveis indicadoras do modelo.*

É possível renomear os indicadores (*Name*), definir o número de categorias associadas a estes indicadores *(# of RC's*) e denominar cada uma das categorias (*Labels for RC1 e*

*RC2*) (Figura 4.8). No nosso exemplo todos os indicadores são binários, com categorias denominadas *Não* e *Sim*.

No software WinLTA é necessário o preenchimento de informações sobre cada um dos parâmetros do modelo nas abas denominadas *gamma* e *rho* (Figuras 4.9 e 4.10). Os parâmetros *gamma* referem-se às probabilidades não condicionais ou prevalências das classes latentes. No lado esquerdo da Figura 4.9 são definidas restrições para os parâmetros *gamma*. O valor 1 indica que os parâmetros são livres e podem ser estimados. Para especificar que o parâmetro é fixo deve-se usar o valor 0. Neste exemplo, todos os parâmetros são livres. No lado direito da Figura 4.9, na coluna *Starting Value* são especificados os valores iniciais para os parâmetros *gamma*. Como estes parâmetros são complementares e representam probabilidades, sua soma deve ser igual a 1.

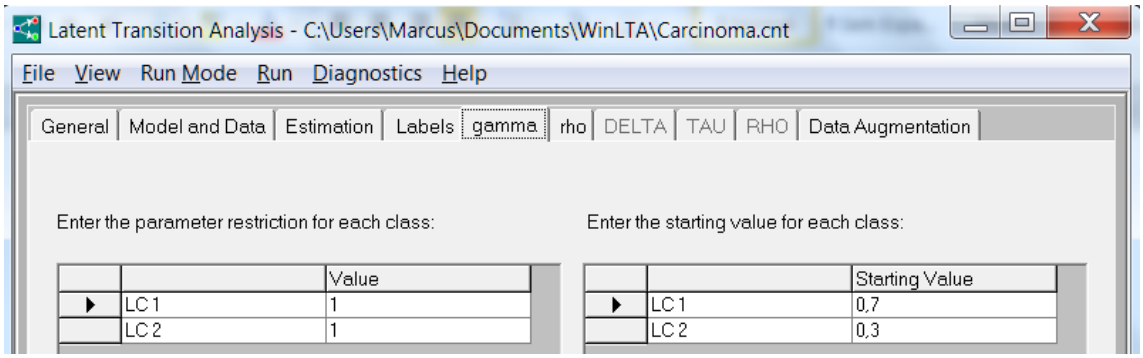


*Figura 4.9 Aba para definir as restrições e valores iniciais dos parâmetros gamma*

Os *rho*'s são as probabilidades de resposta ao item, ou seja, são probabilidades condicionais às classes latentes. No lado esquerdo da Figura 4.10 são definidas restrições para os parâmetros *rho*. Novamente o valor 1 indica que os parâmetros são livres e o valor 0 indica que os parâmetros são fixos. Para definir restrições de igualdade entre diferentes parâmetros, deve-se usar os valores iguais ou maiores do que 2. Neste exemplo, todos os parâmetros são livres. No lado direito da Figura 4.10 é necessária a especificação dos valores iniciais para os parâmetros *rho*'s. Como estes parâmetros são complementares dentro de uma mesma classe latente e representam probabilidades, sua soma deve ser igual a 1.

