UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DEPARTAMENTO DE ESTATÍSTICA

MODELAGEM COM VARIÁVEIS LATENTES CONTÍNUAS E CATEGÓRICAS

Um tutorial usando *software* R

Marcos Aurélio Eustorgio Filho[1]
Leila Denise Alves Ferreira Amorim[1]

[1] Departamento de Estatística, Universidade Federal da Bahia, Salvador Bahia.

Salvador
2021

SUMÁRIO

1 RESUMO

A modelagem com variáveis latentes é uma metodologia cujo uso vem crescendo ao longo do tempo, sobretudo devido à sua capacidade de aplicação no estudo de diversas questões científicas importantes para as quais os métodos estatísticos mais tradicionais podem não ser adequados. Alguns exemplos da aplicabilidade dessa metodologia envolvem estudos acerca de temas como inteligência, padrões comportamentais de indivíduos e ainda qualidade de vida, que apesar de não serem diretamente observáveis, podem se manifestar através de outras variáveis, tornando possível a definição de construtos e o estudo da sua relação com as variáveis observadas que os mensuram. Dentre os casos particulares de modelagem com variáveis latentes incluem-se os modelos de equações estruturais (SEM, *Structural Equation Modeling*, em inglês), a análise de classes latentes (LCA, *Latent Class Analysis*, em inglês), e a análise de perfis latentes (LPA, *Latent Profile Analysis*, em inglês). Essas metodologias têm uma grande vantagem se comparadas com as técnicas tradicionais por permitirem múltiplas relações entre as variáveis que compõe o modelo. A aplicação de métodos para modelar variáveis latentes necessita do uso de algum *software* estatístico, mas a grande maioria dos *softwares* que implementam esses métodos requerem o pagamento de licenças anuais ou semestrais. Contudo uma vasta quantidade de métodos para modelagem de variáveis latentes tem sido incorporada no *software* estatístico R, gratuito e de código livre, possibilitando a implementação de técnicas para estudar os casos particulares de modelagem com variáveis latentes definidos nesse trabalho. Neste trabalho, um dos objetivos centrais é entender quais metodologias são mais adequadas a cada tipo de problema, analisar a importância da verificação dos pressupostos dos métodos nas conclusões obtidas a partir do ajuste dos modelos, e fornecer um breve tutorial de aplicação dessas metodologias. Deste modo, espera-se contribuir para maior divulgação e utilização correta de metodologias envolvendo variáveis latentes, de forma gratuita, por pesquisadores de diversas áreas do conhecimento.

Palavras Chaves: Modelagem, Variáveis latentes, R, Tutorial

2 INTRODUÇÃO

A modelagem com variáveis latentes é uma metodologia cujo uso vem crescendo ao longo do tempo, sobretudo devido à sua capacidade de aplicação no estudo de diversas questões científicas importantes para as quais os métodos estatísticos mais tradicionais podem não ser adequados. Alguns exemplos da aplicabilidade dessa metodologia envolvem estudos acerca de temas como inteligência, características psicológicas de indivíduos e ainda qualidade de vida, temas esses que apesar de não poderem ser quantificados diretamente, podem se manifestar através de outras variáveis observadas, estabelecendo o que é conhecido como a relação entre o construto ou variável latente e o estudo da sua relação com as variáveis observadas que os mensuram. Dentre os casos particulares de modelagem com variáveis latentes incluem-se a análise fatorial confirmatória (AFC), modelos de equações estruturais (SEM, *Structural Equation Modeling*, em inglês), a análise de classes latentes (LCA, *Latent Class Analysis*, em inglês) e a análise de perfis latentes (LPA, *Latent Profile Analysis*, em inglês).

De acordo com a natureza dos indicadores estudados, algumas metodologias se fazem mais adequadas. No caso de variáveis latentes contínuas, as técnicas mais usadas incluem a análise fatorial confirmatória (AFC) e a modelagem com equações estruturais (SEM), que estudam e quantificam as relações entre variáveis indicadoras e variáveis latentes, ambas contínuas. No contexto de variáveis latentes categóricas as técnicas mais usadas incluem a análise de perfil latente (LPA) e a análise de classes latentes (LCA), metodologias essas que diferem no tipo de variáveis indicadoras, a primeira considerando indicadores contínuos e a segunda apenas indicadores categóricos. As análises de classes e perfis latentes são usadas para identificar subgrupos, tipos ou categorias de indivíduos de uma população em estudo, segundo Collins e Lanza (2013). A LCA permite identificar padrões de resposta com base em características observadas, relacionando-as a um conjunto de classes latentes, enquanto a LPA, também conhecida como modelo de mistura Gaussiano (*Gaussian mixture model*, em inglês) (Gibson, 1959), classifica os indivíduos nas classes da variável latente categórica a partir das respostas de cada indivíduo para as variáveis indicadoras contínuas do modelo.

Neste relatório essas metodologias para modelagem com variáveis latentes são sumarizadas. Diferentes métodos de estimação para os modelos de equações estruturais são aplicados para análise de duas bases de dados, sendo feita uma breve discussão acerca das estimativas obtidas, com ênfase em sua validade e interpretabilidade. Os resultados da aplicação envolvendo modelos de equações estruturais também são comparados com os obtidos no *software* STATA (StataCorp., 2017), que incorpora em SEM métodos de estimação estendidos dos modelos lineares generalizados (MLG). Para os métodos envolvendo variáveis latentes categóricas (LCA e LPA) são utilizadas outras duas bases de

dados, visando ilustrar a aplicabilidade das duas metodologias e também suas principais diferenças. Existem diversos *softwares* capazes de realizar a implementação dos métodos citados nesse trabalho, como MPLUS e STATA, contudo foi dado enfoque ao uso do R pelo mesmo ser gratuito, facilitando a reprodutibilidade das análises. As análises de dados são realizadas usando os pacotes *lavaan*, *poLCA* e *tidyLPA*, disponíveis no *software* R (R Core Team, 2019), com apresentação detalhada da sintaxe a ser adotada para cada metodologia.

3 MODELOS DE EQUAÇÕES ESTRUTURAIS

No contexto de SEM as múltiplas relações entre variáveis observadas e construtos são expressas matematicamente na forma de dois submodelos que incorporam matricialmente essas relações. O sistema de equações estruturais para SEM pode ser representado da seguinte forma:

$$\eta = B\eta + \Gamma\xi + \zeta \tag{3.1}$$

com as incógnitas presentes na equação definidas da seguinte forma: η representa um vetor $m \times 1$ de variáveis latentes endógenas, ξ representa um vetor $k \times 1$ de variáveis latentes exógenas, B é uma matriz $m \times m$ de coeficientes relacionando as variáveis latentes endógenas entre si, Γ é uma matriz $m \times k$ de coeficientes relacionando variáveis endógenas a variáveis exógenas e ζ é um vetor $m \times 1$ de ruídos. Usando-se essa notação a matriz B apresenta zeros em sua diagonal principal (Kaplan, 2000). Assim uma variável latente endógena não estaria relacionada com ela própria matematicamente.

As variáveis latentes são relacionadas com as observadas por meio do modelo de mensuração, que pode ser definido por duas equações, uma para as variáveis endógenas (3.2) e outra para as exógenas (3.3):

$$y = \Lambda_y\eta + \epsilon \tag{3.2}$$

e

$$x = \Lambda_x\xi + \delta \tag{3.3}$$

onde Λ_y e Λ_x representam as matrizes $p \times m$ e $q \times k$, respectivamente, de cargas fatoriais, ϵ e δ são vetores de dimensões $p \times 1$ e $q \times 1$, respectivamente, contendo os erros de mensuração em y e x.

Usualmente para fins de definir a escala da variável latente a primeira coluna das matrizes Λ_x e Λ_y são fixadas com valor 1. Alternativamente isso pode ser feito fixando-se com valor 1 a diagonal da matriz de variância-covariância Φ das variáveis latentes exógenas.

Nesse modelo assume-se que os erros de mensuração ϵ e δ têm esperança zero, cada um com distribuição normal multivariada, independentes entre si, e independentes das variáveis exógenas latentes (ξ), das variáveis endógenas latentes (η) e dos ruídos (ζ). Além disso, assume-se que as observações são amostradas independentemente e que as variáveis

exógenas latentes (ξ) têm distribuição normal multivariada. Os ruídos estruturais (ζ) têm esperança zero, distribuição normal multivariada e são independentes das variáveis exógenas latentes (ξ). Sob essas suposições, os indicadores observados x e y têm uma distribuição normal multivariada (Amorim et al., 2012), tal que

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim N_{p+q}(0, \Sigma(\Omega))$$

com $\Sigma(\Omega)$ representando a matriz de covariância populacional dos indicadores. Essa matriz é função dos parâmetros do modelo $\Omega = (B, \Gamma, \Lambda_x, \Lambda_y, \Psi, \Theta_\delta, \Theta_\epsilon, \Phi)$, podendo ser expressa por:

$$\Sigma(\Omega) = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}$$

$$\Sigma(\Omega) = \begin{pmatrix} \Lambda_y(I-B)^{-1}(\Gamma\Phi\Gamma' + \Psi)[(I-B)^{-1}]'\Lambda_y' + \Theta_\epsilon & \Lambda_y(I-B)^{-1}\Gamma\Phi\Lambda_x' \\ \Lambda_y(I-B)^{-1}\Gamma\Phi\Lambda_x' & \Lambda_x\Phi\Lambda_x' + \Theta_\delta \end{pmatrix}$$

sendo Φ a matriz de covariância $k \times k$ das variáveis exógenas latentes, Ψ é a matriz de covariância $m \times m$ de termos de ruído, e Θ_ϵ e Θ_δ são as matrizes de covariância dos erros de mensuração ϵ e δ, respectivamente.

3.1 Estimação dos parâmetros do modelo

Existem diversas formas de realizar a estimação dos parâmetros do vetor Ω, sendo a estimação por máxima verossimilhança (MV) e usando métodos de mínimos quadrados as mais usuais.

Em SEM os métodos de estimação consistem na obtenção, via algoritmos iterativos, das estimativas para o vetor Ω, denominadas como $\hat{\Omega}$, visando minimizar a função de discrepância $F(S, \hat{\Sigma})$, que é um escalar mensurando a distância entre a matriz de covariância amostral (S) e a matriz de covariância ajustada $\Sigma(\hat{\Omega})$ (Kaplan, 2000).

3.1.1 Estimação via máxima verossimilhança

Assumindo a hipótese de normalidade multivariada e independência para as $n = N - 1$ observações, onde a constante N representa o número total de observações, a função de ajuste por MV será dada pela seguinte equação:

$$F_{ML} = \log|\Sigma(\Omega)| + tr(S\Sigma(\Omega)^{-1}) - \log|S| - p$$

onde p é o número de variáveis observadas.

3.1.2 Implementação computacional para SEM com método MV

Por meio do *software* R existem alguns pacotes que permitem a implementação desse método de estimação, sendo os mais conhecidos o ***lavaan*** e o ***sem***. Para ilustrar a aplicação computacional será utilizado o pacote *lavaan*. Se todos os dados são contínuos, o estimador padrão do pacote é o estimador de máxima verossimilhança, que pode ser definido através da opção: ***(estimator = "ML")***.

Para esse método de estimação o *lavaan* oferece também algumas variantes denominadas "robustas", que, segundo o pacote, fornecem erros padronizados robustos à não normalidade multivariada dos dados e estatísticas de teste escalonadas. A seguir a definição de cada uma das opções de estimação com suas correspondentes sintaxes são listadas:

- **ML**: Método tradicional baseado na suposição de normalidade multivariada.

```
modelo=sem(mee,data=dados,estimator="ML")
```

- **MLM**: Estimação de máxima verossimilhança com erros padronizados robustos à não normalidade e estatística de teste escalonada Satorra-Bentler.

```
modelo=sem(mee,data=dados,estimator="MLM")
```

- **MLMVS**: Estimação de máxima verossimilhança com erros padronizados robustos à não normalidade e estatística de teste ajustada por média e variância (também conhecida como a abordagem Satterthwaite).

```
modelo=sem(mee,data=dados,estimator="MLMVS")
```

- **MLR**: Estimação de máxima verossimilhança com erros padronizados robustos à não normalidade (Huber-White) e uma estatística de teste escalonada que é assintoticamente igual à estatística do teste de Yuan-Bentler. Pode ser utilizada para dados completos ou incompletos.

```
modelo=sem(mee,data=dados,estimator="MLR")
```

Mais variantes desse método podem ser encontradas em Rosseel e Yves (2018).

Nesse relatório o objeto **model** contém a sintaxe definida pelo usuário para especificar o modelo sob investigação, e **dados** denomina a base de dados do usuário. Exemplo da especificação de objeto **model** é apresentado no capítulo de Resultados, na seção contendo aplicações com modelos de equações estruturais.

3.1.3 Estimação via métodos de mínimos quadrados generalizados

O estimador de mínimos quadrados generalizados (GLS, em inglês) assume que os dados são provenientes de uma distribuição normal multivariada. A forma geral da função de discrepância ajustada pelo GLS é dada por:

$$F_{GLS} = [S - \Sigma(\Omega)]' W^{-1} [S - \Sigma(\Omega)]$$

em que W^{-1} é uma matriz de pesos que pondera os desvios $(S - \Sigma(\Omega))$ em termos de suas variâncias e covariâncias com outros elementos.

As duas escolhas mais comuns para W^{-1} são a matriz identidade, $W^{-1} = I$ e a matriz de covariância amostral $W^{-1} = S^{-1}$. Quando $W^{-1} = S^{-1}$, então a função de discrepância ajustada pelo GLS pode ser definida por:

$$F_{GLS} = \frac{1}{2} tr[S^{-1}(S - \Sigma(\Omega))]^2$$

$$F_{GLS} = \frac{1}{2} tr(I - S^{-1}\Sigma(\Omega))^2$$

Sob a suposição de normalidade multivariada, o estimador GLS é assintoticamente eficiente e normal (Amorim et al, 2012).

