



JUSCELIA DIAS MENDONÇA

**ANÁLISE DA EFICIÊNCIA DE ESTIMAÇÃO DE
PARÂMETROS DA TRI PELO SOFTWARE ICL**

LAVRAS - MG

2012

JUSCELIA DIAS MENDONÇA

**ANÁLISE DA EFICIÊNCIA DE ESTIMAÇÃO DE PARÂMETROS DA
TRI PELO SOFTWARE ICL**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, área de concentração em Estatística e Experimentação Agropecuária, para a obtenção do título de Mestre.

Dr. Marcelo Silva de Oliveira
Orientador

Dr. Ulisses Azevedo Leitão
Coorientador

LAVRAS-MG

2012

**Ficha Catalográfica Elaborada pela Divisão de Processos Técnicos da
Biblioteca da UFLA**

Mendonça Juscelia Dias.

Análise da eficiência da estimação de parâmetros da Teoria de Resposta ao Item pelo Software ICL / Juscelia Dias Mendonça. – Lavras : UFLA, 2012.

125 p. : il.

Dissertação (mestrado) – Universidade Federal de Lavras, 2012.

Orientador: Marcelo Silva de Oliveira.

Bibliografia.

1. Simulações. 2. Psicometria. 3. Estudo da confiabilidade de software. 4. Modelos matemáticos. I. Universidade Federal de Lavras. II. Título.

CDD – 519.544

JUSCELIA DIAS MENDONÇA

**ANÁLISE DA EFICIÊNCIA DE ESTIMAÇÃO DE PARÂMETROS DA
TEORIA DE RESPOSTA AO ITEM PELO SOFTWARE ICL**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, área de concentração em Estatística e Experimentação Agropecuária, para a obtenção do título de Mestre.

Aprovada em 28 de fevereiro de 2012.

Dr. Ronei Ximenes Matins UFLA

Dr. Júlio Sílvio de Sousa Bueno Filho UFLA

Dr. Marcelo Silva de Oliveira
Orientador

Dr. Ulisses Azevedo Leitão
Coorientador

LAVRAS-MG

2012

DEDICATÓRIA

*Dedico esse trabalho à
minha mãe e ao meu filho
Mateus.*

AGRADECIMENTOS

Primeiramente, agradeço a Deus por ter me dado força, saúde e condições para a conclusão deste trabalho.

Agradeço à minha mãe Regina, por sempre ter tido palavras de incentivo nos momentos difíceis e ao meu filho Mateus, pela compreensão dos momentos de ausência.

Este trabalho não é um trabalho “meu” e sim um trabalho “nosso”, pois a sua conclusão é o resultado do trabalho de um grupo de pessoas. Por isso, agradeço ao meu orientador professor Dr. Marcelo Silva de Oliveira, por ter dividido comigo um pouco de seu conhecimento e por ter, cobrado e incentivado nos momentos certos. Agradeço ao meu coorientador, professor Dr. Ulisses Azevedo Leitão por sempre tão prestativamente ter me ajudado, agradeço também a seus orientados, Jefferson Adriano Neves e Thiago Dias, tanto pela elaboração do software que realizou as simulações, quanto pelos momentos de discussão.

Creio que sou uma pessoa privilegiada por ter feito parte da turma do mestrado de 2010, e ter tido a oportunidade de conhecer pessoas maravilhosas as quais levarei em meu coração por toda a minha vida. Por isso agradeço especialmente ao Juliano Bortolini e à Larissa Andrade por terem me ajudado nos momentos de dúvidas, agradeço à Danielle Gonçalves pela amizade e companheirismo. Não posso deixar de mencionar a Tábata, a Iábita, o Juracy, o Marcelo, a Mariele, o Wederson e a Elayne que também fizeram parte desses anos de minha vida e que também contribuíram para meu crescimento.

Agradeço a todos os professores do DEX pelos ensinamentos proporcionados, em especial, ao professor Dr. Júlio Sílvio de Sousa Bueno

Filho, por ter sido sempre tão prestativo. Agradeço também a todos os funcionários do DEX e a CAPES pela bolsa concedida.

Agradeço a todas as pessoas que tiveram pensamento positivo para meu sucesso na conclusão deste trabalho, ao meu irmão Juliano, aos meus amigos e familiares de Uberlândia, aos professores da UFU onde fiz a graduação, em especial, agradeço ao professor Dr. Ednaldo Carvalho Guimarães, por todo o incentivo no ingresso ao mestrado.

A todos, muito obrigada!

RESUMO

Atualmente, importantes decisões como o planejamento de políticas públicas educacionais e a seleção de concursos públicos, são tomadas a partir de resultados obtidos em algumas avaliações. Devido a tal importância, os instrumentos utilizados para a obtenção dos resultados devem ser válidos e precisos. Com esse objetivo, a Teoria de Resposta ao Item (TRI) que é um conjunto de modelos matemáticos, que procuram representar a probabilidade de um indivíduo dar certa resposta a um item, em função de características do item e do traço latente dos examinados, vem sendo desenvolvida. O modelo mais utilizado atualmente é o modelo logístico de três parâmetros para itens dicotômicos e unidimensionais. Este modelo possui três parâmetros dos itens que são: discriminação (a), dificuldade (b), acerto casual de indivíduos com baixa habilidade (c), e a habilidades dos examinados (θ). Para estimar tais parâmetros, existem softwares específicos, livres e comerciais. O ICL (*Item response theory Command Language*) é um software livre que faz estimação de parâmetros dos modelos da TRI, ele foi desenvolvido por Bradley Alvin Hanson e utiliza o algoritmo EM para realizar suas estimativas. Neste trabalho, foram realizadas simulações verificando o comportamento do software frente ao aumento de número de itens e examinados, e maior variação do parâmetro de discriminação e dificuldade. Os resultados mostram que o ICL, nas situações analisadas, produz estimativas próximas das verdadeiras para os parâmetros, sendo que os melhores resultados são obtidos na estimação das habilidades dos examinados.

Palavras-chave: Simulações. Psicometria. Estudo da confiabilidade de software. Modelos matemáticos.

ABSTRACT

Currently, important decisions such as the planning of public policies in education and selection of tenders, are taken from results obtained in some assessments. Because of such importance, the instruments used to obtain the results should be valid and accurate. In order, the Item Response Theory (IRT) which is a set of mathematical models that seek to represent the probability of an individual to give some response to an item, depending on the characteristics of the item and examined the latent trait, has been developed. Currently the model that is most used is the logistic model that have three-parameters for dichotomous items and one-dimensional. This model has three parameters of the items that are: discrimination (a), difficulty (b), hit random individuals with low ability (c), and examined the abilities (θ). To estimate these parameters, there are specific software, free and commercials. The ICL (Item response theory Command Language) is a free software that makes parameter estimation of IRT models. It was developed by Alvin Bradley Hanson [Hanson, 2002] and uses the EM algorithm to make its estimates. In this work, simulations were performed by checking the behavior of the software due to increased number of items and examined, and greater variation of the parameter of discrimination and difficulty. The results show that the ICL, in the analyzed situations, produces estimates close to the reals for the parameters, and the best results are obtained in the estimation of the skills examined.

Keywords: Simulations. Psychometrics. Study of software reliability. Mathematical models.

LISTA DE FIGURAS

Figura 1	Regressão de X_j em θ_i , para três níveis de competência. Corte em k_j , mostrando que as áreas hachuradas acima da reta $X_j = k_j$ correspondem às probabilidades de respostas corretas dadas os respectivos níveis de habilidade.....	26
Figura 2	Curva Característica do Item (CCI)	28
Figura 3	Gráfico das Funções Distribuições Logística (azul) e Normal (vermelha) sendo utilizado 1 como parâmetro de escala. Verificar letra diferente dentro da figura.....	30
Figura 4	Gráfico das Funções Distribuições Logística (azul) e Normal (vermelha) sendo utilizado 1,702 como parâmetro de escala.....	31
Figura 5	Curvas Características dos Itens com diferentes valores de b	33
Figura 6	Curvas Características de Cinco Itens com Diferentes Valores de a e b.....	35
Figura 7	Curva Característica do Item do Modelo Logístico de 3 Parâmetros.....	39
Figura 8	Curvas Características dos Itens e Curvas de Informação dos Itens de Modelos Logísticos de 1 Parâmetro.....	41
Figura 9	Curvas Características dos Itens e Curvas de Informação dos Itens de Modelos Logísticos de 2 Parâmetros.....	42
Figura 10	Característica do Item e Curva de Informação do Item de Modelos Logísticos de 3 Parâmetros.....	43
Figura 11	Gráfico do viés e erro quadrático médio do parâmetro a, sendo que os símbolos representam respectivamente: • 40 itens, Δ 80 itens e + 120 itens.....	69

Figura 12	Gráfico do viés e do erro quadrático médio para o parâmetro b , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	70
Figura 13	Gráfico do viés e erro quadrático médio para o parâmetro c , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	71
Figura 14	Gráfico do viés e do erro quadrático médio para o parâmetro θ , estimado por verossimilhança, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados.....	72
Figura 15	Gráfico do viés e do erro quadrático médio para o parâmetro θ pelo método Bayesiano, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados.....	73
Figura 16	Gráfico do viés e erro quadrático médio do parâmetro a , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	77
Figura 17	Gráfico do viés e do erro quadrático médio para o parâmetro b , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	78
Figura 18	Gráfico do viés e do erro quadrático médio para o parâmetro c , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	79
Figura 19	Gráfico do viés e erro quadrático médio do parâmetro θ estimado por verossimilhança, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados.....	80
Figura 20	Gráfico do viés e do erro quadrático médio para o parâmetro	