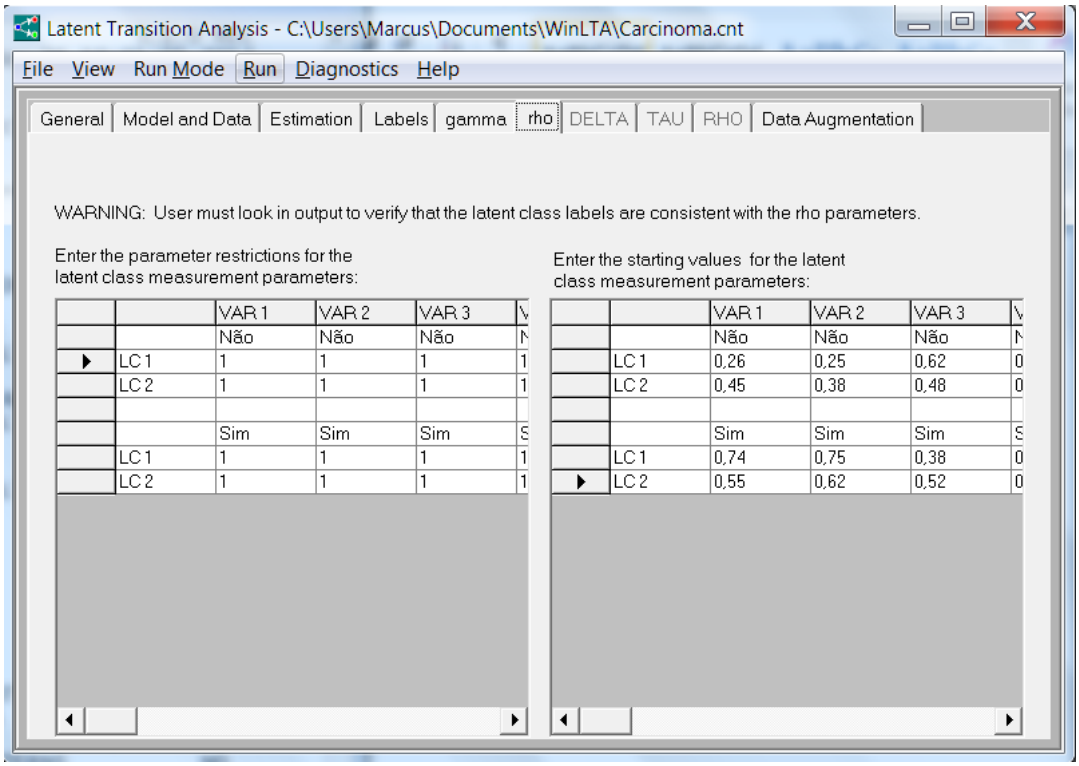


*Figura 4.10 Aba para definir as restrições e valores iniciais dos parâmetros rho*

A Figura 4.11 apresenta um exemplo de valores iniciais definidos para *rho*. Note que as linhas 2 e 6 são complementares, somando 1 para cada uma das colunas.

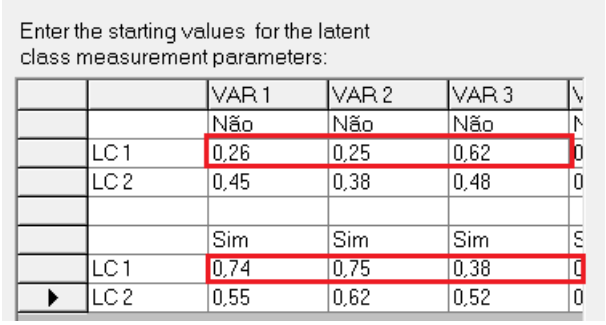


*Figura 4.11 Valores iniciais complementares de rho.*

Após a conclusão de todas as especificações do arquivo controle, este pode ser salvo. Para Salvar clique em *File* e, posteriormente, em *Save* (Figura 4.12) para obtenção de arquivo com extensão *.cnt*.

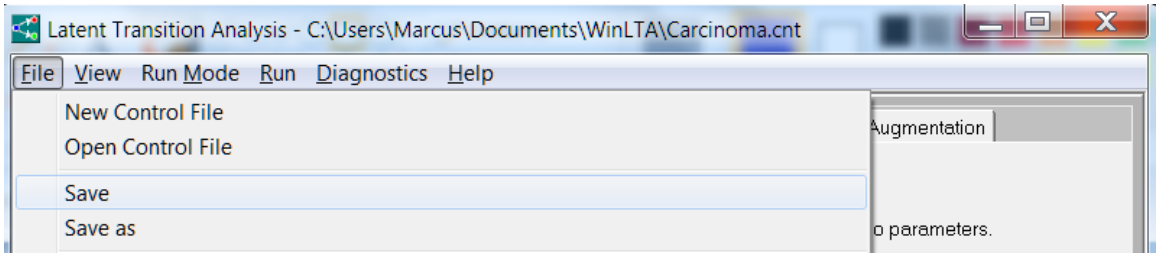


*Figura 4.12 Salvado o arquivo de controle (.cnt)*

Após ter o arquivo de controle pronto, este deve ser executado no WinLTA para implementar a LCA e gerar os outputs. Para isto clique em *Run* (Figura 4.13).