3.1.4 Implementação computacional para SEM com método MMQ

De forma semelhante ao método anterior a utilização do GLS via R é feita também pela opção *estimator*, definida da seguinte forma: $(\boldsymbol{estimator = "GLS"})$.

Para esse tipo de estimação pode-se utilizar as seguintes sintaxes:

- **GLS**: Mínimos quadrados generalizados.

```
modelo=sem(mee,data=dados,estimator="GLS")
```

- **WLS**: Mínimos quadrados ponderados.

  O ajuste via método de mínimos quadrados ponderados dispensa o pressuposto de pertencimento dos dados à uma distribuição de probabilidade, sendo também conhecido como estimador ADF.

```
modelo=sem(mee,data=dados,estimator="WLS")
```

- **DWLS**: Mínimos quadrados diagonalmente ponderados.

```
modelo=sem(mee,data=dados,estimator="DWLS")
```

- **ULS**: Mínimos quadrados não ponderados.

```
modelo=sem(mee,data=dados,estimator="ULS")
```

4 ANÁLISE DE CLASSES LATENTES

A análise de classes latentes (LCA, em inglês) é uma técnica usada para a modelagem com variáveis latentes e indicadores categóricos, pela qual os indivíduos estudados são divididos em subgrupos, também chamados classes latentes, mediante aos seus padrões de resposta. O modelo é ajustado com base em indicadores categóricos, não sendo necessário pressupor a distribuição dos mesmos. O modelo também supõe independência local, ou seja, numa mesma classe latente os indicadores devem ser independentes.

No contexto de LCA o objetivo principal é identificar uma sucessão de subgrupos ou classes latentes que são mutualmente exclusivas. O número de classes pode ser definido pela comparação de estatísticas de ajuste, afim de se encontrar o modelo que se adequa melhor aos dados. Para que seja feita a definição do modelo, será feita uma breve introdução sobre as notações utilizadas nesse modelo:

- j: representa as variáveis observadas, $j = 1, \cdots, J$.
- r_j: são as categorias de resposta possíveis para cada variável, $r_j = 1, \cdots, R_j$.
- C: representa o número total de classes latentes.

Assim a tabela de contingência formada por tabulação cruzada das J variáveis tem:

$$W = \prod_{j=1}^{J} R_j \qquad \text{células.}$$

Há um padrão de resposta correspondente a cada uma das W células, que pode ser expresso da seguinte forma:

- $y = (r_1, r_2, ..., r_j)$: é o vetor particular de resposta das J variáveis.
- Y: é a matriz referente aos padrões de resposta, com W linhas e J colunas.

Para cada padrão de resposta é associada uma probabilidade de ocorrência, denotada por $P(Y = y)$, tal que: $\sum P(Y = y) = 1$.

Este tipo de modelo tem dois parâmetros fundamentais (Collins e Lanza, 2013), que são:

$\boldsymbol{\gamma_c}$: As probabilidades não condicionais ou prevalências da classe, que representam as probabilidades dos indivíduos pertencerem à c-ésima classe latente. Como mencionado acima, as classes latentes são mutuamente exclusivas e exaustivas; em outras

palavras, cada indivíduo é membro de uma e apenas uma classe latente. Assim sendo: $\sum_{c=1}^{C} \gamma_c = 1$.

$\boldsymbol{\rho}$: As probabilidades condicionais, que representam as probabilidades da resposta r_j condicionadas ao pertencimento à c-ésima classe latente. Assim $\rho_{j,r_j|c}$ representa a probabilidade da resposta r_j para a variável observada j, condicional ao indivíduo pertencer à classe latente c. Como cada indivíduo fornece uma e apenas uma resposta alternativa à variável j, o vetor de probabilidades item-resposta para uma determinada variável condicional em uma determinada classe latente sempre soma 1, ou seja, $\sum_{r_j=1}^{R_j} \rho_{j,r_j|c} = 1$.

A equação seguinte expressa como a probabilidade de se observar um vetor de resposta específico é uma função de ρ e γ:

$$P(Y = y) = \sum_{c=1}^{C} \gamma_c \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)} \tag{4.1}$$

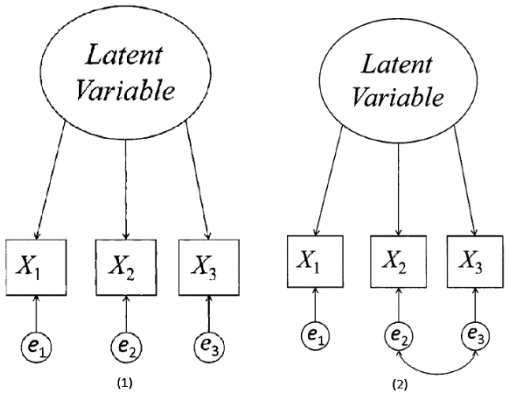
Para se obter a probabilidade de se obter um padrão de resposta específico condicional a uma classe c tem-se:

$$P(Y = y|C = c) = \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)} \tag{4.2}$$

onde y_j representa o j-ésimo elemento de um padrão de resposta y e $I(y_j = r_j)$ é uma função indicadora com valor 1 quando a resposta para a variável $j = r_j$, e 0 caso contrário.

Não há pressupostos de que os indicadores das classes latentes ou as classes latentes em si estão em qualquer nível de medida diferente do nominal. Como os indicadores são categóricos, sua distribuição conjunta é multinomial. Pressupostos estritos de distribuição, como a normalidade multivariada, são desnecessários (Collins e Lanza, 2013). No entanto, um pressuposto importante é o da independência local, que é representada graficamente na Figura 1. Na Figura 1.1 há setas conectando as variáveis observadas X_1, X_2 e X_3 à variável latente, mas nenhuma outra seta conectando entre si os componentes das variáveis observadas. Isso significa que as três variáveis observadas estão relacionadas apenas pela variável latente (Collins e Lanza, 2013).

Figura 1 – Diagrama representando a independência local



Fonte: Adaptado (Collins e Lanza, 2013)

A Figura 1.2, por sua vez, ilustra uma violação da independência local. As variáveis observadas X_2 e X_3 estão relacionadas entre si, não só através da variável latente, mas também através dos seus componentes de erro. As implicações da hipótese de independência local podem ser vistas na Equação 4.2, pois de acordo com as leis da probabilidade, a probabilidade de se observar um vetor de respostas específico y condicional a uma classe latente c pode ser encontrado apenas multiplicando-se os parâmetros individuais correspondentes a uma classe latente particular. Sem essa suposição, a Equação 4.2 teria que ser muito mais complicada porque as respostas deveriam ser condicionadas não só na adesão à classe latente, mas uma à outra (Collins e Lanza, 2013).

Há um outro tipo de probabilidade de grande interesse ao se estudar os modelos de classe latentes, denominadas probabilidades posteriores ou probabilidades de classificação, que permitem descrever o pertencimento a uma determinada classe latente c dado o padrão de resposta apresentado pelo indivíduo, sendo definida como:

$$P(C = c|Y = y)$$

Em geral as probabilidades posteriores não podem ser calculadas diretamente. Assim, uma forma para o cálculo das mesmas é a utilização do teorema de Bayes:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Fazendo $P(A) = P(L = c)$ e $P(B) = P(Y = y)$ temos a seguinte equação:

$$P(C = c|Y = y) = \frac{P(Y = y|C = c) \times P(C = c)}{P(Y = y)}$$

As estimativas das prevalências das classes latentes e das probabilidades de resposta ao item em uma LCA fornecem os elementos necessários para obter a probabilidade a posteriori de adesão (Collins e Lanza, 2013). Substituindo os valores de $P(Y = y)$ e

$P(C = c)$ na equação acima, teremos:

$$P(C = c|Y = y) = \frac{\prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)} \times \gamma_c}{\sum_{c=1}^{C} \gamma_c \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)}} \tag{4.3}$$

Um grau elevado de certeza de classificação ocorre quando são obtidos altos valores para a probabilidade de classificação para cada indivíduo. Assim o mesmo terá uma grande probabilidade de pertencer a uma classe latente e, consequentemente, baixas probabilidades de pertencer às demais classes.

Definição do modelo com covariáveis

O modelo de classes latentes com covariáveis, também conhecido como modelo de regressão de classe latente, generaliza o modelo básico de classe latente ao permitir a inclusão de covariáveis que têm efeito sobre as prevalências de cada uma das classes. Segundo Collins e Lanza (2013), as covariáveis são incorporadas na LCA usando uma estrutura de regressão logística, e como em qualquer estrutura de regressão, o conjunto de covariáveis pode incluir variáveis categóricas, variáveis quantitativas, ou uma combinação de ambas.

Considerando uma covariável X, o novo modelo pode ser expresso da seguinte forma:

$$P(Y = y|X = x) = \sum_{c=1}^{C} \gamma_c(x) \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} \rho_{j,r_j|c}^{I(y_j=r_j)} \tag{4.4}$$

que é um modelo bastante semelhante ao definido na Equação (4.2), contudo com a presença do termo $\gamma_c(x)$, que para o caso de uma única covariável X pode ser definido como:

$$\gamma_c(x) = P(L = c|X = x) = \frac{e^{\beta_{0c}+\beta_{1c}x}}{1 + \sum_{c'=1}^{C-1} e^{\beta_{0c}+\beta_{1c}x}} \qquad \text{para} \quad c' = 1, \cdots, (C-1)$$

(4.5)

A regressão logística requer considerar uma das classes da variável latente como referência. As probabilidades de resposta ao item ainda são estimadas, mas não as prevalências de classe latente. Em vez das prevalências de classe latente, os coeficientes estimados são as regressões (β's) e as prevalências de classe latente podem ser expressas como funções dos coeficientes de regressão e dos valores individuais das covariáveis correspondentes (Collins e Lanza, 2013).

4.1 Estimação no modelo de classes latentes

No contexto de modelos de classes latentes, as estimativas são obtidas através da maximização da função de log-verossimilhança para os dados via algoritmos iterativos, pois não há forma analítica exata para obtenção das estimativas dos parâmetros. Nesse contexto a função de log-verossimilhança assume a seguinte forma:

$$ln(L) = \sum_{i}^{N} ln \sum_{c=1}^{C} \gamma_c \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} (\rho_{j,r_j|c})^{I(y_j=r_j)}$$

e é maximizada em relação aos parâmetros γ_c e $\rho_{j,r_j|c}$, usando o algoritmo (EM) proposto por Dempster, Laird e Rubin (1977). O algoritmo EM prossegue iterativamente, começando com valores iniciais arbitrários de $\hat{\gamma}_c$ e $\hat{\rho}_{j,r_j|c}$ que são nomeados como $\hat{\gamma}_c^{old}$ e $\hat{\rho}_{j,r_j|c}^{old}$. Na etapa de esperança são calculadas as probabilidades de associação à classe usando a Equação (4.4), substituindo em $\hat{\gamma}_c^{old}$ e $\hat{\rho}_{j,r_j|c}^{old}$. Na etapa de maximização, atualize as estimativas dos parâmetros maximizando a função de log-verossimilhança, dada as probabilidades posteriores $\hat{P}(C = c|Y = y)$, com:

$$\hat{\gamma}_c^{new} = \frac{1}{N} \sum_{i=1}^{N} \hat{P}(C = c|Y = y) \tag{4.6}$$

como as novas prevalências e

$$\hat{\rho}_{jc}^{new} = \frac{\sum_{i=1}^{N} Y_{ij} \hat{P}(L = c|Y = y)}{\sum_{i=1}^{N} \hat{P}(L = c|Y = y)} \tag{4.7}$$

definindo as novas probabilidades de resposta condicionadas à classe, para mais detalhes ver Everitt e Hand (1981) e Everitt (1984). $\hat{\rho}_{jc}^{new}$ é o vetor de comprimento R_j de probabilidades de resposta condicionais a uma classe c para a j-ésima variável observada; e Y_{ij} é uma matriz de dimensão $N \times R_j$ de resultados observados $I(y_j = r_j)$ nessa variável. O algoritmo repete essas etapas, atualizando as estimativas antigas pelas novas, até que log-verossimilhança seja maximizado e deixe de aumentar além de algum valor arbitrariamente pequeno (Linzer e Lewis, 2011).

Implementação computacional para LCA

Por meio do *software* R existem alguns pacotes que permitem a implementação desse método de estimação, sendo *poLCA* o pacote mais conhecido, que é utilizado para ilustrar a implementação computacional. É importante ressaltar que o *poLCA* só aceita valores inteiros positivos para as categorias de resposta das variáveis manifestas, ou seja, em variáveis que possuem categorias de resposta codificadas como 0, valores negativos ou decimais, o pacote retorna uma mensagem de erro e não realiza a estimação.