	θ estimado pelo método Bayesiano, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados.....	81
Figura 21	Gráfico do Viés e do Erro Quadrático Médio para o parâmetro a , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	84
Figura 22	Gráfico do Viés e do Erro Quadrático Médio para o parâmetro b , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	85
Figura 23	Gráficos do viés e do erro quadrático médio para o parâmetro c , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens.....	86
Figura 24	Gráfico do viés e erro quadrático médio para o parâmetro $\hat{\theta}$ estimado por verossimilhança, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados.....	87
Figura 25	Gráfico do viés e do erro quadrático médio para o parâmetro $\hat{\theta}$ Estimado pelo método Bayesiano, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados.....	88

LISTA DE TABELAS

Tabela 1	Exemplo Contendo Valores Sorteados de Parâmetros de 20 Itens.....	60
Tabela 2	Exemplo de Habilidades Sorteadas de 15 Examinados.....	61
Tabela 3	Tabela de respostas de 20 Examinados a 20 Itens.....	62
Tabela 4	Tabela com esquema de análise dos fatores, sendo que S1, S2 e S3 representam as 3 situações analisadas.....	65
Tabela 5	Resumo dos Resultados da Situação 1.....	75
Tabela 6	Comparação entre o viés e a amplitude dos valores sorteados na situação 1.....	76
Tabela 7	Resumo da Situação 2.....	82
Tabela 8	Comparação entre o viés e a amplitude dos valores sorteados na situação 2.....	83
Tabela 9	Resumo da situação 3.....	89
Tabela 10	Comparação entre o viés e a amplitude dos valores sorteados.....	90
Tabela 11	Vieses encontrados para os parâmetros a, b, c e θ nas 3 situações estudadas.....	91

SUMÁRIO

1	INTRODUÇÃO.....	15
2	REFERENCIAL TEÓRICO.....	18
2.1	Competência e habilidade.....	18
2.2	Dimensionalidade de um teste.....	19
2.3	Parâmetros dos itens.....	19
2.4	Classificações das respostas de um examinado a um item.....	21
2.5	Modelos de teoria de resposta ao item (TRI).....	22
2.6	Modelo de itens dicotômicos, unidimensionais para uma única população.....	23
2.6.1	Modelo da ogiva normal de 2 parâmetros.....	23
2.6.1.1	Justificativa do modelo usando a ogiva normal de 2 parâmetros.....	25
2.6.2	Gráficos da TRI.....	28
2.6.3	Modelo com a ogiva normal e modelo logístico.....	29
2.6.4	Modelos logísticos de 1 , 2 e 3 parâmetros.....	31
2.6.4.1	Modelo logístico de 1 parâmetro ou modelo de rash.....	32
2.6.4.2	Modelo logístico de dois parâmetros.....	34
2.6.4.3	Modelo logístico de 3 parâmetros.....	37
2.7	Função informação do item	39
2.7.1	Modelo de 1 parâmetro.....	40
2.7.2	Modelo de 2 parâmetros.....	40
2.7.3	Modelo logístico de 3 parâmetros.....	41
2.8	Estimação de parâmetros.....	44
2.8.1	Estimação por máxima verossimilhança conjunta.....	44
2.8.1.1	Estimação dos parâmetros dos itens quando as habilidades são conhecidas (ou supostamente conhecidas).....	44
2.8.1.1.2	Algoritmo Newton-Raphson.....	48
2.8.1.1.3	Estimação das habilidades dos examinados quando os parâmetros dos itens são conhecidos.....	53
2.8.1.1.4	Algoritmo de Newton-Raphson.....	56
2.8.2	Algoritmo EM utilizado no ICL.....	56
2.8.3	Sobre o ICL.....	57
3	METODOLOGIA.....	59
3.1	Software utilizado para realizar as simulações.....	63
3.2	Situações analisadas.....	64

4	RESULTADOS E DISCUSSÃO.....	68
4.1	Resultados da situação 1.....	68
4.2	Resultados da situação 2.....	77
4.3	Resultados da situação 3.....	84
5	CONCLUSÃO.....	92
	REFERÊNCIAS.....	93
	ANEXOS.....	95

1 INTRODUÇÃO

O desenvolvimento de avaliações que refletem de forma fidedigna o resultado de uma aprendizagem é algo que preocupa pedagogos, psicólogos educacionais e educadores em geral. Atualmente, várias decisões são tomadas a partir de avaliações em larga escala como: o planejamento de políticas públicas educacionais e a seleção de candidatos para ocupar vagas em universidades e no serviço público. Devido à importância das decisões tomadas a partir de resultados avaliativos, os instrumentos utilizados para a obtenção desses resultados devem ser válidos e precisos.

Dessa forma, é grande o interesse no desenvolvimento e aplicação de métodos que medem, com precisão, características humanas. Esse interesse é historicamente antigo: desde o século XIX, temos registros de testes com objetivo de medir a inteligência. A Teoria Clássica dos Testes (TCT), teoria utilizada para representar o comportamento como uma categoria ou número, foi sendo aos poucos desenvolvida, até que em 1950, já estava bem axiomatizada. Porém, tal teoria apresentava alguns problemas e limitações, como a dificuldade de se comparar o desempenho de indivíduos utilizando testes diferentes e o resultado do teste não ser uma medida absoluta, pois varia de teste para teste (PASQUALI, 2003).

Devido às limitações da TCT, aos poucos foi sendo desenvolvida a Teoria de Resposta ao Item (TRI). A TRI, também conhecida como Teoria da Curva Característica ou Teoria do Traço Latente, que é um conjunto de modelos matemáticos que considera o item como unidade básica de análise e procura representar a probabilidade de um indivíduo dar certa resposta a um item em função dos parâmetros do item e do(s) traço(s) latente(s) do indivíduo (ANDRADE, TAVARES & VALLE, 2000).

Vários autores contribuíram para seu desenvolvimento, sendo o seu responsável mais direto Frederic Lord (1968,1980), porém não podemos deixar de citar Birnbaum (1968) e Rash (1960), devido as suas importantes contribuições para essa Teoria. É importante ressaltar que a TRI, apesar de já estar bem desenvolvida por volta de 1968, só foi efetivamente aplicada por volta de 1980 devido à complexidade dos recursos computacionais necessários.

Em comparação com a TCT, podem ser citadas como vantagens principais da TRI a possibilidade de conhecer a característica das questões, identificando as que realmente contribuem para a avaliação da aprendizagem; o acompanhamento do desenvolvimento de um indivíduo ao longo do tempo; a comparação de resultados obtidos com diferentes indivíduos (BAKER, 2001).

No Brasil, a Teoria de Resposta ao Item passou a ser utilizada pelo Exame Nacional do Ensino Médio (ENEM) do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) do Ministério da Educação (MEC) após 2009, o ENEM é uma das principais avaliações em larga escala do país, sendo seus resultados utilizados para o ingresso de estudantes em diversas instituições de ensino superior. No entanto, a TRI no Brasil já vinha sendo utilizada desde 1995, primeiramente pela pesquisa AVEJU, da Secretaria de Estado da Educação de São Paulo, a partir de então, outras avaliações em larga escala passaram a utilizá-la, como: o Sistema de Avaliação do Rendimento Escolar do Estado de São Paulo (SARESP), Sistema de Avaliação da Educação Básica (SAEB) do INEP/MEC e Sistema Mineiro de Avaliação da Educação Pública (SIMAVE) da Secretaria de Estado de Educação de Minas GERAIS (MOREIRA JUNIOR, 2010).

Internacionalmente, a TRI tem sido utilizada por importantes avaliações como no PISA (Programme for International Student Assessment), realizado pela Organização para a Cooperação e o Desenvolvimento Econômico (OECD); o LAMP (Literacy Assessment and Monitoring), iniciativa do Instituto de

Estatística da UNESCO; o Estudo Regional Comparativo e Explicativo, realizado pelo Laboratório Latino-americano de Avaliação da Qualidade da Educação (LLECE), da UNESCO (ANDIFES, 2010); o teste TOEFL (Test of English as a Foreign Language) e o teste GRE (Graduate Record Examination), aplicados via testagem adaptativa por computador (computerized adaptive testing - CAT) pelo Educational Testing Service – ETS .

Atualmente existem diversos softwares que realizam a estimação dos parâmetros da TRI, porém, grande parte não é gratuita. Um dos softwares disponíveis atualmente é o ICL (*Item response theory Command Language*) desenvolvido por Bradley Alvin Hanson [Hanson, 2002]. Hanson, autor do software, trabalhou na área de avaliação educacional e psicometria, tendo desenvolvido vários softwares nesta área, dos quais vários estão disponíveis livremente na internet. Devido à sua brilhante carreira e prematura morte, anualmente o Conselho Nacional de Mensuração em Educação dos Estados Unidos, criou o prêmio Bradley Hanson, que é apresentado anualmente e premia, os melhores trabalhos com avanços na área de avaliação educacional.