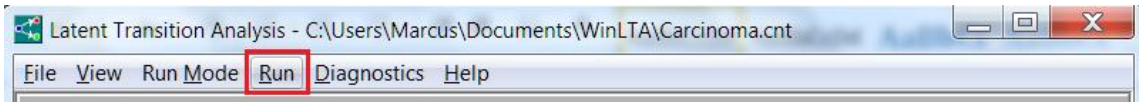


*Figura 4.13 Executando o modelo no WinLTA*

Caso não haja erros, então a mensagem apresentada na Figura 4.14 é disponibilizada na tela.

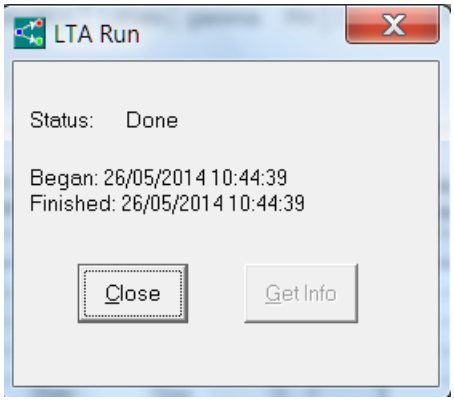


*Figura 4.14 Aviso de finalização do ajuste do modelo no WinLTA*

## 4.3 Resultados no WinLTA

O *output* do WinLTA é bastante auto-explicativo. Para visualizar o *output* clique em *View* e em *Current LTA Outfile* (Figura 4.15). Para facilitar o entendimento vamos apresentá-lo separadamente em 5 *displays* (Figuras 4.16-4.20).

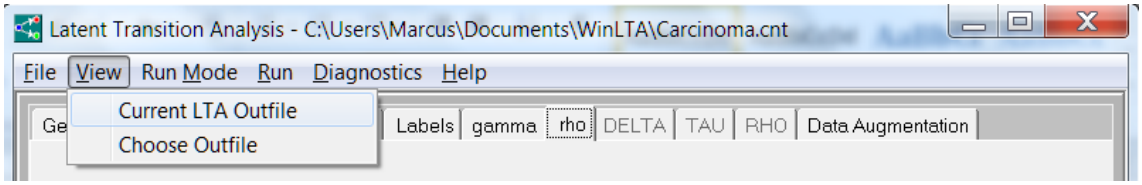


*Figura 4.15 Abrindo o output gerado pelo WinLTA.*

A Figura 4.16 apresenta as informações gerais sobre o arquivo, como a data de criação do *output* e os comentários que foram definidos na primeira aba do arquivo de controle.

```
 PROGRAM VERSION:                          3.1.0 REL
 PROGRAM STARTED:          Mon May 26 10:44:39 2014

* Modelo de Classe Latente
* Com o banco de dados do Carcinoma
* Testando o WinLTA

 ***************************************************
```