No *software* R o pacote necessário para a implementação do modelo de classes latentes pode ser instalado e carregado da seguinte forma:

1. Instalação

```
#Instalando pacote
install.packages("poLCA")
```

2. Carregamento

```
#Carregando pacote
require(poLCA)
```

A seguir são apresentadas as sintaxes padrões utilizadas pelo pacote para especificar e estimar um modelo de classes latentes:

1. Especificação:

```
#Especificando o conjunto de variáveis a ser utilizado:
formula=cbind(X1, X2,...,Xn)~1
```

2. Estimação

```
#Estimando modelo de classes latentes:
modelo=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 opções listadas acima para a estimação do modelo de classes latentes, via pacote *poLCA*, são definidas da seguinte forma: (1)**data**: corresponde ao conjunto de dados; (2)**nclass**: número de classes latentes; (3)**maxiter**: número máximo de iterações usado pelo algoritmo de estimação; (4)**graphs**: se TRUE exibe gráficos para as estimativas dos parâmetros; (5)**tol**: valor de tolerância para julgar quando a convergência é atingida; (6)**na.rm**: se TRUE remove linhas onde há observações faltantes; (7)**probs.start**: lista de matrizes de probabilidades condicionais de resposta que podem servir como os valores iniciais para o algoritmo de estimação EM; (8)**nrep**: número de vezes que o modelo é estimado, usando valores diferentes de probs.start, com o *poLCA* retornando apenas as estimativas correspondentes ao modelo com a maior log-verossimilhança; (9)**verbose**: se TRUE indica que o *poLCA* deve produzir na tela os resultados do modelo; (10)**calc.se**: se TRUE indica que o *poLCA* deve calcular os erros padrões das estimativas das probabilidades condicionais de resposta e das prevalências de classe latente.

Ordem das classes

Como as classes latentes são categorias não ordenadas, a ordem numérica das classes latentes estimadas pelo modelo é arbitrária e é determinada unicamente pelos valores iniciais do algoritmo EM. Se **probs.start** for definido como NULL (o padrão) ao utilizar o *poLCA*, a função gerará os valores iniciais aleatoriamente em cada execução (Linzer e Lewis, 2011). Dessa forma execuções repetidas para o mesmo modelo, considerando o mesmo conjunto de dados, podem fornecer as mesmas estimativas contudo a ordem das classes latentes pode vir alterada, e para fins interpretativos é importante se ater às possíveis mudanças na ordem das classes antes de qualquer conclusão acerca das estimativas do modelo. Na maioria dos casos as classes latentes são rotuladas segundo suas probabilidades condicionais, e caso deseje-se reestimar o modelo mantendo a mesma ordem observada é possível tanto a fixação da ordem das classes quanto fixar os valores iniciais utilizados pelo algoritmo, fazendo com que, independentemente da estimação, a ordem das classes se mantenha assim como seu significado.

Para alterar a ordem das classes é utilizada à função ***poLCA*.reorder**. Assumindo um modelo hipotético com 3 classes latentes a função pode ser utilizada da seguinte forma:

```
#Definindo um conjunto hipotético de variáveis
formula=cbind(X1,X2,X3,X4)~1

#Estimando o modelo
modelo=poLCA(formula,data,nclass=3)

#Atribuindo valores iniciais a um objeto
probs.start <- modelo$probs.start

#Definindo novos valores iniciais
new.probs.start <- poLCA.reorder(probs.start, c(1, 3, 2))

#Reestimando o modelo com os novos valores iniciais
modelo <- poLCA(formula, data, nclass = 3, probs.start = new.probs.start)
```

Feito isto os rótulos das classes latentes 3 e 2 foram alterados, ou seja, na saída do *software* R a ordem das probabilidades condicionais será trocada, e a classe latente que antes era rotulada como 2 passa a ser rotulada como 3. Pode-se ainda fixar os valores iniciais obtendo um modelo com as classes na mesma ordem e com probabilidades condicionais inalteradas:

```
#Definindo um conjunto hipotético de variáveis
formula=cbind(X1,X2,X3,X4)~1

#Estimando o modelo
modelo=poLCA(formula,data,nclass=3)

#Atribuindo valores iniciais a um objeto
probs.start <- modelo$probs.start

#Reestimando o modelo com os mesmos valores iniciais do modelo anterior
modelo <- poLCA(formula, data, nclass = 3, probs.start = probs.start )
```

Neste caso o novo modelo apresentará ainda as 3 classes latentes, que manterão os mesmos rótulos que a estimação anterior e consequentemente seus significados.

Também é possível fixar as classes latentes de um modelo para que elas sigam a ordem decrescente de prevalências de classes, ou seja, a classe com rótulo 1 será a com maior prevalência e assim por diante. Utilizando a mesma sintaxe para o caso de um modelo hipotético com 3 classes latentes teremos:

```
#Atribuindo valores iniciais a um objeto que ordena
#classes por ordem decrescente de suas prevalências
probs.start <- poLCA.reorder(modelo$probs.start,
order(modelo$P, decreasing = TRUE))


#Nova estimativa para o modelo
modelo <- poLCA(formula, data ,nclass=3,probs.start = probs.start)
```

4.1.1 Estimação no modelo de classes latentes com covariáveis

O modelo de classes latentes com covariáveis tem uma função de log-verossimilhança quase idêntica à do modelo padrão, exceto pela função $\gamma_c(x)$, que substitui as prevalências das classes latentes. A nova função de log-verossimilhança pode ser expressa da seguinte forma:

$$ln(L) = \sum_{i}^{N} ln \sum_{c=1}^{C} \gamma_c(x) \prod_{j=1}^{J} \prod_{r_j=1}^{R_j} (\rho_{j,r_j|c})^{I(y_j=r_j)}$$

Os parâmetros estimados pelo modelo de classe latente com covariáveis são os $(R-1)$ vetores de coeficientes β_r e, como no modelo básico de classe latente, as probabilidades

condicionais de resposta condicionais à classe ρ_{jrk}. Dadas as estimativas $\hat{\beta}_r$ e $\hat{\rho}_{jrk}$ desses parâmetros, as probabilidades posteriores de associação de classe no modelo com covariáveis são obtidas substituindo γ_r na Equação (4.3) pela função $\gamma_c(x)$ da seguinte forma:

$$\hat{P}(C = c|Y = y, X = x) = \frac{\prod_{j=1}^{J}\prod_{r_j=1}^{R_j}\rho_{j,r_j|c}^{I(y_j=r_j)} \times \gamma_c(x)}{\sum_{c=1}^{C}\gamma_c(x)\prod_{j=1}^{J}\prod_{r_j=1}^{R_j}\rho_{j,r_j|c}^{I(y_j=r_j)}} \tag{4.8}$$

para encontrar os valores de $\hat{\beta}_r$ e $\hat{\rho}_{jrk}$ que maximizam a função acima, o *poLCA* usa também o algorítimo EM. O proceso de estimação começa com valores iniciais para $\hat{\beta}_r^{old}$ e ρ_{jrk}^{old} que serão usados para calcular as probabilidades posteriores utilizando a Equação (4.8). Os coeficientes das correspondentes covariáveis são atualizados de acordo com a fórmula:

$$\hat{\beta}_r^{new} = \hat{\beta}_r^{old} - H_\beta^{-1}\nabla_\beta \tag{4.9}$$

onde ∇_β é o gradiente e H_β é a matriz hessiana da função de log-verossimilhança com respeito a β. As probabilidades $\hat{\rho}_{jrk}^{new}$ são atualizadas da seguinte forma:

$$\hat{\rho}_{jr}^{new} = \frac{\sum_{i=1}^{N} Y_{ij}\hat{P}(L = c|Y = y, X = x)}{\sum_{i=1}^{N}\hat{P}(L = c|Y = y, X = x)} \tag{4.10}$$

Essas etapas são repetidas até a convergência, atualizando as estimativas dos parâmetros em cada iteração. As fórmulas para o gradiente e matriz Hessiana são fornecidas em Bandeen-Roche et al (1997).

Implementação computacional para LCA com covariáveis

O pacote *poLCA* do *software* R é utilizado novamente para a implementação computacional desta metodologia. A seguir são apresentadas as sintaxes padrões utilizadas pelo pacote para especificar e estimar um modelo de classes latentes com covariáveis:

1. Especificação:

```
#Especificando o conjunto de variaveis e covariável
a ser utilizado:
formula=cbind(X1, X2,...,Xn)~W1
```

Aqui considerou-se a inclusão da covariável W_1 após o til, diferentemente da sintaxe de especificação do modelo padrão. Para o uso de mais covariáveis, tem-se a seguinte sintaxe:

```
#Especificando o conjunto de variaveis e covariáveis a serem
utilizadas:
formula=cbind(X1, X2,...,Xn)~W1+W2
```

Se for de interesse considerar a interação entre duas covariáveis, pode ser utilizada a seguinte sintaxe:

```
#Especificando o conjunto de variaveis e covariáveis a serem
utilizadas:
formula=cbind(X1, X2,...,Xn)~W1*W2
```

2. Estimação:

   Para a estimação não há alterações de sintaxe se comparada à do modelo sem covariáveis.

5 ANÁLISE DE PERFIS LATENTES

A análise de perfis latentes (LPA, em inglês) é um caso de modelo com variáveis latentes categóricas, mas que, diferentemente da LCA (com indicadores categóricos), admite a incorporação de indicadores contínuos. Ao contrário da LCA, a LPA não necessita do pressuposto de independência local dos indicadores uma vez que as variâncias e covariâncias dos mesmos são incorporadas via matrizes de variância-covariância dos indicadores contínuos. O objetivo da LPA é definir um número de subgrupos ou perfis latentes que são mutuamente exclusivas para que possam ser estudadas as medidas dos indicadores dos indivíduos classificados em cada um dos perfis latentes, visando entender o comportamento dos indivíduos dispostos em cada perfil.

Sejam $\gamma_1, \gamma_2, ..., \gamma_M$ indicadores contínuos relativos a $i = 1, ..., n$ indivíduos, e sejam $k = 1, ..., K$ perfis da variável latente denotada por C, onde $c_i = k$ se o i-ésimo indivíduo pertencer ao perfil k, π_k a proporção de indivíduos em cada perfil latente com $\sum \pi_k = 1$. As relações entre as respostas dos indicadores para cada indivíduo e a variável latente podem ser expressas a partir do seguinte modelo:

$$f(y_i) = \sum_{k=1}^{K} [\pi_k f_k(y_i)]$$

onde $y_i = (\gamma_{i1}, ..., \gamma_{Mi})$ corresponde ao vetor de respostas de cada indivíduo para os M indicadores contínuos, $f(y_i)$ é a função densidade de probabilidade multivariada para toda população e $f_k(y_i) = f(y_i|c_i = k)$ é a função densidade específica para o perfil k. Dessa forma o modelo de LPA especifica a distribuição conjunta dos indicadores contínuos através de uma mistura das distribuições específicas para cada um dos k perfis latentes.

Segundo Masyn e Little (2013, p.584) nas primeiras aplicações de modelos de mistura finita, que são uma classe mais geral de modelos ao qual a LPA faz parte, a distribuição das variáveis contínuas dentro dos perfis latentes é considerada normal multivariada, denotada da seguinte forma:

$$[y_i|c_i = k] \sim MVN(\alpha_k, \Sigma_k)$$

onde α_k é o vetor de médias para os indicadores dos indivíduos pertencentes ao perfil latente k, e Σ_k a matriz de variância-covariância dos indicadores dos indivíduos pertencentes ao perfil latente k. Desta forma, os parâmetros do modelo são as médias específicas por classe, variâncias e covariâncias das variáveis indicadoras. Como o pressuposto do modelo é de que as distribuições específicas dentro dos perfis latentes sejam normal multivariada com matriz de variância-covariância Σ_k, não se faz necessário o pressuposto de independência local dos indicadores já que a matriz Σ_k incorpora as relações entre indicadores, presentes na Figura 1.2.

5.1 Restrições para matriz Σ_k

Em relação as matrizes Σ_k, durante a construção do modelo podem ser adotadas algumas restrições que podem impactar nos resultados obtidos pelo método. Segundo Masyn e Little (2013, p.585-586), as restrições mais usuais consideradas para a matriz Σ_k, presumindo que não haverá restrições para α_k dentro e através dos perfis em todos os casos, são:

1. Variâncias iguais e covariâncias fixadas em 0.

$$\Sigma_k = \begin{bmatrix} \theta_{11} & & & \\ 0 & \theta_{22} & & \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & \theta_{MM} \end{bmatrix}$$

   Nessa estrutura as variâncias são estimadas como iguais entre os perfis, indicadas pela ausência de um subscrito para qualquer um dos elementos diagonais da matriz, enquanto as covariâncias são restritas a zero. Segundo Rosenberg (2019), esse modelo é altamente restrito, mas também parcimonioso, já que os perfis são estimados de forma que as variâncias dos indicadores não variem entre os perfis e as relações entre as variáveis não sejam estimadas. Desta forma, menos graus de liberdade são usados para explicar as observações que compõem os dados.

2. Variâncias diferentes por perfis e covariâncias fixadas em 0.

$$\Sigma_k = \begin{bmatrix} \theta_{11k} & & & \\ 0 & \theta_{22k} & & \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & \theta_{MMk} \end{bmatrix}$$

   Nessa estrutura a covariância entre as variáveis indicadoras é fixada como 0 e as variâncias entre as mesmas variáveis assumem diferentes valores em relação aos perfis latentes. Segundo Rosenberg (2019), esse modelo é mais flexível e menos parcimonioso que o modelo 1. Esse modelo é definido como tendo parametrização diagonal de classe variável.

3. Variâncias e covariâncias iguais por perfil.

$$\Sigma_k = \Sigma = \begin{bmatrix} \theta_{11} & & & \\ \theta_{21} & \theta_{22} & & \\ \vdots & \vdots & \ddots & \\ \theta_{M1} & \theta_{M2} & \dots & \theta_{MM} \end{bmatrix}$$

Nessa estrutura são permitidas covariâncias entre as variáveis indicadoras, contudo a medida não muda entre os perfis. Este modelo também é referido como parametrização irrestrita invariável à classe.

4. Variâncias diferentes e covariâncias iguais, por perfis.

$$\Sigma_k = \begin{bmatrix} \theta_{11k} & & & \\ \theta_{21} & \theta_{22k} & & \\ \vdots & \vdots & \ddots & \\ \theta_{M1} & \theta_{M2} & \dots & \theta_{MMk} \end{bmatrix}$$

Esse modelo especifica que as variâncias sejam estimadas com valores diferentes entre os perfis, e as covariâncias sejam estimadas como iguais entre perfis. Este modelo não pode ser implementado através do pacote mclust do *software* R, contudo sua implementação é possível através do *software* MPLUS, com as funções de interface para MPLUS descritas na seção 5.3.

5. Variâncias iguais e covariâncias diferentes, por perfis.

$$\Sigma_k = \begin{bmatrix} \theta_{11} & & & \\ \theta_{21k} & \theta_{22} & & \\ \vdots & \vdots & \ddots & \\ \theta_{M1k} & \theta_{M2k} & \dots & \theta_{MM} \end{bmatrix}$$

Esse modelo especifica que as variâncias sejam estimadas com valores iguais entre os perfis, já as covariâncias são estimadas com valores diferentes entre perfis. Assim como o modelo do item 4, este modelo não pode ser implementado através do pacote mclust do *software* R, contudo sua implementação é possível através do *software* MPLUS, com as funções de interface para MPLUS descritas a seguir.

6. Variâncias e covariâncias diferentes, por perfis.

$$\Sigma_k = \begin{bmatrix} \theta_{11k} & & & \\ \theta_{21k} & \theta_{22k} & & \\ \vdots & \vdots & \ddots & \\ \theta_{M1k} & \theta_{M2k} & \dots & \theta_{MMk} \end{bmatrix}$$

Neste caso θ_{mmk} representa a variância do indicador m dentro do perfil latente k, e θ_{mjk} representa a covariância entre os indicadores m e j dentro da mesmo perfil. Nessa estrutura são permitidas covariâncias entre as variáveis indicadoras dentro

dos perfis, contudo as variâncias e covariâncias para as mesmas variáveis variam de perfil para perfil.

As matrizes Σ_k com estrutura diagonal assumem o pressuposto de independência local. Nestes casos, as covariâncias entre os indicadores são fixas em zero dentro dos perfis, enquanto as variâncias são restritas para variar ou não entre os perfis latentes. Segundo Masyn e Little (2013, p.586), especificar uma matriz Σ_k classe-invariante diagonal irá fornecer uma solução que é equivalente a aplicar um algoritmo de agrupamento *k-means* aos indicadores de classe latente.

É importante ressaltar que a estrutura da matriz Σ_k mais adequada pode variar conforme a situação. Logo, é importante avaliar qual modelo é o mais adequado para estudar o problema seja por questão de parcimônia no número de perfis latentes ou em relação a medidas de ajuste do modelo, como AIC e BIC.

5.2 Homogeneidade e separação dos perfis

Em um modelo de LPA a homogeneidade e separação dos perfis latentes são aspectos importantes de se avaliar para obter conclusões de quão bem o modelo classifica os indivíduos entre os perfis. O primeiro aspecto a se analisar é a homogeneidade dentro dos perfis, que sobretudo está relacionada à comparação entre a variabilidade estimada de cada indicador dentro dos perfis $\hat{\theta}_{mmk}$ e a variabilidade geral estimada dos indicadores na amostra $\hat{\theta}_{mm}$. Perfis com valores menores de $\hat{\theta}_{mmk}$ são mais homogêneas em relação ao indicador m do que perfis com valores maiores de $\hat{\theta}_{mmk}$. Avaliar a homogeneidade dos perfis é importante porque deseja-se agrupar indivíduos semelhantes com base em suas respostas para as variáveis indicadoras para se concluir e interpretar o significado dos perfis.

Segundo Masyn e Little (2013, p.588), uma forma inicial para avaliar o grau de separação entre os perfis é através da distância entre as médias estimadas dos indicadores em cada um deles. Contudo, essa avaliação não seria suficiente pois o mais relevante é mensurar o grau de sobreposição entre as distribuições específicas de cada perfil. Como, por pressuposto do método, as distribuições específicas por perfil são normais multivariadas, o grau de sobreposição entre elas dependerá não somente de suas médias estimadas, mas também das variâncias das distribuições. Para quantificar a separação entre dois perfis j e k em respeito a uma variável indicadora m, pode ser calculada a seguinte diferença média padronizada:

$$\hat{d}_{mjk} = \frac{\hat{a}_{mj} - \hat{a}_{mk}}{\hat{\sigma}_{mjk}} \tag{5.1}$$

com $\hat{\sigma}_{mjk}$ sendo o seguinte desvio padrão combinado:

$$\hat{\sigma}_{mjk} = \sqrt{\frac{(\hat{\pi}_j)(\hat{\theta}_{mmj}) + (\hat{\pi}_k)(\hat{\theta}_{mmk})}{(\hat{\pi}_j + \hat{\pi}_k)}}$$

Segundo Masyn e Little (2013, p.588) valores para $|\hat{d}_{mjk}| > 2.0$ significam que há menos que 20% de sobreposição entre as distribuições, desta forma, valores grandes para $|\hat{d}_{mjk}|$ significam um alto grau de separação entre os perfis j e k em relação ao indicador m. Já valores pequenos para $|\hat{d}_{mjk}|$ como $|\hat{d}_{mjk}| < 0.85$ correspondem na existência de mais de 50% de sobreposição entre as distribuições específicas de classe, resultando em um baixo grau de separação em relação ao indicador m.

5.3 Estimação no modelo de perfis latentes

A estimação do modelo é feita através da maximização da função de log verossimilhança dos dados, denotada por $l(\phi, y)$, para $\phi = (\pi, \theta, \alpha, \Sigma)$. Contudo a maximização da função $l(\phi, y)$ é difícil de ser realizada, sendo necessária a utilização do algoritmo (EM) para obtenção das estimativas dos parâmetros do modelo que maximizam a função de log verossimilhança $l(\phi, y)$ dada por:

$$l(\phi, y) = \sum_{i=1}^{n} \log f(y_i) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k f_k(y_i) \right]$$

Segundo Masyn e Little (2013, p.590), as funções de log verossimilhança para modelos de mistura finita podem ser um desafio para os algoritmos de estimação, e por isso são utilizados vários conjuntos de valores iniciais para as estimativas dos parâmetros, evitando problemas de não convergência ou de pontos de máximo locais.

Implementação computacional para LPA

No *software* R existem alguns pacotes que permitem a implementação computacional do modelo de perfis latentes, mas a ilustração é feita usando os pacotes mclust e *tidyLPA*, este último permite além da implementação do modelo, o uso de funcionalidades para interação com o MPLUS, que também é capaz de implementar modelos LPA. No *software* R o pacote necessário para a implementação do modelo de perfis latentes pode ser instalado e carregado da seguinte forma:

1. Instalação