O ICL é um software livre, rápido e de fácil manipulação, tal software utiliza o algoritmo EM para a estimação dos parâmetros dos itens e habilidades dos examinados dos modelos logísticos de 1,2 e 3 parâmetros para itens dicotômicos e modelos de crédito parcial e generalizados para itens politômicos. Existem versões do ICL disponíveis para Windows, Macintosh e Linux, este trabalho terá como objetivo estudar a confiabilidade e o comportamento das estimações feitas pelo ICL.

2 REFERENCIAL TEÓRICO

2.1 Competência e habilidade

Na área da educação, a TRI é uma metodologia que tem por objetivos estimar a(s) habilidade(s) de um indivíduo, em uma área de conhecimento avaliada e as características dos itens consideradas relevantes, ou seja, que podem interferir na resposta dada por determinado examinado a um item.

A habilidade é uma variável latente, ou seja, uma variável que não pode ser medida diretamente, diferentemente de variáveis como: peso, altura, temperatura, etc. Por isso, variáveis como: ansiedade, satisfação, inteligência, conhecimento, que não são diretamente aferidas, são classificadas como latentes. Esse tipo de variável é aferida, a partir de variáveis secundárias observáveis relacionadas a ela, no caso da competência, a variável secundária observada é a resposta dada pelo examinado a um item.

A TRI propõe modelos para variáveis latentes e é atualmente aplicada em diversas áreas como: educação, qualidade de vida, psiquiatria, psicologia, satisfação do consumidor e várias outras. A TRI tem como unidade básica o item, ou seja, cada questão de um teste, sendo então o teste, um conjunto de itens.

Na literatura, a variável latente que está sendo medido através da TRI é chamado de diversas maneiras como: traço latente, variável latente, traço, habilidade, habilidade latente, dentre outras. Neste trabalho, será usado o termo habilidade conforme o Guia de Elaboração e Revisão de Itens, volume 1 do INEP.

A habilidade que está sendo medida será representada por θ e o termo, examinado, usado para designar o indivíduo avaliado. É importante ressaltar que

a habilidade não deve ser considerada uma característica inata do indivíduo, sendo que ela pode mudar ao longo do tempo, através da instrução do indivíduo (HAMBLETON, 1991).

2.2 Dimensionalidade de um teste

Considerando o conjunto de todas m habilidades que afetam a resposta do(s) examinado(s) em, pelo o menos, um item de um teste de J itens, teremos o vetor:

$$\Theta = (\Theta_1, \Theta_2, \dots, \Theta_m)^t$$

Lord (1968) chama esse vetor de espaço latente completo, onde a habilidade de cada examinado é um ponto desse espaço latente. A dimensão do teste será então m , ou seja, a dimensão do espaço latente completo.

Um teste é chamado unidimensional se $m=1$, dessa forma existe uma única habilidade afetando a resposta do examinado. Neste caso; $\Theta = (\Theta_1)$. Para simplificação de notação, será usado o símbolo Θ para representar um espaço latente unidimensional e o símbolo θ_{ij} para representar um valor assumido pela variável Θ .

2.3 Parâmetros dos itens

Além da habilidade, a TRI também estima os parâmetros dos itens que são características do item, tais parâmetros são a_j , b_j , c_j , sendo:

- 1) b_j é o parâmetro de dificuldade do item j .
- 2) a_j é o parâmetro de discriminação do item j .
- 3) c_j é o parâmetro da probabilidade de acerto casual do item j .

Associado a cada item j , teremos um vetor de parâmetros δ_j . Dessa forma: $\delta_j = (a_j, b_j, c_j)$.

Assim, Δ será o conjunto de todos os vetores de parâmetros dos J itens, ou seja:

$$\Delta = (\delta_1, \delta_2, \dots, \delta_j, \dots, \delta_J), \text{ ou na forma matricial, } \Delta = \begin{bmatrix} a_1 & a_2 & \dots & a_j & \dots & a_J \\ b_1 & b_2 & \dots & b_j & \dots & b_J \\ c_1 & c_2 & \dots & c_j & \dots & c_J \end{bmatrix}$$

Segundo Baker (2001), os parâmetros dos itens são invariantes em uma população de examinados. Essa é uma característica muito importante da TRI, pois isso significa que, não importa qual é a média da habilidade do grupo, os parâmetros dos itens estimados serão os mesmos, ou seja, são independentes da habilidade.

Esta é a principal vantagem da TRI, frente à TCT, pois nesta última, os parâmetros dos itens estimados dependem da população que está respondendo ao teste. Se por exemplo, aplicarmos o mesmo item a um grupo de uma população com baixa habilidade média e a outro grupo da mesma população, com alta habilidade média, obteremos dois valores diferentes para o parâmetro dificuldade. Usando a primeira população, com alta habilidade, teremos um valor do parâmetro de dificuldade do item estimado baixo, enquanto para a segunda população o valor estimado desse mesmo parâmetro para o mesmo item será alto.

O mesmo ocorre com o parâmetro de discriminação, na TCT ele varia de acordo com a homogeneidade do grupo. Se tomarmos um grupo homogêneo, a discriminação estimada será alta, enquanto em um grupo um pouco mais heterogêneo, a discriminação estimada será baixa.

Devido à invariância dos parâmetros dos itens, e sua independência com a habilidade, com a TRI pode-se:

- a) Comparar um único grupo fazendo uma única prova.
- b) Comparar um único grupo, dividido em dois subgrupos, fazendo duas provas, totalmente distintas (nenhum item comum).
- c) Comparar um único grupo, dividido em dois subgrupos, fazendo duas provas, parcialmente distintas.
- d) Comparar dois grupos fazendo uma única prova.
- e) Comparar dois grupos fazendo duas provas, parcialmente distintas.
- f) Acompanhar a evolução de um examinado ou de um grupo de examinado ao longo do tempo.
- g) Avaliar corretamente os itens. Por exemplo: o parâmetro de dificuldade de um item estimado pela TRI será sempre o mesmo, independente da habilidade do grupo de examinados. Diferentemente da TCT, onde a dificuldade estimada de um item depende da habilidade do grupo de examinados. Se um grupo de examinados, com alta habilidade, responder a um item, a dificuldade estimada pode ser baixa enquanto que, se o mesmo item for respondido por outro grupo, com habilidade menor, a dificuldade estimada poderá ser maior.
- h) Estimar a habilidade independentemente dos parâmetros dos itens. Enquanto na TCT a nota do examinado depende e varia de acordo com a dificuldade do teste (mais fácil ou mais difícil), na TRI, a habilidade estimada é sempre a mesma independentemente da dificuldade do teste.

2.4 Classificações das respostas de um examinado a um item

De acordo com Bragion (2010), a resposta dada por um examinado a um item pode ser classificada como:

- a) Dicotômica:** Nesse caso, as respostas são classificadas apenas em certas ou erradas.

b) Politômica: Além das classificações certas ou erradas, são atribuídas probabilidades para outras categorias de resposta.

c) Contínua: São utilizados valores dentro de um intervalo de números reais para classificar a resposta. Utilizada em questões abertas.

É importante ressaltar, que não apenas os testes de múltipla escolha são dicotômicos, as questões abertas também podem ser dicotômicas, basta que sejam corrigidas como certas ou erradas. Da mesma forma, os testes de múltipla escolha, também podem ser politômicos.

2.5 Modelos de Teoria de Resposta ao Item (TRI)

Na literatura, existem vários modelos de teoria de resposta ao item, tais modelos dependem de três fatores: (Andrade, Tavares & Valle, 2000)

- i- da natureza do item: dicotômicos ou não dicotômicos (politômicos ou contínuo)
- ii- do número de populações (ou grupos) envolvidas: apenas uma ou mais de uma
- iii- da quantidade de habilidades que estão sendo medidas: apenas uma ou mais de uma.

Neste trabalho, somente serão considerados itens dicotômicos, unidimensionais, para uma única população.

Difícilmente tem-se apenas uma única habilidade sendo medida em um item, por isso muitas vezes é admitido que um teste é unidimensional se houver uma habilidade dominante afetando a resposta do examinado. Essa consideração deve ser feita observando a população, pois um teste pode ser unidimensional para uma população e multidimensional para outra.

Outro conceito importante é o de uma ou mais populações (ou grupos) envolvidas, esse conceito deve levar em conta se, as características da população

são diferentes com relação à habilidade a ser estimada. Por exemplo, se a habilidade considerada for a criatividade, alunos do 5º e 6º ano poderão ser tomados como uma única população. Já se a habilidade pesquisada for gramática, pode ser considerado que estas mesmas duas populações são distintas. Neste trabalho, um grupo será considerado uma amostra de uma população obtida através do processo de amostragem aleatória simples.

2.6 Modelo de itens dicotômicos, unidimensionais para uma única população

Esse tipo de modelo possui as seguintes suposições:

- a) **Unidimensionalidade:** A unidimensionalidade é a suposição de que o espaço latente completo do item é unidimensional, ou seja, a resposta dada pelo examinado depende de uma única competência e das características do item consideradas pelo modelo. Geralmente a dimensionalidade do teste é verificada através de análise fatorial.