*Figura 4.16 Output do WinLTA: Parte 1*

O *output* apresenta ainda informações sobre a localização dos arquivos: de controle, do banco de dados e do output, bem como informações básicas sobre o modelo (LCA ou LTA), números de variáveis observadas e latentes, número de padrões de resposta e total de observações (*subjects*), dentre outras (Figura 4.17).

```
**************************************************

 INFORMATION ABOUT THIS JOB:

 RUN TYPE: PARAMETER ESTIMATION BY EM

 CONTROL DATA READ FROM FILE:
C:\Carcinoma.cnt

 DATA ANALYZED IN THIS RUN READ FROM FILE:
C:\Carcinoma_com_freq.dat

 OUTPUT SAVED IN FILE:
C:\WinLTA\Out_winLTA.txt

 STATIC LATENT VARIABLE                    YES
 NUMBER OF LATENT CLASSES                    2
 NUMBER OF MANIFEST ITEMS                    7

 DYNAMIC LATENT VARIABLE                    NO

 NUMBER OF SUBJECTS                        118
 NUMBER OF UNIQUE RESPONSE PATTERNS         20
 MAXIMUM NUBMER OF ITERATIONS             5000
 CONVERGENCE CRITERION          .00000100000000
 MISSING DATA IN RESPONSE PATTERNS          NO
 PRINT RESIDUALS                            NO

**************************************************
```

*Figura 4.17 Output do WinLTA: Parte 2*

A lista de restrições de todos os parâmetros do modelo é também apresentada (Figura 4.18). Neste exemplo todos os parâmetros são estimados livremente, logo têm o valor 1 associado a eles.

```
THE FOLLOWING PARAMETER RESTRICTIONS HAVE BEEN SPECIFIED
WHERE 0=FIXED TO START VALUE
      1=FREE
      2 OR GREATER MEANS CONSTRAINED EQUAL TO ANY OTHER
        PARAMETER WITH THE SAME DESIGNATION

**************************************************

LITTLE RHO PARAMETERS
LITTLE RHOS ARE PROBABILITIES OF RESPONSES
TO ITEMS MEASURING THE STATIC LATENT VARIABLE
CONDITIONAL ON LATENT CLASS MEMBERSHIP

          RESPONSE CATEGORY  1

          | V N | V N | V N | V N | V N | V N | V N |
          | A ã | A ã | A ã | A ã | A ã | A ã | A ã |
          | R o | R o | R o | R o | R o | R o | R o |
          |     |     |     |     |     |     |     |
          | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
          |     |     |     |     |     |     |     |
          |     |     |     |     |     |     |     |
LC 1        1     1     1     1     1     1     1
LC 2        1     1     1     1     1     1     1

          RESPONSE CATEGORY  2

          | V S | V S | V S | V S | V S | V S | V S |
          | A i | A i | A i | A i | A i | A i | A i |
          | R m | R m | R m | R m | R m | R m | R m |
          |     |     |     |     |     |     |     |
          | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
          |     |     |     |     |     |     |     |
          |     |     |     |     |     |     |     |
LC 1        1     1     1     1     1     1     1
LC 2        1     1     1     1     1     1     1
  **************************************************

  GAMMA PARAMETER RESTRICTIONS
  GAMMAS ARE UNCONDITIONAL PROBABILITIES OF MEMBERSHIP
  IN EACH LATENT CLASS OF THE STATIC LATENT VARIABLE

  LC 1        1
  LC 2        1

  **************************************************
```