```
#Instalando pacote
install.packages("tidyLPA")
```

2. Carregamento

```
#Carregando pacote
require(tidyLPA)
```

Estimação

A seguir são apresentadas as sintaxes padrões utilizadas pelo pacote *tidyLPA* para especificar e estimar um modelo de perfis latentes, bem como outras funções úteis para análise descritiva sobre os resultados do modelo:

```
#Estimação de um modelo
estimate_profiles(df, n_profiles, models = NULL, variances = "equal",
                  covariances = "zero", package = "mclust")
```

As opções listadas acima para a estimação do modelo de perfis latentes, via pacote *tidyLPA*, são definidas da seguinte forma: (1)**df**: data.frame de dados numéricos; (2)**n_profiles**: vetor inteiro do número de perfis a ser estimado; (3)**models**: vetor inteiro cujo valor representa o tipo do modelo utilizado, definido como NULL por padrão. Os modelos são construídos a partir dos argumentos relacionados à matriz de variância-covariância, conforme descrito anteriormente; (4)**variances**: vetor alfanumérico, especificando quais componentes de variação estimar. O padrão é '*equal*' (variâncias iguais entre perfis), a outra opção é '*varying*' (variâncias diferentes entre perfis). Cada elemento deste vetor refere-se a um dos modelos que se deseja executar; (5)**covariances**: vetor alfanumérico, especificando quais componentes de covariância estimar. O padrão é "zero" (não estimar covariâncias), as outras opções são '*equal*' (covariâncias iguais entre perfis) e '*varying*' (covariâncias diferentes entre perfis). Cada elemento deste vetor refere-se a um dos modelos que se deseja executar; (6)**package**: especifica qual pacote usar para estimação do modelo: '*mclust*' ou '*MplusAutomation*' (requer que o MPLUS seja instalado), o padrão é 'mclust'.

Caso o usuário tenha o MPLUS instalado, é possível realizar a estimação do modelo utilizando o MPLUS via *software* R. Assim é possível que sejam utilizados os modelos com restrições para a matriz Σ_k enumerados como modelos 4 e 5 anteriormente. Para definição dos tipos de modelo utilizados (diferentes restrições para matriz Σ_k), duas interfaces estão disponíveis, sendo que a primeira especifica quais modelos serão utilizados a partir do argumento `models`:

- Estimando um modelo hipotético tipo 1(variâncias iguais e covariâncias fixadas em 0), com 3 perfis latentes:

```
estimate_profiles(dados,n_profiles = 3, models = 1)
```

- Estimando modelos hipotéticos tipo 1, 2 e 3, com 3 perfis latentes:

```
estimate_profiles(dados,n_profiles = 3, models = c(1:3))
```

A segunda forma é especificar as variâncias/covariâncias a serem estimadas através dos argumentos `variances` e `covariances`:

- Variâncias iguais por perfil e covariâncias fixas em zero:

```
estimate_profiles(dados,n_profiles = 3,
                  variances = "equal", covariances = "zero")
```

- Variâncias diferentes por perfil e covariâncias iguais por perfil:

```
estimate_profiles(dados,n_profiles = 3,
                  variances = "varying", covariances = "equal")
```

Da mesma forma que o argumento `models`, é possível estimar mais de um tipo de modelo simultaneamente, utilizando a segunda interface de especificação:

- Estimando dois modelos simultaneamente:

```
estimate_profiles(dados,n_profiles = 3,
variances = c("equal", "varying"),covariances = c("zero", "equal"))
```

Funções úteis

Um conjunto de funções auxiliares para implementação do método de LPA via R é apresentado a seguir:

- `compare_solutions()`: Esta função é útil quando há a necessidade de comparar diferentes tipos de modelo LPA, com diferentes números de perfis, para saber qual é o mais adequado aos dados segundo os critérios de ajuste de modelo AIC, BIC e os critérios AWE (Approximate Weight of Evidence, em inglês), CLC (Classification Likelihood Criterion, em inglês) e KIC (Kullback Information Criterion, em inglês), que podem ser encontrados em Akogul e Erisoglu (2017). Algumas formas que estas funções podem ser utilizadas:

– Atribuindo conjuntos de modelos a um objeto da classe '*tidyLPA*':

```
modelo=estimate_profiles(dados,n_profiles=1:4, models = c(1:3,6))
```

– Comparando apenas BIC (*default* da função):

```
compare_solutions(modelo)
```

– Comparando apenas AIC, BIC e valor de suas Log-verossimilhanças:

```
compare_solutions(modelo,statistics = c("AIC","BIC","LogLik"))
```

Caso a função aplicada ao modelo seja armazenada em um objeto, o mesmo terá a classe 'bestLPA' e com ele é possível que sejam criados gráficos dos critérios de ajuste por modelo usando a função `plot()`:

– Armazenando o objeto resultante da função:

```
comparacao=compare_solutions(modelo)
```

– Construindo gráfico do critério BIC:

```
plot(comparacao)
```

– Armazenando objeto resultante da função, com mais de um critério de ajuste:

```
comparacao2=compare_solutions(modelo,statistics = c("AIC",
"BIC","AWE","CLC"))
```

– Construindo gráfico dos critérios AIC, BIC, AWE e CLC:

```
plot(comparacao2)
```

Após executada a função, o *output* apresenta uma sugestão do modelo com melhor ajuste levando em consideração o conjunto de critérios de ajuste avaliados.

- `get_data()`: Com essa função é possível obter base de dados com respostas de cada indivíduo para os indicadores, em diferentes modelos, com diferentes números de perfis. Pode ser utilizada da seguinte forma:

  – Estimando modelos hipotéticos:

```
modelo=estimate_profiles(dados,n_profiles=1:4, models = c(1:3,6))
```

– Base de dados com respostas individuais baseadas em LPA:

```
base_dados=get_data(modelo$model_x_class_x)
```

Para essa função é necessário especificar o tipo e o número de perfis do modelo que se deseja ajustar aos dados. Para isto basta trocar o algarismo **x** pelo tipo e numero de perfis do modelo respectivamente.

- `get_estimate()`: Com essa função é possível obter as estimativas para os parâmetros para cada tipo de modelo, com diferentes números de perfis. Pode ser utilizada da seguinte forma:

```
#Estimando modelos hipotéticos
modelos=estimate_profiles(dados,n_profiles=1:4, models = c(1:3,6))
```

– Guardando estimativas de ambos modelos em um objeto:

```
estimativas=get_estimates(modelos)
```

– Guardando estimativas de um modelo particular em um objeto:

```
estimativas=get_estimates(modelos$model_x_class_x)
```

Assim como na função `get_data()`, quando se deseja obter informações de um modelo específico que está contido em um objeto do tipo *tidyLPA*, é necessário informar o tipo e número de perfis do mesmo substituindo o algarismo `x`.

- `plot_density()`: Com essa função é possível construir gráficos das densidades específicas por perfil de um ou mais modelos, com diferentes números de perfis, em relação a cada um dos indicadores, informando a separação entre perfis em relação às variáveis indicadoras. Pode ser utilizada da seguinte forma:

```
#Estimando modelos hipotéticos
modelos=estimate_profiles(dados,n_profiles=1:4, models = c(1:3,6))
```

– Construindo grafico de desidades especificas para multiplos modelos:

```
plot_density(modelos)
```

Caso deseje-se construir o gráfico de densidade específica apenas para um modelo de interesse é necessário reestimar o novo modelo com tipo e número de perfis específicos, armazenar em um objeto que terá a classe *tidyLPA*, e utilizar a função da seguinte forma:

- Novo modelo hipotético com tipo e número de perfis específico:

```
novo_modelo=estimate_profiles(dados,n_profiles=3, models = 6)
```

- Construindo gráfico de densidades especificas para novo modelo:

```
plot_density(novo_modelo)
```

- `plot_profiles()`: Cria gráficos de pontos separando indivíduos por perfis em relação às variáveis indicadores e número de perfis em cada modelo, disponibilizando as seguintes informações:
  - Barras refletindo um intervalo de confiança para os centróides da classe.
  - Caixas que refletem os desvios-padrão dentro de cada classe; uma caixa engloba cerca de 64% das observações em uma distribuição normal.
  - Dados brutos, cuja coloração é ponderada pela probabilidade posterior de classe, de modo que cada ponto de dados seja mais visível para a classe que seja mais provável ser membro.

A função **plot_profiles()** pode ser utilizada da seguinte forma:

```
#Estimando modelos hipotéticos
modelos=estimate_profiles(dados,n_profiles=1:4, models = c(1:3,6))

#Construindo gráfico de perfis por variaveis indicadoras
plot_profiles(modelos)
```

Caso deseje-se construir o gráfico de perfis apenas para um modelo de interesse é necessário reestimar o novo modelo com tipo e número de perfis específicos, armazenar em um objeto que terá a classe *tidyLPA*, e utilizar a função da seguinte forma:

- Novo modelo hipotético com tipo e numero de perfis especifico:

```
novo_modelo=estimate_profiles(dados,n_profiles=3, models = 6)
```

– Construindo grafico de desidades especificas para multiplos modelos:

```
plot_density(novo_modelo)
```

- `print()`: Retorna um conjunto de informações solicitadas sobre um ou vários modelos contidos em um objeto do tipo *tidyLPA*. Essas informações podem ser os critérios de ajuste utilizados pela função **compare_solutions()** ou ainda valores de log-verossimilhança e entropia dos modelos. A função pode ser utilizada da seguinte forma:

```
#Modelo hipotético
modelo=estimate_profiles(dados,n_profiles=1:4, models = c(1:3,6))

#Exibindo AIC, BIC, Entropia e Log-verossimilhança de modelos
print(modelo,stats = c("AIC", "BIC", "Entropy", "LogLik","Entropy"))
```

6 RESULTADOS

Neste capítulo são apresentados os resultados das análises de dados com implementação das metodologias estudados no *software* R para os modelos com equações estruturais (SEM), análise de classes latentes (LCA) e análise de perfis latentes (LTA).

6.1 Aplicações usando SEM

Para ilustrar a implementação dos métodos de estimação para ajuste de SEM são utilizados dois conjuntos de dados: o primeiro intitulado Democracia Política (PoliticalDemocracy), disponível através do próprio pacote lavaan e com diversas aplicações em Bollen (1989), que contém diversas medidas de democracia política e industrialização em países em desenvolvimento entre 1960 e 1965; o segundo conjunto de dados (STATAExample32g), disponível em StataCorp (2017), registra resultados de um instrumento fictício medindo a capacidade matemática através de variáveis binárias (q1-q8) e a atitude de cada aluno em relação à matemática usando as variáveis na escala Likert (att1-att5). Esse segundo exemplo está disponível no manual do *software* STATA. Os procedimentos iniciais para leitura da base de dados e instalação do pacote lavaan são apresentados a seguir:

Inicialmente foi realizada análise exploratória para avaliação de pressupostos importantes para os métodos de estimação dos parâmetros desse modelo, sobretudo a normalidade multivariada. Foram implementados testes para avaliação da normalidade multivariada (testes de Mardia, de Henze-Zirklers e de Royston) na análise dos dois conjuntos de dados usando o pacote MVN do *software* R.

6.1.1 Análise exploratória e avaliação de pressupostos

No contexto de modelagem com equações estruturais (SEM, em inglês) existem diversos métodos de estimação usualmente utilizados no processo de estimação dos parâmetros desse modelo. Os métodos de estimação mais comumente utilizados são máxima verossimilhança (EMV) e o método de mínimos quadrados (MMQ), que estão disponíveis na maioria dos *software*s que possuem algum módulo de implementação de SEM, a citar o R, MPLUS, LISREL, dentre outros. Todavia, uma suposição importante para a validade da estimação dos parâmetros do modelo por meio dos métodos anteriormente citados é que as variáveis sob investigação obedeçam ao pressuposto de normalidade multivariada. Esse pressuposto crucial, no entanto, nem sempre é verificado na fase inicial das análises, o que implica na geração de estimativas não confiáveis, ou pouco precisas para os parâmetros do modelo sob investigação, que podem consequentemente gerar conclusões errôneas acerca do fenômeno estudado. A seguir é apresentada a sintaxe para instalação

dos pacotes necessários para análise usando o *software* R, bem como a leitura dos dados.

No *software* R os pacotes necessários para essa avaliação podem ser instalados da seguinte forma:

- Instalando pacote *lavaan* para a implementação dos modelos SEM e pacote *MVN* utilizado na implementação dos testes de normalidade multivariada:

```
#lavaan
install.packages("lavaan")
#MVN
install.packages("MVN")
```

- Carregando pacotes:

```
require(lavaan)
require(MVN)
```

As bases de dados usadas nesta secção do relatório podem ser lidas usando:

1. Dados *Political Democracy* (Bollen, 1989):

```
require(lavaan)
data("PoliticalDemocracy")
```

2. Dados STATA-Example32g (StataCorp., 2017):

```
library(haven)
gsem_cfa <- read_dta("http://www.stata-press.com/data/r13/gsem_cfa.dta")
```

Como a base de dados Política e Democracia é constituída de variáveis contínuas, uma breve análise gráfica com histogramas para cada variável bem como o cálculo de medidas resumo são apresentados. Embora as variáveis da base de dados STATA-Example32g sejam categóricas os testes de normalidade multivariada também serão realizados, visando apenas corroborar a importância do pressuposto para os métodos de estimação abordados nessa secção.

```
#Histogramas univariados para a base de dados política e democracia
require(MVN);require(lavaan)
mvn(PoliticalDemocracy,univariatePlot = "histogram")
```

Figura 2 – Histogramas para as variáveis da base de dados Política e Democracia

Fonte: o autor (2019)

Já a tabela 1 apresenta as medidas resumo para as variáveis da base de dados Política e Democracia.

Tabela 1 – Medidas resumo para variáveis Política e Democracia (n=75)

Variáveis	Média	Desvio padrão	Mediana	Amplitude	Assimetria	Curtose
y1	5.46	2.62	5.40	8.75	-0.09	-1.15
y2	4.26	3.95	3.33	10.00	0.32	-1.47
y3	6.56	3.28	6.67	10.00	-0.59	-0.72
y4	4.45	3.35	3.33	10.00	0.12	-1.21
y5	5.14	2.61	5.00	10.00	-0.23	-0.78
y6	2.98	3.37	2.23	10.00	0.89	-0.47
y7	6.20	3.29	6.67	10.00	-0.55	-0.73
y8	4.04	3.25	3.33	10.00	0.45	-0.96
x1	5.05	0.73	5.08	2.95	0.25	-0.75
x2	4.79	1.51	4.96	6.49	-0.35	-0.57
x3	3.56	1.41	3.57	5.42	0.08	-0.94

A seguir são apresentados os resultados dos testes de Mardia, de Henze-Zirklers e de Royston para averiguar a existência de normalidade multivariada na análise do primeiro conjunto de dados denominado Política e Democracia.

1. Teste Mardia

```
require(MVN)
result <- mvn(data = PoliticalDemocracy, mvnTest = "mardia")
result$multivariateNormality

##              Test         Statistic           p value Result
## 1 Mardia Skewness  344.494411631556 0.010065220670416     NO
```

```
## 2 Mardia Kurtosis -1.22166682255403 0.221833647591673    YES
## 3             MVN               <NA>               <NA>     NO
```

2. Teste de Henze-Zirklers

```
require(MVN)
result <- mvn(data = PoliticalDemocracy, mvnTest = "hz")
result$multivariateNormality

##            Test       HZ      p value MVN
## 1 Henze-Zirkler 1.044869 5.276166e-10  NO
```

3. Teste de Royston

```
require(MVN)
result <- mvn(data = PoliticalDemocracy, mvnTest = "royston")
result$multivariateNormality

##      Test        H      p value MVN
## 1 Royston 145.5517 1.518825e-26  NO
```

Verificou-se que todos os testes escolhidos rejeitaram a hipótese de normalidade multivariada, ao nível de 5% de significância.

Análise similar foi feita para o segundo conjunto de dados, denominado STATA-Example32g. Os resultados são apresentados a seguir.

1. Teste Mardia

```
require(MVN)
result <- mvn(data = gsem_cfa, mvnTest = "mardia")
result$multivariateNormality

##              Test         Statistic           p value Result
## 1 Mardia Skewness  1169.58404197763 0.999385215378509    YES
## 2 Mardia Kurtosis -8.31556979439821                 0     NO
## 3             MVN               <NA>              <NA>     NO
```

2. Teste de Henze-Zirklers

```
require(MVN)
result <- mvn(data = gsem_cfa, mvnTest = "hz")
result$multivariateNormality

##            Test      HZ p value MVN
## 1 Henze-Zirkler 1.01103       0  NO
```

3. Teste de Royston

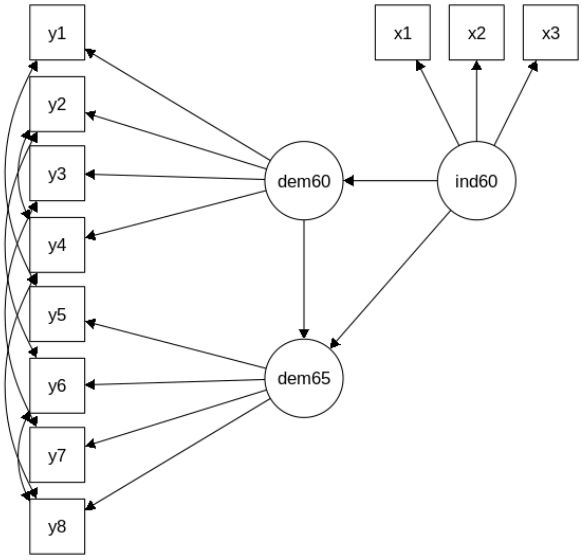
```
require(MVN)
result <- mvn(data = gsem_cfa, mvnTest = "royston")
result$multivariateNormality

##      Test        H       p value MVN
## 1 Royston 1440.991 7.171084e-297  NO
```

Os testes de Mardia, de Henze-Zirklers e de Royston rejeitaram a hipótese de normalidade multivariada, ao nível de 5% de significância, para a segunda aplicação, o que já era esperado dada a natureza categórica das variáveis.

Implementação dos modelos teóricos para SEM

O modelo teórico para a primeira aplicação é apresentado na Figura 3. A seguir é apresentada a sintaxe no *software* R para implementação do modelo e uso de diferentes métodos de estimação para análise destes dados.

Figura 3 – Diagrama de caminhos para os dados de política democrática



Fonte: (Rosseel,Y., 2012)

A sintaxe do pacote *lavaan* do *software* R correspondente para especificar o diagrama da Figura 3 é definida como:

```
model <- '
# measurement model
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8
# regressions
dem60 ~ ind60
dem65 ~ ind60 + dem60
# residual correlations
y1 ~~ y5
y2 ~~ y4 + y6
y3 ~~ y7
y4 ~~ y8
y6 ~~ y8
'
```

Para a implementação dos métodos de estimação foram utilizadas as seguintes opções:

1. Estimação via máxima verossimilhança

```
fit_ML <- sem(model, data=PoliticalDemocracy,estimator="ML")
```

2. Estimação via método de mínimos quadrados generalizados

```
fit_GLS=sem(model, data=PoliticalDemocracy,estimator="GLS")
```

Para as demais variações desses métodos de estimação é necessário mudar o método em **estimator**, conforme descrito anteriormente.

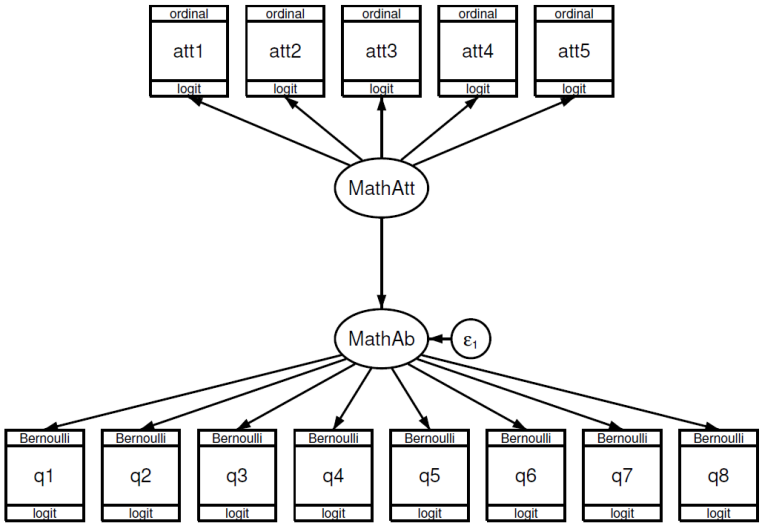
Na análise dos dados da aplicação 1, contudo, houve problema de convergência para o método IV (mínimos quadrados ponderados, WLS em inglês), que apresentou a seguinte mensagem de erro:

```
## Warning in lav_model_vcov(lavmodel = lavmodel,
    lavsamplestats = lavsamplestats, :  lavaan WARNING:
##    Could not compute standard errors!  The information matrix could
##    not be inverted.  This may be a symptom that the model is not
##    identified.
## Warning in lav_object_post_check(object):  lavaan WARNING: some estimated
lv variances are negative
## Warning in lavaan::lavaan(model = model, data = PoliticalDemocracy,
    estimator = "WLS", :  lavaan WARNING: not all elements of the gradient
are (near) zero;
##                  the optimizer may not have found a local solution;
##                  use lavInspect(fit, "optim.gradient") to investigate
```

Não sendo possível a obtenção das estimativas dos parâmetros devido à não inversibilidade da matriz de informação, o que torna o modelo não identificável.

O modelo teórico para a aplicação utilizando os dados STATA-Example32g é apresentado na Figura 4.

Figura 4 – Diagrama de caminhos para o modelo sobre capacidade e atitude matemática



Fonte: (StataCorp., 2017)

A sintaxe do pacote *lavaan* do *software* R correspondente para especificar o diagrama da Figura 4 é definida como:

```
model2 <- '
# measurement model
MathAtt =~ att1 + att2 + att3 + att4 + att5
MathAb =~ q1 + q2 + q3 + q4 + q5 + q6 + q7 + q8
# regressions
MathAb ~ MathAtt
'
```

Para a implementação de outros métodos de estimação para a aplicação 2 foram utilizadas as seguintes opções:

1. Estimação via máxima verossimilhança

```
fit2_ML <- sem(model2, data=gsem_cfa,estimator="ML")
```

2. Estimação via método de mínimos quadrados generalizados

```
fit2_GLS=sem(model2, data=gsem_cfa,estimator="GLS")
```

Comparação entre as estimativas dos parâmetros do modelo via diferentes métodos de estimação

Os resultados do ajuste destes modelos com a comparação das estimativas obtidas pelos diferentes métodos de estimação são apresentados nas Tabelas 2 e 3.

Tabela 2 – Cargas fatoriais e coeficientes de regressão padronizados para diferentes métodos de estimação para os dados sobre Política e Democracia

	Máx.Vero	Máx.Vero-II	Máx.Vero-III	MMQ	MMQ-II	MMQ-III
Cargas Fat. **(SE)**						
x_1	0.920 *	0.920 *	0.920 *	0.926 *	0.966 *	0.983 *
x_2	0.973 **(0.139)**	0.973 **(0.144)**	0.973 **(0.145)**	0.959 **(0.174)**	0.965 **(0.249)**	0.985 **(0.097)**
x_3	0.872 **(0.152)**	0.872 **(0.139)**	0.872 **(0.140)**	0.870 **(0.196)**	0.835 **(0.205)**	0.835 **(0.080)**
y_1	0.850 *	0.850 *	0.850 *	0.874 *	0.857 *	0.857 *
y_2	0.717 **(0.182)**	0.717 **(0.140)**	0.717 **(0.150)**	0.764 **(0.198)**	0.687 **(0.134)**	0.707 **(0.016)**
y_3	0.722 **(0.151)**	0.722 **(0.134)**	0.722 **(0.130)**	0.751 **(0.149)**	0.696 **(0.110)**	0.680 **(0.012)**
y_4	0.846 **(0.145)**	0.846 **(0.127)**	0.846 **(0.146)**	0.853 **(0.158)**	0.877 **(0.140)**	0.868 **(0.017)**
y_5	0.808 *	0.808 *	0.808 *	0.822 *	0.827 *	0.796 *
y_6	0.746 **(0.169)**	0.746 **(0.171)**	0.746 **(0.181)**	0.801 **(0.197)**	0.715 **(0.124)**	0.733 **(0.016)**
y_7	0.824 **(0.160)**	0.824 **(0.166)**	0.824 **(0.173)**	0.848 **(0.176)**	0.817 **(0.122)**	0.829 **(0.016)**
y_8	0.828 **(0.158)**	0.828 **(0.171)**	0.828 **(0.189)**	0.840 **(0.181)**	0.820 **(0.132)**	0.834 **(0.017)**
Regressões **(SE)**						
$dem_{60} \leftarrow ind_{60}$	0.447 **(0.399)**	0.447 **(0.338)**	0.447 **(0.342)**	0.466 **(0.487)**	0.445 **(0.227)**	0.432 **(0.092)**
$dem_{65} \leftarrow ind_{60}$	0.182 **(0.221)**	0.182 **(0.206)**	0.182 **(0.225)**	0.191 **(0.276)**	0.172 **(0.277)**	0.150 **(0.063)**
$dem_{65} \leftarrow dem_{60}$	0.885 **(0.098)**	0.885 **(0.086)**	0.885 **(0.087)**	0.872 **(0.101)**	0.898 **(0.124)**	0.909 **(0.016)**

Legenda: **Máx.Vero**: Máxima Verossimilhança; **Máx.Vero- II**: Máx.Vero. com erros padronizados robustos e uma estatística de teste escalonada Satorra-Bentler; **Máx.Vero- III**: Máx.Vero. com erros padronizados robustos (Huber-White); **MMQ**:Mínimos quadrados generalizados; **MMQ-II**: Mínimos quadrados diagonalmente ponderados; **MMQ-III**: Mínimos quadrados não ponderados; Asterisco: Ausência de erro padrão.

Tabela 3 – Cargas fatoriais e coeficientes de regressão padronizados para diferentes métodos de estimação para os dados sobre Capacidade Matemática

	Máx.Vero	Máx.Vero- II	Máx.Vero- III	MMQ	MMQ-II	MMQ-III	MMQ-IV (ADF)
Cargas Fat. **(SE)**							
att_1	0.550 *	0.550 *	0.550 *	0.542 *	0.518 *	0.569 *	0.524 *
att_2	0.252 **(0.105)**	0.252 **(0.101)**	0.252 **(0.099)**	0.260 **(0.111)**	0.237 **(0.82)**	0.261 **(0.035)**	0.246 **(0.098)**
att_3	-0.664 **(0.163)**	-0.664 **(0.147)**	-0.664 **(0.142)**	-0.662 **(0.174)**	-0.670 **(0.164)**	-0.649 **(0.072)**	-0.705 **(0.145)**
att_4	-0.488 **(0.124)**	-0.488 **(0.118)**	-0.488 **(0.119)**	-0.484 **(0.130)**	-0.515 **(0.121)**	-0.486 **(0.049)**	-0.488 **(0.110)**
att_5	0.334 **(0.111)**	0.334 **(0.107)**	0.334 **(0.112)**	0.314 **(0.118)**	0.345 **(0.095)**	0.329 **(0.039)**	0.331 **(0.103)**
q_1	0.542 *	0.542 *	0.542 *	0.553 *	0.542 *	0.527 *	0.537 *
q_2	0.242 **(0.109)**	0.242 **(0.107)**	0.242 **(0.113)**	0.251 **(0.108)**	0.238 **(0.076)**	0.171 **(0.173)**	0.265 **(0.098)**
q_3	0.359 **(0.120)**	0.359 **(0.117)**	0.359 **(0.125)**	0.371 **(0.119)**	0.386 **(0.094)**	0.523 **(0.246)**	0.423 **(0.111)**
q_4	0.207 **(0.108)**	0.207 **(0.105)**	0.207 **(0.107)**	0.201 **(0.105)**	0.195 **(0.074)**	0.097 **(0.166)**	0.200 **(0.096)**
q_5	0.485 **(0.134)**	0.485 **(0.122)**	0.485 **(0.116)**	0.479 **(0.128)**	0.481 **(0.109)**	0.449 **(0.227)**	0.502 **(0.107)**
q_6	0.389 **(0.122)**	0.389 **(0.116)**	0.389 **(0.121)**	0.387 **(0.119)**	0.386 **(0.094)**	0.325 **(0.198)**	0.405 **(0.104)**
q_7	0.442 **(0.128)**	0.442 **(0.124)**	0.442 **(0.127)**	0.449 **(0.125)**	0.417 **(0.099)**	0.263 **(0.187)**	0.462 **(0.112)**
q_8	0.383 **(0.122)**	0.383 **(0.117)**	0.383 **(0.120)**	0.385 **(0.120)**	0.396 **(0.096)**	0.438 **(0.224)**	0.419 **(0.108)**
Regressões **(SE)**							
$MathAb \leftarrow MathAtt$	0.472 **(0.028)**	0.472 **(0.027)**	0.472 **(0.030)**	0.517 **(0.032)**	0.481 **(0.022)**	0.516 **(0.149)**	0.558 **(0.027)**

Legenda: **Máx.Vero**: Máxima Verossimilhança; **Máx.Vero- II**: Máx.Vero. com erros padronizados robustos e uma estatística de teste escalonada Satorra-Bentler; **Máx.Vero- III**: Máx.Vero. com erros padronizados robustos (Huber-White); **MMQ**:Mínimos quadrados generalizados; **MMQ-II**: Mínimos quadrados diagonalmente ponderados; **MMQ-III**: Mínimos quadrados não ponderados; **MMQ-IV**: Mínimos quadrados ponderados; Asterisco: Ausência de erro padrão.

Apesar da diferença na definição matemática entre os dois principais métodos de estimação, não são observadas diferenças importantes nos resultados das análises conduzidas com os dados da primeira aplicação, a não ser pelo método de mínimos quadrados não ponderados, o qual obteve erros-padrão entre 2 a 3 vezes menores. Para as análises conduzidas com a segunda base de dados quase todos os tipos de estimação apresentaram

estimativas e erros-padrão muito próximos, com uma leve ressalva para a estimação via mínimos quadrados ponderados, que genericamente apresentou erros-padrão menores que os demais. Entretanto, em ambos os conjuntos de dados o pressuposto de normalidade multivariada não foi atendido. Deste modo, recomenda-se o uso das estimativas obtidas via ADF.

Ressalta-se ainda que mesmo no caso da segunda aplicação, a interpretação das estimativas torna-se complexa porque os modelos de equações estruturais definidos nas equações (3.1), (3.2) e (3.3) assumem uma relação linear entre indicadores e construtos. No entanto, a natureza das variáveis da segunda base de dados é dicotômica ou politômica, tornando assim inadequado o uso de um modelo linear para obtenção de quaisquer estimativas. Neste caso, recomenda-se o uso de um modelo linear generalizado para SEM que atualmente está disponível no *software* STATA. Para que as estimativas obtidas por esta abordagem pudessem ser comparadas com as obtidas pelos métodos disponíveis no pacote lavaan do R, estão apresentadas as estimativas não padronizadas do modelo na Tabela 4.

Tabela 4 – Estimativas não padronizadas do modelo para avaliar capacidade matemática usando diferentes métodos de estimação nos *softwares* R e STATA

	Máx.Vero	Máx.Vero- II	Máx.Vero- III	MMQ	MMQ-II	MMQ-III	MMQ-IV (ADF)	STATA
Cargas Fat. (**SE**)								
att1	1 *	1 *	1 *	1 *	1 *	1 *	1 *	1 *
att2	0.446 (0.105)	0.446 (0.101)	0.446 (0.099)	0.476 (0.111)	0.444 (0.082)	0.446 (0.035)	0.457 (0.098)	0.379 (0.097)
att3	-1.232 (0.163)	-1.232 (0.147)	-1.232 (0.142)	-1.268 (0.174)	-1.321 (0.164)	-1.164 (0.072)	-1.367 (0.145)	-1.593 (0.361)
att4	-0.865 (0.124)	-0.865 (0.118)	-0.865 (0.119)	-0.874 (0.130)	-0.969 (0.121)	-0.832 (0.049)	-0.901 (0.110)	0.810 (0.153)
att5	0.597 (0.111)	0.597 (0.107)	0.597 (0.112)	0.579 (0.118)	0.656 (0.095)	0.569 (0.039)	0.619 (0.103)	0.523 (0.117)
q1	1 *	1 *	1 *	1 *	1 *	1 *	1 *	1 *
q2	0.436 (0.109)	0.436 (0.107)	0.436 (0.113)	0.437 (0.108)	0.429 (0.076)	0.318 (0.173)	0.481 (0.098)	0.345 (0.105)
q3	0.661 (0.120)	0.661 (0.117)	0.661 (0.125)	0.654 (0.119)	0.711 (0.094)	0.990 (0.246)	0.785 (0.111)	0.545 (0.139)
q4	0.378 (0.108)	0.378 (0.105)	0.378 (0.107)	0.358 (0.105)	0.356 (0.074)	0.181 (0.166)	0.367 (0.096)	0.286 (0.095)
q5	0.894 (0.134)	0.894 (0.122)	0.894 (0.116)	0.868 (0.128)	0.888 (0.109)	0.852 (0.227)	0.934 (0.107)	0.817 (0.187)
q6	0.711 (0.122)	0.711 (0.116)	0.711 (0.121)	0.692 (0.119)	0.706 (0.094)	0.610 (0.198)	0.745 (0.104)	0.603 (0.147)
q7	0.814 (0.128)	0.814 (0.124)	0.814 (0.127)	0.796 (0.125)	0.770 (0.099)	0.498 (0.187)	0.858 (0.112)	0.721 (0.171)
q8	0.706 (0.122)	0.706 (0.117)	0.706 (0.120)	0.689 (0.120)	0.730 (0.096)	0.831 (0.224)	0.779 (0.108)	0.581 (0.143)
Regressões (**SE**)								
$MathAb \leftarrow MathAtt$	0.145 (**0.028**)	0.145 (**0.027**)	0.145 (**0.030**)	0.168 (**0.032**)	0.157 (**0.022**)	0.149 (**0.149**)	0.178 (**0.027**)	0.581 (**0.147**)

Máx.Vero: Máxima Verossimilhança, **Máx.Vero- II**: Máx.Vero. com erros padronizados robustos e uma estatística de teste escalonada Satorra-Bentler, **Máx.Vero- III**: Máx. Vero. com erros padronizados robustos (Huber-White). **MMQ**:Mínimos quadrados generalizados, **MMQ-II**: Mínimos quadrados diagonalmente ponderados, **MMQ-III**: Mínimos quadrados não ponderados, **MMQ-IV**: Mínimos quadrados ponderados. STATA: Estimativas via método STATA, *: Ausência de erro padrão.

Fonte: o autor (2021)

É possível observar que as estimativas oriundas do STATA diferem das obtidas pelos variados métodos de estimação do *software* R. Este resultado é esperado uma vez que o modelo logístico é utilizado no STATA, enquanto o modelo linear é usado no R (Tabela 4). Como o modelo aplicado pelo STATA foi o logístico conclui-se que o aumento de uma unidade no construto MathAb reflete num aumento de 0.345 no log da odds do indivíduo responder corretamente ao problema matemático 2 e o mesmo ocorre com as demais cargas fatoriais relativas aos 8 problemas matemáticos. Já para as variáveis att2 a att5, um aumento de uma unidade no construto MathAtt reflete num incremento no log da odds de 0.379 em att2 usando a função de ligação logito.

6.2 Aplicações com LCA

Para ilustrar a implementação dos métodos de estimação em LCA foram usados os dados de um estudo cujo propósito é descrever padrões de uso recente de maconha entre estudantes do ensino médio dos EUA. Considere dados parciais do estudo "*Monitoring the Future*" (1976-2013) que tem sido usado para estimar tendências históricas no uso de álcool, cigarro e maconha.

Tabela 5 – Descrição dos dados do estudo "*Monitoring the Future*"

Nome da variável	Descrição	Níveis de resposta
lifetime	Frequência de uso de maconha durante a vida do participante.	(1) Usou (2) Não usou
prev_yr	Frequência de uso de maconha ao longo do ano anterior.	(1) Usou (2) Não usou
prev_mo	Frequência de uso de maconha ao longo do mês anterior.	(1) Usou (2) Não usou
next_mo	Quão provável é que o participante use maconha no próximo ano.	(1)Usará (2)Não usará
aprv_try	O participante desaprova pessoas que tenham usado maconha uma ou duas vezes.	(1)Não desaprova (2)Desaprova
aprv_occ	O participante desaprova pessoas que fumam maconha ocasionalmente	(1)Não desaprova (2)Desaprova
aprv_regg	O participante desaprova pessoas que fumam maconha regularmente.	(1)Não desaprova (2)Desaprova
sex	Gênero do participante	(1)Masculino (2)Feminino
year	Ano da pesquisa	(1)1999 (2)2000 (3)2001

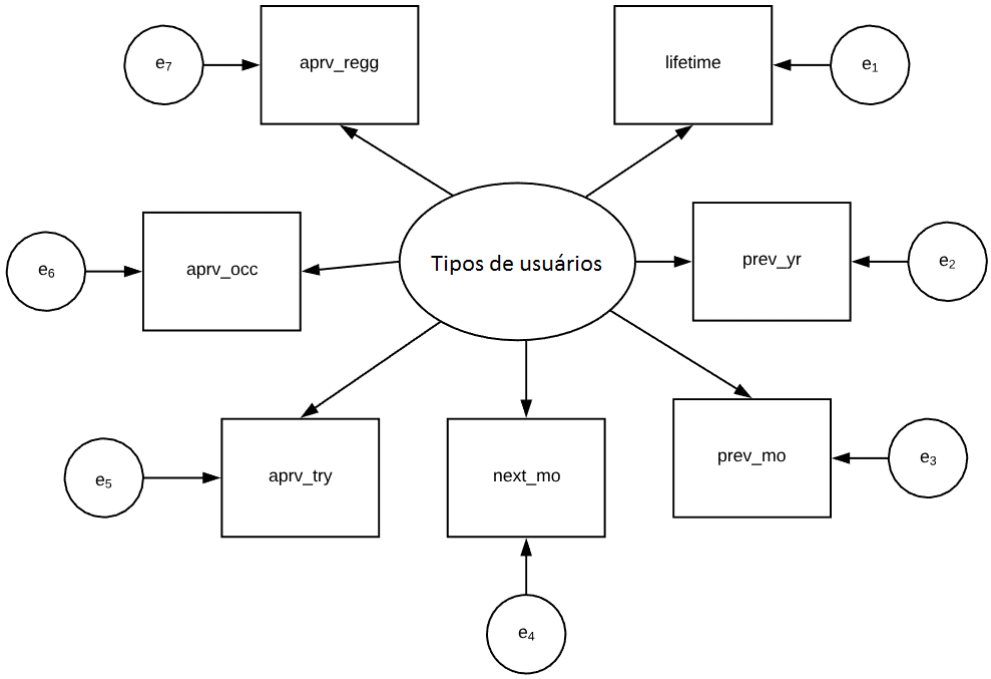
A leitura da base de dados, disponível no arquivo *"mjuseR.txt"*, é feita da seguinte forma:

```
#Base de dados disponível em arquivo externo
        #Obs: o diretório pode variar conforme local de download
require(readr)
mjuseR <- read_table2("mjuseR.txt")
```

Implementação do modelo teórico para LCA

O modelo teórico para a aplicação sobre o uso de maconha é apresentado na Figura 5. A seguir é apresentada a sintaxe no *software* R para implementação do modelo de classes latentes com o uso do método de estimação definido para o mesmo.

Figura 5 – Diagrama de caminhos para os dados do estudo *Monitoring the Future*



Fonte: o autor (2021)

Para aplicação do modelo foram consideradas 4 classes latentes, com base em uma análise prévia sobre o número de classes mais adequado considerando-se uma boa interpretabilidade e por critérios de seleção de modelos como AIC, BIC e entropia. Mais informações sobre os critérios de seleção de modelos no contexto de LCA podem ser em encontrados em Linzer e Lewis (2011).

A sintaxe *poLCA* correspondente para especificar o diagrama da Figura 5 é definida como:

```
#Especificando variáveis utilizadas na estimação do modelo:
formula=cbind(LIFETIME, PREV_YR, PREV_MO, NEXT_MO, APRV_TRY, APRV_OCC,
APRV_REG)~1
```

Dessa forma a estimação do modelo LCA, considerando 4 classes, pode ser feita via seguinte comando:

```
require(poLCA)
modelo=poLCA(formula,mjuseR,nclass=4,nrep = 50)
```

Como há uma quantidade grande de parâmetros a ser estimada, foram considerados 50 conjuntos de valores iniciais para estimação do modelo, no qual foram retornadas as estimativas dos parâmetros relativas ao modelo que apresentou maior valor de log-verossimilhança.

O resultado do ajuste do modelo é apresentado na Tabela 6. As 4 classes latentes foram organizadas de forma que os indivíduos dispostos na classe "Conservadores" têm altas probabilidades de não ter usado, não desejar usar, e nem aprovar o uso de maconha, enquanto que os indivíduos distribuídos nas classes "Experimentadores" têm probabilidades maiores de já terem usado maconha alguma vez nas suas vidas e probabilidades menores de desaprovação ao uso ou experimentação da droga se comparados com os indivíduos da classe anterior. Os indivíduos dispostos nas classes "Fumante ocasional" e "Fumante ativo" têm altas probabilidades de já terem experimentado e de terem consumido a droga em períodos recentes, e apresentam baixas probabilidades de desaprovação ao uso da droga. As prevalências de cada classe indicam a proporção estimada de indivíduos que compõem a mesma. Dessa forma, 55% dos indivíduos participantes da pesquisa foram classificados como "Conservadores" e aproximadamente 1% como "Experimentadores".

Tabela 6 – Estimativas para modelo de 4 classes aplicado aos dados do estudo *Monitoring the future*

	Conservadores	Experimentadores	Fumantes ocasionais	Fumantes ativos
Prevalência de classes	0.555	0.095	0.124	0.226
		Prob.resposta ao item		
Frequência do uso de maconha durante a vida do participante				
Usou	0.112	0.296	1.000	1.000
Não usou	0.888	0.704	0.000	0.000
Frequência de uso de maconha ao longo do ano anterior				
Usou	0.000	0.000	0.867	1.000
Não usou	1.000	1.000	0.133	0.000
Frequência de uso de maconha ao longo do mês anterior				
Usou	0.000	0.000	0.220	0.731
Não usou	1.000	1.000	0.780	0.269
Quão provável é que o participante use maconha no próximo ano				
Usará	0.015	0.196	0.288	0.875
Não usará	0.985	0.804	0.712	0.125
O participante desaprova pessoas que tenham usado maconha uma ou duas vezes				
Não desaprova	0.161	0.979	0.680	1.000
Desaprova	0.839	0.021	0.320	0.000
O participante desaprova pessoas que fumam maconha ocasionalmente				
Não desaprova	0.003	0.974	0.195	0.995
Desaprova	0.997	0.026	0.805	0.005
O participante desaprova pessoas que fumam maconha regularmente				
Não desaprova	0.001	0.404	0.009	0.596
Desaprova	0.999	0.596	0.991	0.404

Implementação do modelo teórico pra LCA com covariáveis

Dois modelos com a mesma quantidade de classes latentes foram estimados, mas com a inclusão de covariáveis. O primeiro modelo inclui a variável SEX (sexo) como covariável, e o segundo a variável YEAR (ano de pesquisa) como covariável. O objetivo é avaliar o efeito de ambas covariáveis na prevalência de cada uma das classes. A sintaxe para implementação do modelo com covariáveis para os mesmos dados é definida como:

```
require(poLCA)

#Especificando variáveis utilizadas na estimação do primeiro modelo:
var_sex=cbind(LIFETIME, PREV_YR, PREV_MO,
                NEXT_MO, APRV_TRY, APRV_OCC, APRV_REG)~as.factor(SEX)


#Especificando variáveis utilizadas na estimação do segundo modelo:
var_year=cbind(LIFETIME, PREV_YR, PREV_MO,
                NEXT_MO, APRV_TRY, APRV_OCC, APRV_REG)~as.factor(YEAR)

#Estimando modelos com 4 classes + covariável
modelo_cov<- poLCA(var_sex,mjuseR,nclass=4,nrep = 50)
modelo_cov_2<- poLCA(var_year,mjuseR,nclass=4,nrep = 50)
```

Como o modelo em questão tem 4 classes latentes e também uma covariável é importante que durante a estimação sejam considerados mais que um conjunto de chutes iniciais. Dessa forma são buscadas estimativas que reflitam em um ponto de máximo global na verossimilhança do modelo. Para realizar essa estimação foram considerados 50 conjuntos de valores iniciais (nrep = 50), obtendo-se as estimativas do modelo com a maior log-verossimilhança.

As estimativas dos modelos considerando as covariáveis sexo e ano da pesquisa são apresentadas na Tabela 7. Durante o processo de estimação do modelo com covariáveis o poLCA fixa uma classe latente e um dos níveis das covariáveis como referência. No presente estudo, a classe latente de referência escolhida foi "Conservadores", e os níveis de referência das covariáveis sexo e ano de pesquisa considerados foram os primeiros, que significam respectivamente (1) sexo masculino e (1) ano de 1999. Deste modo, as estimativas $\hat{\beta}$ exponenciadas (odds ratio) quando maiores que 1 representam o acréscimo na chance que um indivíduo do sexo feminino tem se comparado com um do sexo masculino de pertencer a uma determinada classe em relação a classe de referência ou acréscimo na chance que um indivíduo respondente do estudo nos anos de 2000 ou 2001 tem de pertencer a uma determinada classe em relação a classe de referência se comparados com os indivíduos respondentes do mesmo estudo no ano de 1999.

Tabela 7 – Estimativas para os modelos de classes latentes com covariáveis

Classes	Sexo			Ano de pesquisa		
	Nível da Covariável	$\hat{\beta}$(se)	$\hat{OR}[IC_{95\%}]$	Nível da Covariável	$\hat{\beta}$(se)	$\hat{OR}[IC_{95\%}]$
Conservadores	Masculino	1	-	1999	1	-
Experimentadores	Feminino	0.386 (0.156)	1.470 [1.084; 1.99]	2000	0.211 (0.188)	1.235 [0.854; 1.785]
				2001	0.267 (0.186)	1.306 [0.907; 1.881]
Fumantes ocasionais	Feminino	0.070 (0.139)	1.072 [0.817; 1.408]	2000	-0.179 (0.174)	0.836 [0.594; 1.176]
				2001	-0.015 (0.165)	0.985 [0.713; 1.361]
Fumantes ativos	Feminino	0.263 (0.108)	1.300 [1.053; 1.607]	2000	-0.308 (0.130)	0.735 [0.570; 0.948]
				2001	-0.222 (0.129)	0.801 [0.622; 1.031]

Legenda: Traço significa ausência de estimativa por ser a classe de referência

Os indivíduos do sexo feminino não apresentaram estimativas significativas da odds ratio apenas para a classe "Fumantes ocasionais" em relação à classe de referência, pois o intervalo de confiança das odds ratio dessa classe inclue o valor 1 (Tabela 7). Observou-se que os indivíduos do sexo feminino apresentam chances maiores de pertencer às classes "Experimentadores" e "Fumantes ativos" do que os do sexo masculino em relação à classe denominada "Conservadores". Para as classes "Experimentadores" e "Fumantes ativos" os indivíduos do sexo feminino apresentaram chances de 47% e 30%, respectivamente, de pertencerem às classes citadas em relação à classe "Conservadores".

Considerando o ano de pesquisa como covariável, apenas a classe de "Fumantes ativos", no ano de 2000, apresentou resultados estatisticamente significantes para a odds ratio. Logo, a classe "Fumantes ativos" em comparação com a classe "Conservadores" apresentou chances menores de pertencimento para os indivíduos respondentes da pesquisa no ano de 2000 se comparados com os respondentes no ano de 1999. Um indivíduo respondente no ano de 2000 tem 26,5% menos chances de pertencer a categoria "Fumantes ativos" se comparados com os do ano de 1999.

6.3 Aplicações com LPA

Para ilustrar a implementação dos métodos de estimação em LPA foi utilizado o conjunto de dados denominado iris. Este famoso conjunto de dados coletados pelo botânico Edgar Anders, em 1935, fornece as medidas, em centímetros, das variáveis comprimento e largura das sépalas e comprimento e largura da pétalas, respectivamente, para 50 flores de cada uma das 3 espécies Iris, que são: Iris setosa, versicolor e virginica. Esse conjunto de dados é frequentemente utilizado para ilustrar métodos e algoritmos de classificação. A seguir é apresentada a sintaxe para instalação dos pacotes necessários para análise de perfis latentes usando o *software* R, bem como a leitura dos dados.

1. Instalação

```
#Instalando pacotes
  install.packages("tidyLPA")
  install.packages("mclust")
```

2. Carregando pacote e dados Iris