- b) **Independência Local:** Essa suposição assume que para uma dada habilidade, as respostas aos diferentes itens de um teste são independentes. Isso significa que a resposta de um determinado item, em um teste, não pode influenciar a resposta de outros itens.

2.6.1 Modelo da ogiva normal de 2 parâmetros

Seja Y_{ij} a variável aleatória associada ao acerto do indivíduo i ao item j

sendo:

$i=1, \dots, n$ (examinados) e $j=1, \dots, J$ (itens)

Em itens dicotômicos, a resposta do examinado é classificada em certa ou errada, o valor 1 será associado a respostas corretas e o valor 0, a respostas erradas. A resposta do examinado i ao item j está condicionalmente relacionada à sua competência e aos parâmetros do item, ou seja, à medida que a competência aumenta, esperamos ter um aumento na probabilidade desse indivíduo acertar o item, dessa forma:

- a) $Y_{ij}=1$ se a resposta for correta
- b) $Y_{ij}=0$ se a resposta for errada

Assim, a variável $Y_{ij} \sim \text{Bernoulli}(\pi_{ij})$, sendo sua distribuição de probabilidade dada por:

$$f(Y_{ij} = y_{ij} | \Theta = \theta_i; \delta_j) = \pi_{ij}^{y_{ij}} (1 - \pi_{ij})^{1-y_{ij}}$$

$i=1, \dots, n$ (examinados) e $j= 1, \dots, J$ (itens).

Em que:

- a) π_{ij} é a probabilidade do indivíduo i responder corretamente o item j ;
- b) Θ representa a variável aleatória habilidade;
- c) θ_i é a habilidade do indivíduo i ;
- d) δ_j é o vetor de parâmetros do item j .

Se $y_{ij}=1$, têm-se que:

$$f(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = \pi_{ij}^1 (1 - \pi_{ij})^{1-1} = \pi_{ij}$$

Se $y_{ij}=0$, têm-se que:

$$f(Y_{ij} = 0 | \Theta = \theta_i; \delta_j) = \pi_{ij}^0 (1 - \pi_{ij})^{1-0} = 1 - \pi_{ij}$$

Por definição:

$$\pi_{ij} = f(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j)$$

Segundo Lord (1968), Lawley (1943, 1944) foi o primeiro a desenvolver um modelo usando a ogiva normal para mais de dois itens. Este modelo é dado por:

$$\pi_{ij} = f(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = \int_{-\infty}^{\alpha_j(\theta_i - b_j)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

2.6.1.1 Justificativa do modelo usando a ogiva normal de 2 parâmetros

Em seu trabalho, Lord (1968) supõe que para cada item j , existe uma variável latente X_j que determina a *performance* do examinado no item. De tal forma que, dada uma constante κ_j teremos duas situações:

1. Se $X_j \geq \kappa_j$, então o examinado responderá corretamente o item j ($Y_{ij} = 1$).
2. Se $X_j < \kappa_j$, então o examinado não responderá o item corretamente ($Y_{ij} = 0$).

Supondo que:

- a) A regressão de X_j em θ_i é linear.
- b) A variância da regressão é a mesma para todo θ_i , ou seja, é homocedástica.
- c) A distribuição condicional de X_j dado θ é normal com média $\mu_{j|\theta_i}$ e variância $\sigma_{j|\theta_i}^2$.

A regressão de X_j em θ_i está representada na figura seguinte.

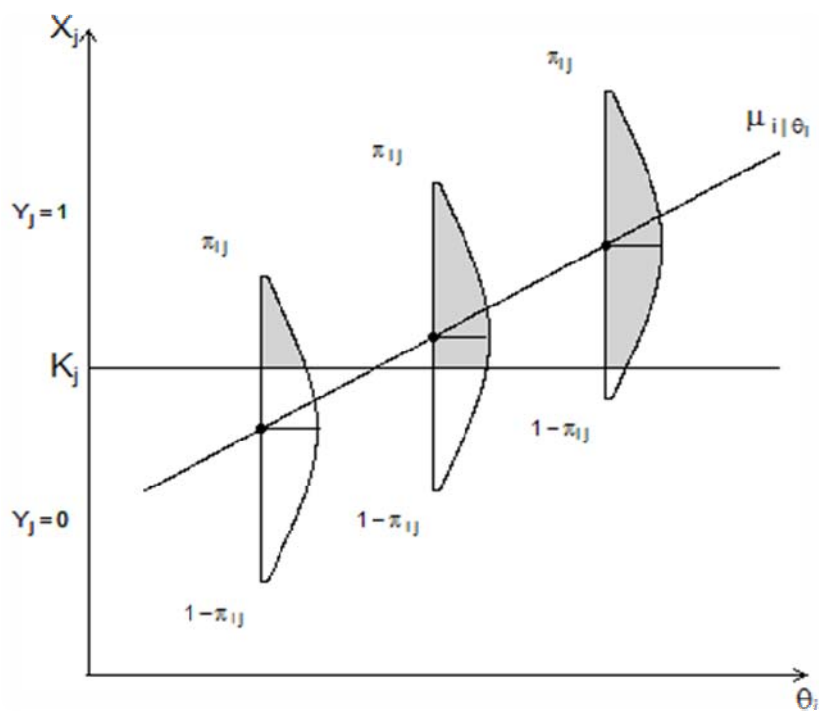


Figura 1 Regressão de X_j em θ_i , para três níveis de competência. Corte em k_j , mostrando que as áreas hachuradas acima da reta $X_j = k_j$ correspondem às probabilidades de respostas corretas dadas os respectivos níveis de habilidade

Observa-se que a área do gráfico que representa a probabilidade de resposta correta para todos os níveis de habilidade é o somatório de todas as áreas que estão acima da reta $X_j = k_j$. Ou seja:

$$P(Y_{ij} = 1 | \Theta = \theta_i; \delta_i) = P(X_j > k_j | \Theta = \theta_i) = P(Z_{j|\Theta=\theta_i} > \frac{k_j - \mu_{j|\theta_i}}{\sigma_{j|\theta_i}}) = P(Z_{j|\Theta=\theta_i} > -L)$$

$$= P(Z_{j|\Theta=\theta_i} < L) = \Phi(L) = \Phi(\alpha_j(b_j - \theta_i)) = \int_{-\infty}^{\alpha_j(\theta_i - b_j)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

(***)

(*) Padronização da variável latente X_j

$$(**) \frac{k_j - \mu_{j|\theta_i}}{\sigma_{j|\theta_i}} (1) = -L$$

Sendo $\mu_{j|\theta_i} = \beta_1 \theta_i$, da mesma forma que a variável X_j foi padronizada, a variável Θ também poderá ser padronizada, ou seja, ambas terão média 0 e desvio padrão 1.

Como ambas variáveis possuem o desvio padrão igual a 1, têm-se que $\beta_1 = \rho(X_j, \Theta) = \rho_j$. Como:

$$\text{Var}[X_j, \theta_i] = \sigma_{X_j}^2 (1 - \rho_j), \text{ sendo } \sigma_{X_j}^2 = 1$$

$$\text{Var}[X_j, \theta_i] = 1(1 - \rho_j) = (1 - \rho_j)$$

Pode-se reescrever a equação (1) de forma que:

$$X_j = \frac{k_j - \mu_{j|\theta_i}}{\sigma_{j|\theta_i}} = \frac{k_j - \rho_j \theta_i}{\sqrt{(1 - \rho_j^2)}} = \frac{\frac{k_j - \rho_j \theta_i}{\rho_j}}{\frac{\sqrt{(1 - \rho_j^2)}}{\rho_j}} = \left(\frac{k_j}{\rho_j} - \theta_i \right) \left(\frac{\rho_j}{\sqrt{(1 - \rho_j^2)}} \right)$$

Fazendo $\alpha_j = \frac{\rho_j}{\sqrt{(1 - \rho_j^2)}}$ e $b_j = \frac{k_j}{\rho_j}$, obtêm-se a equação:

$$(b_j - \theta_i) \alpha_j = \alpha_j (b_j - \theta_i) = -L. \text{ Dessa forma;}$$

$$-L = \alpha_j (b_j - \theta_i)$$

$$-L = \alpha_j (-(\theta_i - b_j))$$

$$L = \alpha_j (\theta_i - b_j)$$

(***) Devido à simetria da distribuição normal:

$$\int_{-L}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \int_{-\infty}^L \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

2.6.2 Gráficos da TRI

A regressão da probabilidade de acerto de um item em função da habilidade, para itens binários (respondidos dicotomicamente) é chamada de Curva Característica do Item (CCI). A CCI é estritamente crescente, sendo que seu crescimento ocorre de maneira não linear.