*Figura 4.18 Output do WinLTA: Parte 3*

A Figura 4.19 apresenta informações sobre os valores iniciais especificados pelo usuário e o histórico de iterações do modelo, incluindo o desvio médio absoluto (*MAD*, em inglês), que é a diferença da média absoluta entre as estimativas dos parâmetros resultantes da iteração atual e as estimativas dos parâmetros da iteração anterior.

```
          START VALUES

*****************************************************

LITTLE RHO PARAMETERS
LITTLE RHOS ARE PROBABILITIES OF RESPONSES
TO ITEMS MEASURING THE STATIC LATENT VARIABLE
CONDITIONAL ON LATENT CLASS MEMBERSHIP

          RESPONSE CATEGORY  1

          | V N | V N | V N | V N | V N | V N | V N |
          | A ã | A ã | A ã | A ã | A ã | A ã | A ã |
          | R o | R o | R o | R o | R o | R o | R o |
          |     |     |     |     |     |     |     |
          | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
          |     |     |     |     |     |     |     |
LC 1      0.260 0.250 0.620 0.790 0.510 0.630 0.310
LC 2      0.450 0.380 0.480 0.260 0.720 0.400 0.720

          RESPONSE CATEGORY  2

          | V S | V S | V S | V S | V S | V S | V S |
          | A i | A i | A i | A i | A i | A i | A i |
          | R m | R m | R m | R m | R m | R m | R m |
          |     |     |     |     |     |     |     |
          | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
          |     |     |     |     |     |     |     |
LC 1      0.740 0.750 0.380 0.210 0.490 0.370 0.690
LC 2      0.550 0.620 0.520 0.740 0.280 0.600 0.280

*****************************************************
GAMMA PARAMETERS
GAMMAS ARE UNCONDITIONAL PROBABILITIES OF MEMBERSHIP IN EACH LATENT CLASS
OF THE STATIC LATENT VARIABLE

LC 1       0.700
LC 2       0.300


  ITERATION HISTORY

 *****************************************************

 STARTING G-SQUARED=   555.266

ITER-   MAD          ITER-   MAD          ITER-   MAD
ATION                ATION                ATION

    1  .1500182806      2  .1800537246      3  .0822739469
    4  .0207630813      5  .0084526819      6  .0050090395
    7  .0035698067      8  .0027162516      9  .0021238564
   10  .0016917391     11  .0013781152     12  .0011561335
   13  .0010035668     14  .0009020744     15  .0008376765
   16  .0008005251     17  .0007841379     18  .0007845263
   19  .0007994513     20  .0008278620     21  .0008694827
   22  .0009246227     23  .0009947240     24  .0010784576
   25  .0011747467     26  .0012805021     27  .0013893255
   28  .0014901261     29  .0015664068     30  .0015976292
   31  .0015641569     32  .0014555611     33  .0012783528
   34  .0010569653     35  .0008266243     36  .0006141196
   37  .0004372785     38  .0003012465     39  .0002025228
   40  .0001338056     41  .0000873493     42  .0000565617
   43  .0000364289     44  .0000233794     45  .0000149699
   46  .0000095711     47  .0000061134     48  .0000039025
   49  .0000024902     50  .0000015886     51  .0000010132
   52  .0000006462
.
```

*Figura 4.19 Output do WinLTA: Parte 4*

As estatísticas de bondade do ajuste do modelo apresentadas são a estatística $G^2$ e seus correspondentes graus de liberdade. As estimativas dos parâmetros *rho*, ou seja, as probabilidades condicionais ou probabilidade de resposta ao item para cada classe latente, bem como estimativas dos parâmetros *gamma*, ou seja, das probabilidades não condicionais ou prevalências das classes latentes, são apresentadas (Figura 4.20).

```
**************************************************

MODEL FIT

G-Squared Test of Model Fit:                62.366
Degrees of Freedom:                           112

**************************************************


 **WARNING**: BE SURE TO INTERPRET THE LATENT CLASSES CAREFULLY
              BASED ON THE ESTIMATED RHO PARAMETERS REPORTED BELOW.
              YOU MAY WISH TO CHANGE THE LABELS YOU PREVIOUSLY ASSIGNED
              TO THE LATENT CLASSES IN ORDER TO MAKE THEM
              CONSISTENT WITH YOUR INTERPRETATION.

**************************************************

LITTLE RHO PARAMETERS
LITTLE RHOS ARE PROBABILITIES OF RESPONSES
TO ITEMS MEASURING THE STATIC LATENT VARIABLE
CONDITIONAL ON LATENT CLASS MEMBERSHIP

          RESPONSE CATEGORY  1

         | V N | V N | V N | V N | V N | V N | V N |
         | A ã | A ã | A ã | A ã | A ã | A ã | A ã |
         | R o | R o | R o | R o | R o | R o | R o |
         |     |     |     |     |     |     |     |
         | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
         |     |     |     |     |     |     |     |
LC 1     0.000 0.017 0.239 0.459 0.021 0.577 0.000
LC 2     0.883 0.646 1.000 1.000 0.777 1.000 0.884

          RESPONSE CATEGORY  2

         | V S | V S | V S | V S | V S | V S | V S |
         | A i | A i | A i | A i | A i | A i | A i |
         | R m | R m | R m | R m | R m | R m | R m |
         |     |     |     |     |     |     |     |
         | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
         |     |     |     |     |     |     |     |
LC 1     1.000 0.983 0.761 0.541 0.979 0.423 1.000
LC 2     0.117 0.354 0.000 0.000 0.223 0.000 0.116

**************************************************

GAMMA PARAMETERS
GAMMAS ARE UNCONDITIONAL PROBABILITIES OF MEMBERSHIP IN EACH LATENT CLASS
OF THE STATIC LATENT VARIABLE

LC 1      0.501
LC 2      0.499

PROGRAM FINISHED: Mon May 26 10:44:39 2014
ELAPSED TIME:    0 HOURS,    0 MINUTES,    0 SECONDS.
```