```
require(tidyLPA)
data("iris")
```

Implementação do modelo teórico para LPA

Durante a implementação do método de LPA serão consideradas restrições para matriz Σ_k que tem opção suportada pelo *software* R, que compreende os modelos tipo 1, 2, 3 e 6, com definições feitas anteriormente. Foram inicialmente estimados modelos de 1 a 6 perfis latentes, que foram comparados de acordo com AIC e BIC. A estimação dos modelos LPA com diferentes tipos de restrição e número de perfis podem ser implementados da seguinte forma:

```
#Modelo com os dados iris
modelos= iris[,1:4] %>%
  estimate_profiles(1:6,models = c(1:3,6))
```

Após estimação, os modelos são armazenados em um objeto do tipo tidyLPA, denominado 'modelos' que é utilizado com a função `compare_solutions()`, que retorna o melhor modelo segundo os critérios de AIC e BIC. Essa sintaxe é dada por:

```
#Comparando modelos para obter melhor estatística de ajuste
modelos %>% compare_solutions(statistics = c("AIC","BIC"))

## Compare tidyLPA solutions:
##
## Model Classes AIC BIC
## 1 1 1498.035 1522.120
## 1 2 1003.830 1042.968
## 1 3 758.859 813.050
## 1 4 758.159 827.404
## 1 5 657.621 741.918
## 1 6 594.440 693.791
## 2 1 1498.035 1522.120
```

```
## 2 2 806.371 857.551
## 2 3 666.362 744.638
## 2 4 645.648 751.020
## 2 5 578.982 711.450
## 2 6 547.726 707.290
## 3 1 787.829 829.978
## 3 2 630.895 688.097
## 3 3 560.709 632.965
## 3 4 558.717 646.026
## 3 5 502.451 604.813
## 3 6 492.440 609.854
## 6 1 787.829 829.978
## 6 2 486.709 574.018
## 6 3 448.372 580.840
## 6 4 452.972 630.600
## 6 5 453.819 676.606
## 6 6 486.847 754.794
##
## Best model according to AIC is Model 6 with 3 classes.
## Best model according to BIC is Model 6 with 2 classes.
##
## An analytic hierarchy process, based on the fit indices AIC, AWE, BIC, CLC,
and KIC (Akogul & Erisoglu, 2017), suggests the best solution is Model 6 with 3
classes.
```

Analisando o *output* da função `compare_solutions()`, tomando como base o critério de ajuste AIC, o modelo tipo 6 (com variâncias e covariâncias diferentes por perfis) e número de perfis latentes igual a 3 apresentou melhor ajuste, já tendo o critério BIC como base o modelo tipo 6 com 2 perfis latentes apresentou o melhor ajuste. Contudo, a função `compare_solutions()` em seu *output* também fornece uma sugestão sobre qual modelo apresenta melhor ajuste aos dados estudados, avaliando além do AIC e BIC os critérios de ajuste AWE, CLC, e KIC via procedimento denominado Analytic Hierarchy Process (AHP), proposto por Akogul e Erisoglu (2017). Desta forma, o modelo tipo 6, com 3 perfis latentes de flores, é escolhido para análise dos dados íris.

```
#Selecionando o modelo com melhor ajuste
best_model=estimate_profiles(iris[,1:4],n_profiles=3,models = 6)
```

A Figura 6 mostra os gráficos de densidades específicas por perfil para o modelo com melhor ajuste. A Figura 6 fornece evidências de que o comprimento e largura das

pétalas são as variáveis que proporcionam o maior grau de separação entre as densidades específicas por perfil se comparadas com as medidas relacionadas às sépalas das flores.

```
#Construindo grafico de desidades especificas
plot_density(best_model)
```

Figura 6 – Gráficos de desidades específicas para os dados Iris

A Figura 7 também fornece as mesmas evidências de que as medidas das pétalas das flores são as variáveis que melhor auxiliam na distinção entre os três tipos de perfis de flores.

```
#Construindo grafico de perfis por variaveis indicadoras
plot_profiles(best_model)
```

Figura 7 – Gráfico de perfis para os dados Iris

A Tabela 8 fornece as médias, desvios-padrão e correlações das variáveis do modelo para a amostra global, bem como as mesmas medidas para cada um dos perfis acrescida da prevalência dos mesmos. Analisando primeiramente as variáveis para as 150 flores, sem considerar a pertinência aos perfis, o comprimento das sépalas das flores apresenta maior média enquanto a largura das pétalas apresenta a menor média. Ao analisar as correlações, a largura das sépalas apresentou baixa correlação com as demais medidas enquanto o comprimento e largura das pétalas possuem alta correlação com o comprimento das sépalas.

No perfil 1, com prevalência de 33%, as flores apresentam maior valor médio para a largura das sépalas se comparados com os perfis 2 e 3, o que fornece uma evidência de que a largura das sépalas discrimina bem essa classe. As flores do perfil 1 apresentam os menores valores médios de comprimento e largura das pétalas se comparados com os demais perfis. Em relação aos desvios-padrão, a largura das sépalas foi a que apresentou maior variabilidade. Nesse perfil a única correlação considerada alta ocorreu entre as medidas das sépalas (Tabela 8).

Nos perfis 2 e 3, com prevalências de 30% e 37%, respectivamente, as médias das variáveis indicadoras apresentaram comportamento análogo, sendo o comprimento das sépalas o que apresenta maior valor médio seguido da largura das sépalas e do comprimento e largura das pétalas, respectivamente. As correlações no perfil 2 apresentaram

valores moderados/altos, enquanto que no perfil 3 apenas as medidas de comprimento apresentaram correlação alta, com as demais correlações sendo moderadas/baixas. Os perfis 2 e 3 não apresentaram grandes distinções a não ser pelo perfil 3 apresentar flores com dimensões, em média, um pouco maiores que no segundo perfil, comportamento que pode também ser observado na Figura 7, que apresenta maior proximidade entre os perfis 2 e 3 (Tabela reftab:tabest).

Analisando as características apresentadas em cada um dos perfis, considerou-se os seguintes rótulos: "flores pequenas" (perfil 1), se distinguem por terem, em média, valores altos para largura das sépalas e pétalas menores do que as flores dos outros perfis, "flores médias" (perfil 2) e "flores grandes" (perfil 3). Estes dois últimos perfis se distinguem por medidas um pouco menores no perfil 2 do que no perfil 3.

Tabela 8 – Estimativas do modelo de perfis latentes para os dados Iris

					Correlações			
		Variável	Média	Desvio padrão	(1)	(2)	(3)	(4)
Amostra global	(1)	Sepal.Length	5.84	0.83	1			
	(2)	Sepal.Width	3.06	0.44	-0.12	1		
	(3)	Petal.Length	3.76	1.77	0.87	-0.43	1	
	(4)	Petal.Width	1.20	0.76	0.82	-0.37	0.96	1
Perfis($\hat{\pi}$)		Variável	Média ($\hat{\alpha}_{mk}$)	Desvio padrão ($\sqrt{\hat{\theta}_{mmk}}$)	Correlações			
					(1)	(2)	(3)	(4)
Perfil 1 (0.33)	(1)	Sepal.Length	5.01	0.35	1			
	(2)	Sepal.Width	3.43	0.38	0.74	1		
	(3)	Petal.Length	1.46	0.17	0.27	0.18	1	
	(4)	Petal.Width	0.25	0.11	0.28	0.23	0.33	1
Perfil 2 (0.30)	(1)	Sepal.Length	5.90	0.53	1			
	(2)	Sepal.Width	2.78	0.31	0.60	1		
	(3)	Petal.Length	4.19	0.44	0.78	0.67	1	
	(4)	Petal.Width	1.29	0.18	0.56	0.80	0.75	1
Perfil 3 (0.37)	(1)	Sepal.Length	6.55	0.62	1			
	(2)	Sepal.Width	2.95	0.33	0.45	1		
	(3)	Petal.Length	5.49	0.57	0.85	0.44	1	
	(4)	Petal.Width	1.99	0.29	0.32	0.58	0.42	1

A Tabela 9 fornece as medidas das variáveis indicadoras do modelo segundo os tipos de flores, já que a classificação verdadeira de cada espécie está presente na base de dados utilizada para ilustrar o método. As flores da espécie *Setosa* apresentam medidas semelhantes as classificadas no perfil 1, com largura média de sépala maior e pétalas menores que as das outras espécies. As flores das espécies *Versicolor* e *Virginica* apresentam medidas semelhantes aos perfis 2 e 3 respectivamente.

Tabela 9 – Média e desvios padrão por espécie das variáveis do conjunto de dados Iris

Variáveis	Virginica		Setosa		Versicolor	
	Média	Desvio padrão	Média	Desvio padrão	Média	Desvio padrão
Sepal.Length	6.59	0.64	5.01	0.35	5.94	0.52
Sepal.Width	2.97	0.32	3.43	0.38	2.77	0.31
Petal.Length	5.55	0.55	1.46	0.17	4.26	0.47
Petal.Width	2.03	0.27	0.25	0.11	1.33	0.20

7 DISCUSSÃO

As metodologias envolvendo modelagem de variáveis latentes tem se mostrado boas alternativas para o estudo de fenômenos considerados complexos nos quais a estatística clássica não consegue explicar satisfatoriamente. No contexto de variáveis latentes contínuas a metodologia de SEM vem sendo amplamente utilizada permitindo quantificar as relações construto-construto e construto indicadores, para o caso em que todos são contínuos. Entretanto, ao realizar a aplicação desta metodologia nem sempre os pesquisadores estão atentos à verificação dos pressupostos necessários, o que acarreta no comprometimento da interpretabilidade das estimativas do modelo. Deste modo, é proposta uma discussão acerta da importância da verificação dos pressupostos necessários para implementação do método. Para além dos métodos clássicos em SEM, que tem como enfoque a análise de variáveis contínuas, ressalta-se que já estão disponíveis métodos de estimação generalizados em SEM, que incorporam outros modelos, como, por exemplo, o logístico e o Poisson. No entanto, a implementação dessas metodologias ainda está restrita a alguns softwares estatísticos, e a interpretação das estimativas provenientes destas análises ainda pouco disseminada na literatura.

No contexto de variáveis latentes categóricas, foram apresentadas as metodologias de LCA e LPA que, a partir das categorias ou classes da variável latente, classificam indivíduos mediante as suas respostas aos indicadores do modelo, indicadores esses que são categóricos na LCA e contínuos na LPA. Embora existam outras metodologias para análise de clusters, como o método *k-means* no caso de indicadores contínuos, e a análise múltipla de correspondência para indicadores categóricos, as metodologias de LPA e LCA também apresentam grande potencial interpretativo, sendo a escolha da técnica mais apropriada direcionada pelo objetivo da análise.

Deste modo, espera-se contribuir para maior divulgação e utilização das metodologias apropriadas para cada tipo de variável por pesquisadores de diversas áreas do conhecimento, sistematizando os conceitos teóricos, a avaliação de pressupostos, adequabilidade dos modelos e fornecendo um tutorial para aplicação das referidas metodologias utilizando o *software* R, gratuito, facilitando, assim, a implementação dos métodos.

8 REFERÊNCIAS BIBLIOGRÁFICAS

1. Akogul, S., & Erisoglu, M. (2017). An Approach for Determining the Number of Clusters in a Model-Based Cluster Analysis. *Entropy*, 19(9), 1-15. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/e19090452>.

2. Amorim, L. et al. (2015) Análise de Classes Latentes: Um tutorial usando Software Estatístico. Relatório Técnico. Salvador: Universidade Federal da Bahia. Instituto de Matemática. 79 p. Disponível em: <http://repositorio.ufba.br/ri/handle/ri/18060>. Acesso em: 18 Mar. 2019.

3. Amorim, L. et al. (2012). Modelos de equações estruturais: princípios básicos e aplicações. Relatório Técnico. Salvador: Universidade Federal da Bahia. Instituto de Matemática, 47 p. Disponível em: <http://repositorio.ufba.br/ri/handle/ri/17684>. Acesso em: 9 Fev. 2019.

4. Bandeen-Roche K, Miglioretti DL, Zeger SL, Rathouz PJ. (1997). Latent Variable Regression for Multiple Discrete Outcomes. *Journal of the American Statistical Association*, 92 (440), 1375-1386.

5. Biemer, P. (2011). *Latent class analysis of survey error*. Hoboken, N.J.: Wiley.

6. Bollen, Kenneth A. (1989). *Structural Equations with Latent Variables*, New York: JohnWiley & Sons.

7. Collins, Linda M.; Lanza, Stephanie T. (2013). *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences*. John Wiley & Sons.

8. Everitt BS (1984). *An Introduction to Latent Variable Models*. London: Chapman and Hall.

9. Everitt, B. S., & Hand, D. J. (1981). *Finite mixture distributions*. London: Chapman and Hall.

10. Kaplan, D. (2000). *Structural Equation Modeling: Foundations and Extensions*. SAGE Publications.

11. 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.

12. Masyn, K.E. Latent Class Analysis and Finite Mixture Modeling. *In*: Little, Todd D.(org.). *The Oxford handbook of quantitative methods*, Vol. 2: Statistical Analysis. Oxford University Press, USA, 2013. p. 584-595.

13. Newsom, J. (2018). Alternative Estimation Methods. [ebook] PDF. Disponível em: <http://web.pdx.edu/~newsomj/semclass/ho_estimate.pdf>. Acesso em: 9 Fev. 2019.

14. Rosenberg, Joshua M. Introduction to tidyLPA. [ebook]. Disponível em: <https://cran.r-project.org/web/packages/tidyLPA/vignettes/Introduction_to_tidyLPA.html>. Acesso em: 20 jul. 2019.

15. Rosseel, Y. (2012). Lavaan: An R package for Structural Equation Modeling. *Journal of statistical software*, 48(2), 1-36.

16. Rosseel, Y. (2018). The lavaan tutorial. [ebook]. Disponível em: <http://lavaan.ugent.be/tutorial/tutorial.pdf> . Acesso em: Fev. 2019.

17. StataCorp. (2017). Stata 15. *Structural Equation Modeling: Reference Manual.* College Station, TX: Stata Press.