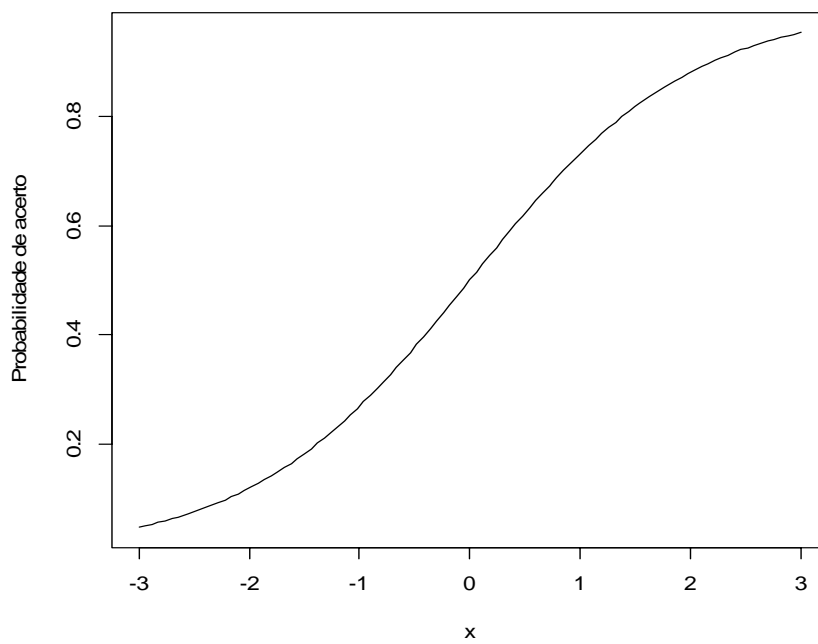


Figura 2 Curva Característica do Item (CCI)

2.6.3 Modelo com a ogiva normal e modelo logístico

Birnbaum (1968) foi o primeiro a usar o modelo logístico para substituir o modelo com a ogiva normal. Tal substituição foi realizada, devido à simplicidade do modelo logístico (ou Função Distribuição Logística) por ter uma integral explícita. A seguir estão as funções do modelo logístico e a normal acumulada.

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \quad (\text{Modelo com a Ogiva Normal})$$

$$\Psi(x) = \frac{e^x}{(1+e^x)} = \frac{\frac{e^x}{e^x}}{\frac{1+e^x}{e^x}} = \frac{1}{\frac{1}{e^x} + 1} = \frac{1}{1+e^{-x}} \quad (\text{Modelo Logístico})$$

Birnbaum (1968) afirma que o modelo logístico pode ser usado para substituir a função $\Phi(x)$, pois ele quase coincide com a normal, sendo:

$$|\Phi(x) - \Psi[(1,702)x]| < 0,01; \quad -\infty < x < +\infty$$

O valor 1,702 será representado por D e chamado de parâmetro de escala, ou seja, o valor que melhor aproxima o gráfico da função distribuição acumulada logística da ogiva normal.

Abaixo estão dois gráficos, cada um tem duas curvas, a função distribuição normal e logística. Observe que no segundo, os dois gráficos praticamente coincidem, isso acontece quando usamos o parâmetro de escala D = 1,702. No primeiro gráfico D=1 e as curvas ficam bem mais distantes.

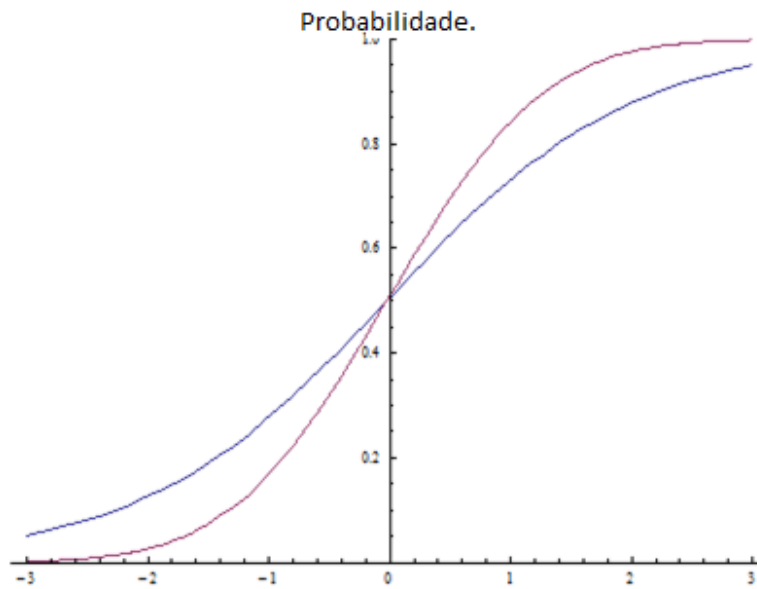


Figura 3 Gráfico das Funções Distribuições Logística (azul) e Normal (vermelha) sendo utilizado 1 como parâmetro de escala

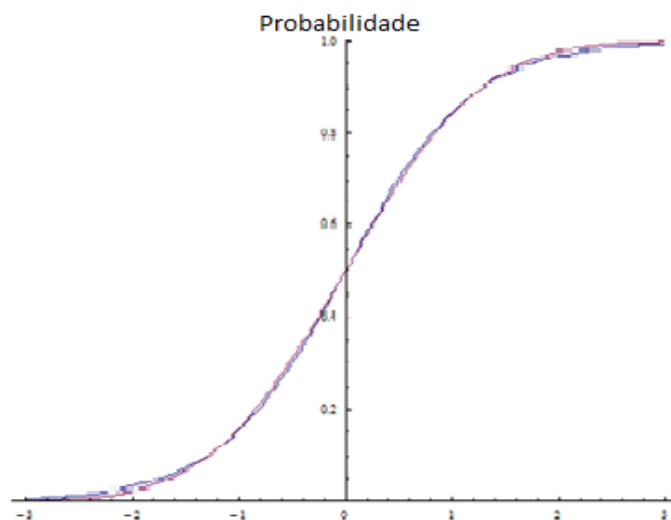


Figura 4 Gráfico das Funções Distribuições Logística (azul) e Normal (vermelha) sendo utilizado 1,702 como parâmetro de escala

Em seu trabalho, Birnbaum (1968) faz menção ao trabalho de Halley (1952), onde há a demonstração de que, quando o parâmetro de escala vale 1,702, temos a melhor aproximação da função distribuição logística com relação à ogiva normal. Tal demonstração encontra-se no Apêndice A deste trabalho.

2.6.4 Modelos logísticos de 1 , 2 e 3 parâmetros

Definiu-se anteriormente que $\delta_j = (a_j, b_j, c_j)'$, é um vetor com os parâmetros dos itens, na literatura os principais modelos para itens dicotômicos, unidimensionais para uma única população (ou grupo) usando estes parâmetros são:

2.6.4.1 Modelo logístico de 1 parâmetro ou modelo de rash

O modelo é dado por:

$$P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = \frac{1}{1 + e^{-D(\theta_i - b_j)}}$$

$i=1 \dots n$ (examinados), $j= 1, \dots, J$ (itens), $\delta_j = (1, b_j, 0)$, $(-\infty < \theta_i < +\infty)$
e $(-\infty < b_j < +\infty)$

Sendo:

- a) Y_{ij} = variável aleatória associada ao acerto ou erro da resposta do indivíduo i ao item j , podendo assumir os valores 0 ou 1.
- b) $P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j)$ é a probabilidade do indivíduo i com habilidade θ_i responder corretamente ao item j .
- c) θ_i é o parâmetro da habilidade do indivíduo i
- d) b_j é o parâmetro de dificuldade do item j .
- e) D 1,702 é o parâmetro de escala.
- f) δ_j o vetor de parâmetros do item j .

Neste modelo, o único parâmetro do item que varia é o parâmetro b , chamado de parâmetro de dificuldade. Tal parâmetro deve estar na mesma escala da habilidade e pode assumir qualquer valor real $(-\infty < b < +\infty)$. A associação entre a dificuldade do item e o parâmetro b é claramente observada no gráfico abaixo.

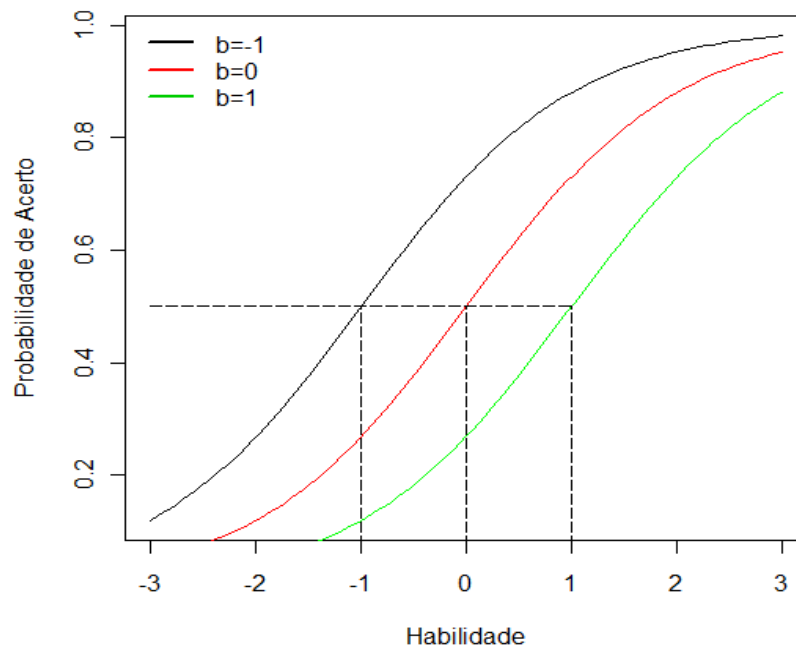


Figura 5 Curvas Características dos Itens com diferentes valores de b

Observa-se que os três gráficos têm a mesma forma, apenas estão deslocados à direita com relação ao gráfico em preto. Isso ocorre, devido à variação do parâmetro de dificuldade. Observa-se também que sempre que a habilidade θ_i é igual ao parâmetro de dificuldade do item ($\theta_i = b_i$), têm-se uma probabilidade de acerto do item de 0,5, ou seja, 50%.