*Figura 4.20 Output do WinLTA: Parte 5*

A ênfase deste tutorial é com a implementação de LCA básica em três diferentes softwares. Como pôde ser observado, resultados similares podem ser obtidos nos software **Mplus**, **R** e **winLTA**. A análise dos resultados requer conhecimento sobre a definição da metodologia e interpretação de seus parâmetros, que não é o foco deste trabalho. Para conhecer um pouco mais sobre LCA recomendamos a leitura de Collins e Lanza (2010).

## Referências Bibliográficas


Asparouhov, T., Muthén, B. O. (2012). *Using Mplus TECH11 and TECH14 to test the number of latent classes.* Mplus Web Notes: No. 14, May 22.

Beath, K. (2014). *RandomLCA: Random Effects Latent Class Analysis, R(>= 2.15.1),* Available at http://cran.r-project.org/web/packages/randomLCA/index.html.

Clark, S. L.; Muthén, B. O. (2009). *Relating Latent Class Analysis Results to Variables not Included in the Analysis.* University of California, Los Angeles.

Collins, L. M.; Lanza, S. T. (2010). *Latent Class and Latent Transition Analysis: With Applications in the Social, Behavioral, and Health Sciences.* Ed. Wiley.

Collins, L. M., Lanza, S. T., Schafer, J. L., & Flaherty, B. P. (2002). *WinLTA users' guide (Version 3.0).* University Park: The Methodology Center, Penn State. Retrieved from http://methodology.psu.edu

Linzer, D. A.; Lewis, J. B. (2011). poLCA: An R Package for Polytomous Variable Latent Class Analysis. Journal of Statistical Software**,** 42(10), 1-29. URL http://www.jstatsoft.org/v42/i10/.

Meyer D., Dimitriadou E., Hornik K., Weingessel A., Leisch F. (2014). *e1071: Misc Functions of the Department of Statistics (e1071),* TU Wien. R package version 1.6-3. http://CRAN.R-project.org/package=e1071

Muthén, L.K.; Muthén, B.O. (1998-2010). *Mplus User's Guide*. Sixth Edition. Los Angeles, CA: Muthén&Muthén.

Schafer, J. L.; Kang, J. (2013). *lcca package for R users' guide* (Version 1.1.0). University Park: The Methodology Center, Penn State. Available from methodology.psu.edu.

# APÊNDICE 1

## Função *entropy*

```
entropy = function(formula, data, n_class, n_decimal = 4){
  library(poLCA)
  library(psych)
  lca_model = poLCA(formula, data, nclass = n_class, verbose =
F)
  prob_posterior = lca_model$posterior

  value = (-prob_posterior) %*% (log(t(prob_posterior + 1e-20)))

  E = 1 - (tr(value)) / ((nrow(prob_posterior)) *
log(ncol(prob_posterior)))

  E = round(E, n_decimal)
  cat(paste("Entropia = ", E, "\n"))
  return(E)
}
```

# APÊNDICE 2

## Função *prob.class_padr*

```
prob.class_padr = function(formula, data, n_class, n_decimal =
4){
  #extraindo os Padrões
   library(poLCA)
   analise1 = poLCA(f, data, nclass = n_class, verbose = F)
   n_var = length(analise1$predcell[1,])-2
   padr = analise1$predcell[,1:n_var]
   freq = analise1$predcell[,(n_var+1)]
   padr[padr == 1] = 0
   padr[padr == 2] = 1

 #Extraindo as probabilidades condicionais e não
 #condicionais
    library(randomLCA)
    analise4 = randomLCA(padr, freq, nclass = n_class)
    mtrx_rho = round(analise4$outcomep * 1, n_decimal)
    vec_gama = round(analise4$classp * 1, n_decimal)

  #Definindo variaveis importantes
    n_padr = length(padr[,1])
    numerador = matrix(0,ncol = n_class, nrow = n_padr)
    denominador = matrix(0,nrow = n_padr)

  #NUMERADOR  DA FORMULA{
   for(g in 1:n_class){
      for(p in 1:n_padr){
        prod = 1
      for(j in 1:n_var){
        if (padr[p,j] == 1){
            if(mtrx_rho[g,j] > 0){
          prod = prod * mtrx_rho[g,j]               }
          }
        else{
         if(mtrx_rho[g,j] < 1){
         prod = prod * (1 - mtrx_rho[g,j])
            }
          }
        }
        numerador[p,g] = prod * vec_gama[g]
      }
    }


  #DENOMINADOR DA FORMULA
    for (g in 1:n_class){
      denominador = denominador + numerador[,g]
```

```
    }

  #Calculando a probabilidade de cada padrão
    prob.pad = padr
    range = length(n_padr) + 1

    for (g in 1:n_class){
      for (p in 1:n_padr){
        prob.pad[p,(n_var + g)] = numerador[p,g] /
denominador[p]
      }
    }

  #Definido os nomes das colunas e linhas
    name_gama = rep("1", n_class)
    for(i in 1:n_class) name_gama[i] = paste0("Class", i)

    name_rho = rep("1", n_var)
    for(i in 1:n_var) name_rho[i] = paste0("V", i)

    names_prob_padr = rep("1", n_var+n_class)

    for(i in 1:(n_var+n_class)){
      if(i <= n_var) names_prob_padr[i] = paste0("V", i)
      else names_prob_padr[i] = paste0("Prob.Class", i-n_var)
    }

    names(vec_gama) = name_gama
    colnames(mtrx_rho) = name_rho
    rownames(mtrx_rho) = name_gama
    colnames(prob.pad) = names_prob_padr

  #Exibindo os Roh's e os gamas
    cat("\n", "  Gamma Values", "\n")
    print(vec_gama)
    cat("\n","  Rho Values", "\n")
    print(mtrx_rho)


  return(round(prob.pad, n_decimal))
                                  }
```