Verificação:

$$\frac{1}{1 + e^{-1,702(\theta_i - b_i)}} = 0,5$$

$$1 = 0,5(1 + e^{-1,702(\theta_i - b_i)})$$

$$2 = 1 + e^{-1,702(\theta_i - b_i)}$$

$$1 = e^{-1,702(\theta_i - b_j)}$$

Para que esta igualdade seja verdadeira: $-1,702(\theta_i - b_j) = 0$, ou seja, $\theta_i = b_j$.

Por isso, conclui-se que o item em preto é mais fácil que o em vermelho, e que ambos, são mais fáceis que o item em verde, pois para atingirmos uma probabilidade de acerto de 50% em todas as curvas, precisamos de habilidades mais altas. Para o item em preto, esta probabilidade é atingida quando o nível de habilidade vale -1, no vermelho quando a habilidade vale 0 e, no verde, quando a habilidade vale 1.

Mais detalhes deste modelo encontram-se no apêndice B deste trabalho.

2.6.4.2 modelo logístico de dois parâmetros

O modelo é dado por:

$$P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = \frac{1}{1 + e^{B(-a(\theta_i - b_j))}}$$

$i=1, \dots, n$ (examinados), $j=1, \dots, J$ (itens), $\delta_j = (a_j, b_j, 0)$,
 $(-\infty < \theta_i < +\infty)$, $(-\infty < b_j < +\infty)$ e $(a > 0)$

Sendo:

- Y_{ij} = variável aleatória associado ao acerto ou erro na resposta do indivíduo i ao item j , podendo assumir os valores 0 ou 1.
- $P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j)$ é a probabilidade do indivíduo i com habilidade θ_i responder corretamente ao item j .
- θ_i é o parâmetro da habilidade do indivíduo i
- δ_j é o vetor de parâmetros do item j .
- b_j é o parâmetro de dificuldade do item j .

- f) a_j é o parâmetro de discriminação do item j.
- g) $D = 1,702$ é o parâmetro de escala.

Observa-se que esse modelo difere do modelo logístico de um parâmetro, pelo fato de ser considerado o parâmetro de discriminação do item, denotado por a ($a > 0$), este parâmetro é proporcional à inclinação da reta tangente à curva no ponto de inflexão (quando $\theta_i = b_j$). A demonstração desta afirmação encontra-se no apêndice C.

Tem-se que a inclinação da curva é máxima no ponto de inflexão e neste ponto a inclinação da reta tangente é igual a $0,4255a_j$. Assim, quanto maior a_j , maior a inclinação da reta tangente e maior também a inclinação da curva, em outras palavras, quanto maior o poder de discriminação de um item, maior será a inclinação de sua curva característica. Observando-se as CCIs, com diferentes valores do parâmetro de discriminação, ficará mais fácil o entendimento do que significa o aumento do valor do parâmetro de discriminação.

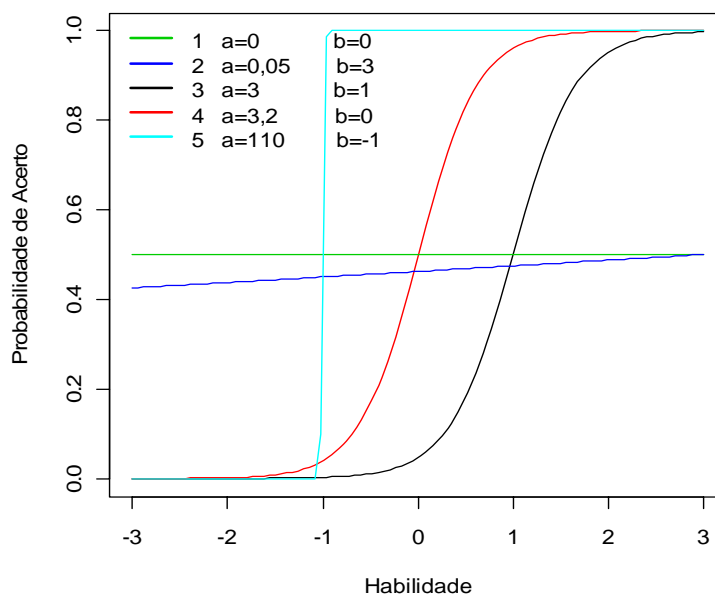


Figura 6 Curvas Características de Cinco Itens com Diferentes Valores de a e b

Neste modelo, o parâmetro a é definido de forma que $a > 0$, porém no gráfico 1 considerou-se hipoteticamente que $a = 0$, apenas para observar-se o que aconteceria se este parâmetro assumisse tal valor. No gráfico 1, onde o parâmetro a vale zero, tem-se uma reta paralela ao eixo x. Isso significa que não importa a habilidade, a probabilidade de acerto do item é sempre a mesma de 50%. Conclui-se, então, que este item não discrimina examinados com diferentes níveis de habilidade.

Já no gráfico 2, em que o parâmetro $a = 0,05$, nota-se que o gráfico ainda se parece muito com uma reta, porém com uma pequena inclinação, ou seja, começa a haver uma associação de probabilidades maiores de acerto com níveis maiores de habilidade.

Nas curvas 3 e 4, observa-se que o valor do parâmetro de discriminação de ambas é muito próximo, 3 e 3,2 respectivamente. Verifica-se também, que as curvas são muito parecidas, a maior diferença é a curva 3 estar deslocado para a

direita, pois, seu parâmetro b é maior (vale 1 enquanto na curva 4 vale 0). Além disso, observa-se um rápido crescimento da probabilidade de acerto na vizinhança do ponto de inflexão, ou seja, em torno de b . A curva 3 discrimina muito bem indivíduos com habilidades entre: $0,5 < \theta_i < 1,5$, já curva 4 discrimina muito bem indivíduos com habilidades entre:

- $0,5 < \theta_i < 0,5$. Para os demais valores a discriminação de ambas as curvas é pequena.

A curva 5, que possui valor de discriminação altíssimo ($\alpha = 110$), praticamente possui uma reta paralela ao eixo y . Nota-se que tal curva discrimina bem, apenas indivíduos em uma pequena faixa de habilidade, neste caso: $-0,9 < \theta_i < -1,1$, para outros valores de habilidade, a discriminação de examinados com diferentes habilidades é pequena neste item.

Lord (1968) considera que um teste, não deve apenas ser constituído por itens com alto valor do parâmetro α , tendo em vista que, seu poder de discriminação é limitado a um pequeno intervalo da habilidade. Ele recomenda que sejam usados, em sua maioria, itens com características semelhantes às dos itens 3 e 4, que possuem relevante poder de discriminação, para um maior intervalo de habilidades, principalmente porque quando inicialmente um teste é aplicado, não se sabe em que nível de habilidade estão os examinados. Baker (2001) classifica o parâmetro α da seguinte forma:

Muito Baixa: de 0,01 à 0,34

Baixa: de 0,35 à 0,64

Moderada: de 0,64 à 1,34

Alta: de 1,35 à 1,69

Muito Alta: > que 1,70

2.6.4.3 Modelo logístico de 3 parâmetros

Neste modelo, além dos parâmetros descritos anteriormente, também é utilizado um parâmetro para representar o acerto casual do item pelos examinados de baixa habilidade, denotado por c_i . Birnbaum (1968) foi quem introduziu este parâmetro ao modelo, considerando o fato de que examinados de baixa habilidade, às vezes, dão corretas respostas para os itens.

O modelo é dado por:

$$P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = c_i + (1 - c_i) \left(\frac{1}{1 + e^{-D(\theta_i - b_j)}} \right)$$

$i=1, \dots, n$ (examinados), $j= 1, \dots, J$ (itens), $\delta_j = (a_j, b_j, c_j)$,

$(-\infty < \theta_i < +\infty)$, $(a_j > 0)$, $(-\infty < b_j < +\infty)$ e $(0 < c_j < 1)$

Sendo:

- Y_{ij} = variável aleatória associado ao acerto ou erro na resposta do indivíduo i ao item j , podendo assumir os valores 0 ou 1.
- $P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j)$ é a probabilidade do indivíduo i com habilidade θ_i responder corretamente ao item j .
- θ_i é o parâmetro da habilidade do indivíduo i
- δ_j é o vetor de parâmetros do item j .
- b_j é o parâmetro de dificuldade do item j .
- a_j é o parâmetro de discriminação do item j .
- c_i é o parâmetro da probabilidade de acerto casual
- $D = 1,702$ é o parâmetro de escala.

Este modelo pode também ser interpretado como:

“A probabilidade do examinado i chutar e acertar mais a probabilidade do examinado i não chutar e acertar.”

Tem-se que a probabilidade de acerto casual não é simplesmente o quociente $\frac{1}{q}$ onde, q que é o número de alternativas do item, essa probabilidade depende de como o item é elaborado. Assim, itens com o mesmo número de alternativas, terão diferentes valores de c_i .