# APÊNDICE 3

## Função *prob.padr*

```
prob.padr = function(formula, data, n_class,
          n_decimal = 4){
  #extraindo os Padrões
    library(poLCA)
      analise1 = poLCA(f, data, nclass = n_class,
          verbose = F)
      n_var = length(analise1$predcell[1,])-2
      padr = analise1$predcell[,1:n_var]
      freq = analise1$predcell[,n_var + 1]

   #Extraindo as probabilidades condicionais e
   # não condicionais
    library(randomLCA)
    padr[padr == 1] = 0
    padr[padr == 2] = 1
    analise4 = randomLCA(padr, freq, nclass =
          n_class)
   mtrx_rho = round(analise4$outcomep * 1,
          n_decimal)
    vec_gama = round(analise4$classp * 1,
          n_decimal)

  n_padr = length(padr[,1])
  prod2 = matrix(0,ncol = n_class, nrow = n_padr)
  prob_padr_ind_class = padr
  prob_padr_ind_class[,n_var+1] = matrix(0,ncol =
          1, nrow = n_padr)

  #Produtorio1
    for(g in 1:n_class){
      for(p in 1:n_padr){
        prod = 1

   #Produtório 2
        for(j in 1:n_var){
          if (padr[p,j] == 1){
            if(mtrx_rho[g,j] > 0){
              prod = prod * mtrx_rho[g,j]
          }
          }
          else{
            if(mtrx_rho[g,j] < 1){
              prod = prod * (1 - mtrx_rho[g,j])
          }
          }
```

```
        }
        if(vec_gama[g] > 0){
          prod2[p,g] = prod * vec_gama[g]
          }
      }
      prob_padr_ind_class[,n_var+1] =
          prob_padr_ind_class[,n_var+1] +
          prod2[,g]
    }

  #Definido os nomes das colunas e linhas
    name_gama = rep("1", n_class)
    for(i in 1:n_class) name_gama[i] =
          paste0("Class", i)

    name_rho = rep("1", n_var)
    for(i in 1:n_var) name_rho[i] = paste0("V",
          i)

    name_prob= rep("1", n_var)
    for(i in 1:(n_var)) name_prob[i] =
          paste0("V", i)
    name_prob[n_var+1] = paste0("Prob.Padr")

    names(vec_gama) = name_gama
    colnames(mtrx_rho) = name_rho
    rownames(mtrx_rho) = name_gama
    colnames(prob_padr_ind_class) = name_prob

  #Exibindo os Roh's e os gamas
    cat("\n", "  Gamma Values", "\n")
    print(vec_gama)
    cat("\n","  Rho Values", "\n")
    print(mtrx_rho)

  return(round(prob_padr_ind_class, n_decimal))
}
```

# APÊNDICE 4

## Função *prob.padr_class*

```
prob.padr_class = function(formula, data, n_class, n_decimal =
4){

  #extraindo os Padrões
    library(poLCA)
    analise1 = poLCA(formula, data, nclass = n_class, verbose =
F)
    n_var = length(analise1$predcell[1,])-2
    padr = analise1$predcell[,1:n_var]
    freq = analise1$predcell[,n_var+1]

  #Extraindo as probabilidades condicionais e não condicionais
    library(randomLCA)
    padr[padr == 1] = 0
    padr[padr == 2] = 1

    analise4 = randomLCA(padr, freq, nclass = n_class)

    mtrx_rho = round(analise4$outcomep * 1, n_decimal)
    vec_gama = round(analise4$classp * 1, n_decimal)

  n_padr = length(padr[,1])
  prob_cond_class = matrix(0,ncol = n_class, nrow = n_padr)
  prob_padr_cond_class = padr

  #Produtorio1
    for(g in 1:n_class){
      for(p in 1:n_padr){
        prod = 1

        #Produtório 2
          for(j in 1:n_var){

            if (padr[p,j] == 1){
              if(mtrx_rho[g,j] > 0){
                prod = prod * mtrx_rho[g,j]
              }
            }
            else{
              if(mtrx_rho[g,j] < 1){
                prod = prod * (1 - mtrx_rho[g,j])
              }
            }
          }
```

```
        prob_cond_class[p,g] = prod
      }
    }

  #Agregando as prbabilidades em uma unica tabela
    for(j in 1:n_class){
      prob_padr_cond_class[n_var + j] = prob_cond_class[,j]
    }

  #Definido os nomes das colunas e linhas
  name_gama = rep("1", n_class)
  for(i in 1:n_class) name_gama[i] = paste0("Class", i)

  name_rho = rep("1", n_var)
  for(i in 1:n_var) name_rho[i] = paste0("V", i)

  names_prob = rep("1", n_var+n_class)

  for(i in 1:(n_var+n_class)){
    if(i <= n_var) names_prob[i] = paste0("V", i)
    else names_prob[i] = paste0("Prob.Padr_Class", i-n_var)
  }

  names(vec_gama) = name_gama
  colnames(mtrx_rho) = name_rho
  rownames(mtrx_rho) = name_gama
  colnames(prob_padr_cond_class) = names_prob

  #Exibindo os Roh's e os gamas
  cat("\n", "  Gamma Values", "\n")
  print(vec_gama)
  cat("\n","  Rho Values", "\n")
  print(mtrx_rho)

  return(round(prob_padr_cond_class, n_decimal))
}
```