Nos modelos logísticos de 1 e 2 parâmetros quando $\theta_i = b_j$, a probabilidade de acerto do item é de 50%, já no modelo de 3 parâmetros, têm-se que:

$$\begin{aligned} P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) &= c_i + (1 - c_i) \frac{1}{1 + e^{D(-a(\theta_i - b_j))}}, \text{ se } \theta_i = b_j, \text{ então:} \\ c_i + (1 - c_i) \frac{1}{1 + e^{D(-a(\theta_i - b_j))}} &= c_i + (1 - c_i) \frac{1}{1 + e^{D(-a(b_j - b_j))}} \\ &= c_i + (1 - c_i) \frac{1}{1 + e^0} = c_i + (1 - c_i) \frac{1}{2} = c_i + \frac{1}{2} - \frac{c_i}{2} = \frac{c_i + 1}{2} = \frac{c_i}{2} + 0,5 \\ &= \frac{c_i}{2} + 50\% \end{aligned}$$

Neste modelo, se $\theta_i = b_j$, a probabilidade de acerto não é igual a 50%, como nos modelos de 1 e 2 parâmetros, a probabilidade nesse modelo é 50% acrescida da probabilidade de acerto casual. O ponto de inflexão continua sendo $\theta_i = b_j$, o parâmetro de discriminação continua proporcional à inclinação da reta tangente à curva de inflexão, mas a proporcionalidade nesse modelo é dada por:

$$\alpha_j = \frac{4}{1,702(1 - c_i)} f'(\theta_j). \text{ As demonstrações destas afirmações estão no Apêndice$$

D.

A CCI do modelo logístico de 3 parâmetros está representada abaixo.

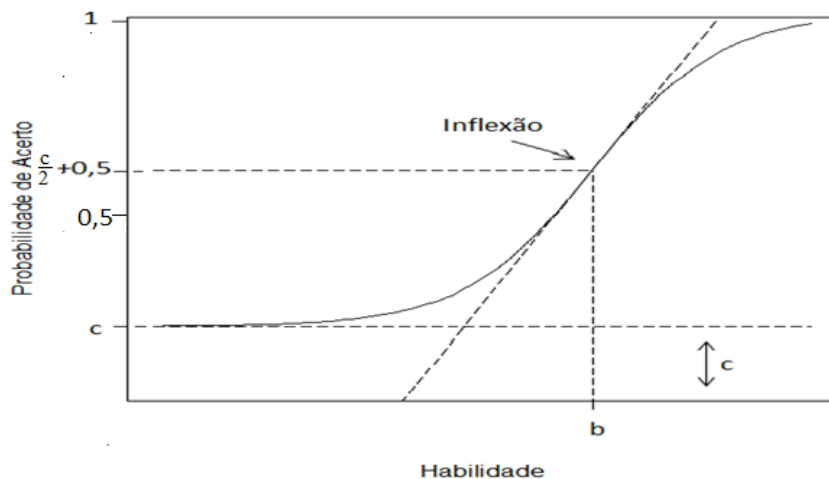


Figura 7 Curva Característica do Item do Modelo Logístico de 3 Parâmetros

2.7 Função informação do item

A função de informação do item é obtida através da Informação de Fisher que é dada por:

$$I_{\theta_i}(\theta_i) = E \left[\left(\frac{\partial \log f(Y_{ij} | \theta_i)}{\partial \theta_i} \right)^2 \right]$$

A informação de Fisher é uma quantidade, tal quantidade representa a informação contida em uma amostra, sobre uma variável aleatória. Por exemplo, calculando-se a informação de Fisher (ou do item) para dois itens respondidos por um certo grupo de examinados, pode-se obter como resultado 0,14 para o primeiro e 0,25 para o segundo. Isso significa que o segundo item contém mais informação a respeito da variável aleatória habilidade que o primeiro, o que pode ser muito útil para seleção de itens relevantes em um teste.

Pode-se reescrever a informação do item da seguinte forma:

$$I_{\mathcal{F}}(\theta_i) = \mathbb{E} \left[\left(\frac{\partial \log f(Y_{ij} | \theta_i)}{\partial \theta_i} \right)^2 \right] = \frac{\left(\frac{\partial \pi_{ij}}{\partial \theta_i} \right)^2}{\pi_{ij}(1-\pi_{ij})}$$

Dessa forma, a expressão para o modelo logístico de 1,2 e 3 parâmetros serão:

2.7.1 Modelo de 1 parâmetro

$$I(\theta_i)_1 = D^2 \pi_{ij} (1 - \pi_{ij})$$

2.7.2 Modelo de 2 parâmetros

$$I(\theta_i)_2 = D^2 a_j^2 \pi_{ij} (1 - \pi_{ij})$$

2.7.3 Modelo logístico de 3 parâmetros

$$I(\theta_i)_3 = D^2 a_j^2 \frac{(1-\pi_{ij})}{\pi_{ij}} \left(\frac{\pi_{ij} - c_j}{1-c_j} \right)^2$$

Detalhes dessas demonstrações podem ser obtidos em Bragion (2010, p.36)

Abaixo estão os gráficos da CCI e da Informação do Item para os modelos logísticos de 1,2 e 3 parâmetros.

A) Modelo Logístico de 1 Parâmetro:

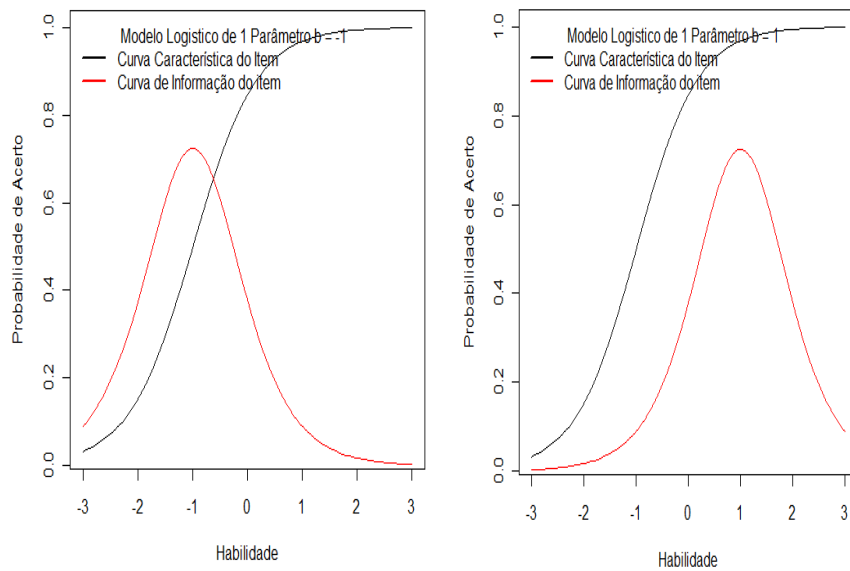


Figura 8 Curvas características dos itens e curvas de informação dos itens de modelos logísticos de 1 parâmetro

B) Modelo Logístico de 2 Parâmetros

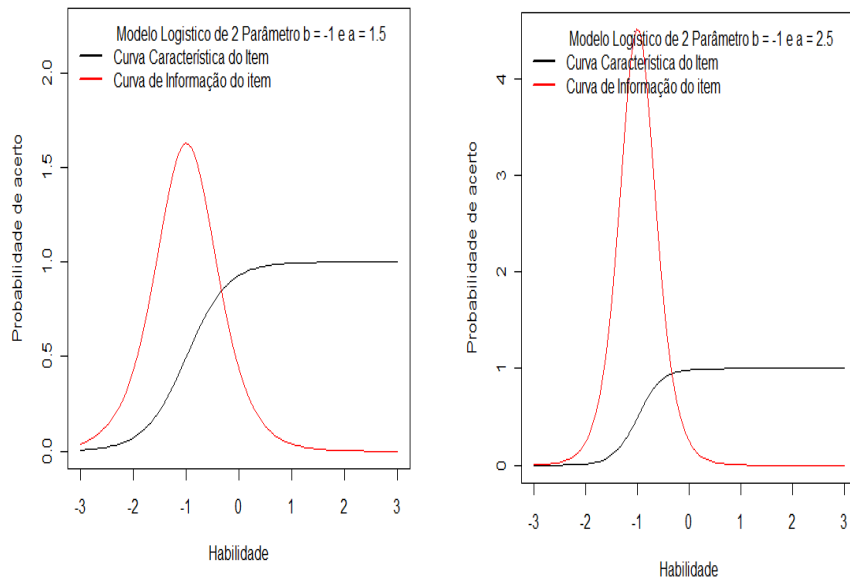


Figura 9 Curvas características dos itens e curvas de informação dos itens de modelos logísticos de 2 parâmetros

C) Modelo Logístico de 3 Parâmetros:

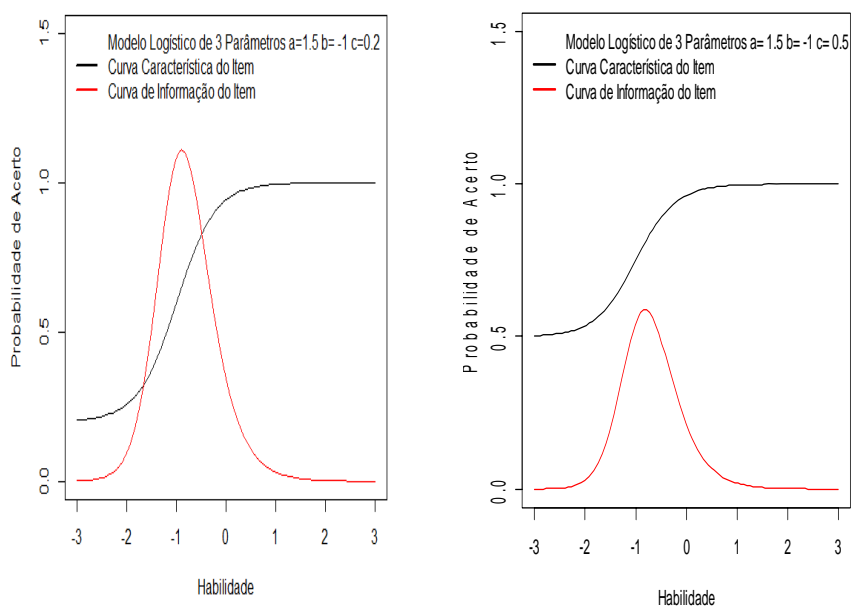


Figura 10 Característica do item e curva de informação do item de modelos logísticos de 3 parâmetros

Observa-se que a informação do item é maior quando a habilidade se aproxima do nível de dificuldade do item (θ_i se aproxima de b_i), ou seja, um item traz mais informação a respeito da variável habilidade se é aplicado para examinados com habilidade próxima à dificuldade do item. Já se a habilidade é muito distante (tanto para mais quanto para menos) da dificuldade do item, a informação do item tende a zero.

Através da figura 9, pode-se verificar que quanto maior o valor de a_i , maior é a informação do item e na Figura 10, observa-se que, quanto maior o valor de c menor é a informação do item. Por isso, o modelo logístico de 3 parâmetros é o que fornece menos informação sobre a habilidade.

2.8 Estimação de parâmetros

Este é um ponto fundamental da TRI; a estimação dos parâmetros dos itens e da habilidade dos examinados. Inicialmente têm-se apenas as respostas dos examinados e suas respectivas notas, através de métodos estatísticos de estimação, obtêm-se os parâmetros dos itens e as habilidades dos examinados. Existem vários métodos de estimação como: Máxima Verossimilhança Marginal, Máxima Verossimilhança Conjunta, Bayesiano, Algoritmo EM, dentre outras.

2.8.1 Estimação por máxima verossimilhança conjunta

Nesse método, os parâmetros são estimados em dois passos:

1º Passo:

Os parâmetros dos itens são estimados supondo conhecidas as habilidades, nesse passo as habilidades supostas inicialmente, são as notas dos examinados padronizadas.

2º Passo:

As habilidades são estimadas supondo conhecidos os parâmetros dos itens, os valores supostos para os parâmetros dos itens são os encontrados no passo 1.

Esse processo de dois estágios é repetido até a convergência das habilidades e dos parâmetros dos itens.

2.8.1.1 Estimação dos parâmetros dos itens quando as habilidades são conhecidas (ou supostamente conhecidas)

As suposições dessa estimação são:

- i. Independência entre as respostas de examinados diferentes.

ii. Independência entre as resposta de um mesmo examinado a diferentes itens (independência local).

Seja $\mathbf{Y}_{ij} = (y_{11}, y_{12}, \dots, y_{1j}, y_{21}, \dots, y_{2j}, \dots, y_{n1}, \dots, y_{nj})'$ o vetor contendo todas as respostas dos n examinados aos J itens e $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n)'$ o vetor com as habilidades dos n examinados. Devido às independências citadas anteriormente, temos:

(por i)

$$\begin{aligned} & f(\mathbf{Y}_{i1} = y_{11}, y_{21}, \dots, y_{n1} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_1) \\ &= f(y_{11} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_1) \cdot f(y_{21} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_1) \dots f(y_{n1} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_1) \\ &= \prod_{i=1}^n f(y_{i1} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_1) \end{aligned}$$

(por i)

$$\begin{aligned} & f(\mathbf{Y}_{i2} = y_{12}, y_{22}, \dots, y_{n2} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_2) \\ &= f(y_{12} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_2) \cdot f(y_{22} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_2) \dots f(y_{n2} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_2) \\ &= \prod_{i=1}^n f(y_{i2} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_2) \end{aligned}$$

⋮

(por i)

$$\begin{aligned} & f(\mathbf{Y}_{iJ} = y_{1J}, y_{2J}, \dots, y_{nJ} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_J) \\ &= f(y_{1J} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_J) \cdot f(y_{2J} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_J) \dots f(y_{nJ} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_J) \\ &= \prod_{i=1}^n f(y_{iJ} | \boldsymbol{\theta} = \theta_i; \boldsymbol{\delta}_J) \end{aligned}$$

Assim:

(por i e por ii)

$$\begin{aligned} & f(\mathbf{Y}_{ij} = y_{11}, y_{12}, \dots, y_{1j}, y_{21}, \dots, y_{2j}, \dots, y_{n1}, \dots, y_{nj} | \boldsymbol{\theta}; \boldsymbol{\Delta}) \\ &= f(y_{11} | \boldsymbol{\theta} = \theta_1; \boldsymbol{\delta}_1) \dots f(y_{nJ} | \boldsymbol{\theta} = \theta_n; \boldsymbol{\delta}_J) \end{aligned}$$

$$\begin{aligned}
&= \prod_{i=1}^n f(y_{i1} | \Theta = \theta_j; \delta_j) \dots \prod_{i=1}^n f(y_{ij} | \Theta = \theta_j; \delta_j) \\
&= \prod_{j=1}^J \prod_{i=1}^n f(y_{ij} = y_{ij} | \Theta = \theta_j; \delta_j)
\end{aligned}$$

Dessa forma, a função de verossimilhança de $\mathbf{\Delta}$, tendo em vista que as habilidades são conhecidas é dada por:

$$L(\mathbf{\Delta}; \mathbf{Y}_{ij}) = \prod_{i=1}^n \prod_{j=1}^J f(y_{ij} = y_{ij} | \Theta = \theta_j; \delta_j)$$

Lembrando que:

$$f(\mathbf{Y}_{ij} = y_{ij} | \Theta = \theta_j; \delta_j) = P(\mathbf{Y}_{ij} = y_{ij} | \Theta = \theta_j; \delta_j)$$

Como Y_{ij} tem distribuição de Bernoulli (assume os valores 0 ou 1):

$$P(\mathbf{Y}_{ij} = y_{ij} | \Theta = \theta_j; \delta_j) =$$

$$P(\mathbf{Y}_{ij} = 1 | \Theta = \theta_j; \delta_j)^{y_{ij}} P(\mathbf{Y}_{ij} = 0 | \Theta = \theta_j; \delta_j)^{1-y_{ij}} = \pi_{ij}^{y_{ij}} (1 - \pi_{ij})^{1-y_{ij}}$$

Assim, a função de verossimilhança pode ser escrita como:

$$\begin{aligned}
L(\mathbf{\Delta}; \mathbf{Y}_{ij}) &= \prod_{i=1}^n \prod_{j=1}^J f(y_{ij} = y_{ij} | \Theta = \theta_j; \delta_j) \\
&= \prod_{i=1}^n \prod_{j=1}^J \pi_{ij}^{y_{ij}} (1 - \pi_{ij})^{1-y_{ij}}
\end{aligned}$$

Aplicando o logaritmo a ambos os membros, teremos a log-verossimilhança.

$$l(\mathbf{\Delta}; \mathbf{Y}_{ij}) = \sum_{i=1}^n \sum_{j=1}^J \{y_{ij} \log \pi_{ij} + (1 - y_{ij}) \log(1 - \pi_{ij})\}$$

Os estimadores de máxima verossimilhança de δ_j , $j= 1, \dots, J$ são encontrados como raízes da equação de verossimilhança.

$$l'(\mathbf{\Delta}; \mathbf{Y}_{ij}) = \frac{\partial l(\mathbf{\Delta}; \mathbf{Y}_{ij})}{\partial \delta_j} = 0 \quad j= 1, \dots, J$$

Sendo:

$$\frac{\partial l(\mathbf{\Delta}; \mathbf{Y}_{ij})}{\partial \delta_j} = \sum_{i=1}^n \left\{ y_{ij} \frac{\partial (\log \pi_{ij})}{\partial \delta_j} + (1 - y_{ij}) \frac{\partial (\log(1 - \pi_{ij}))}{\partial \delta_j} \right\}$$

$$\begin{aligned}
&= \sum_{i=1}^n \left\{ y_{ij} \frac{1}{\pi_{ij}} \frac{\partial(\pi_{ij})}{\partial \delta_j} - (1 - y_{ij}) \frac{1}{(1 - \pi_{ij})} \frac{\partial(\pi_{ij})}{\partial \delta_j} \right\} \\
&= \sum_{i=1}^n \left\{ \frac{\partial(\pi_{ij})}{\partial \delta_j} \left[y_{ij} \frac{1}{\pi_{ij}} - (1 - y_{ij}) \frac{1}{(1 - \pi_{ij})} \right] \right\} = \sum_{i=1}^n \left\{ \frac{\partial(\pi_{ij})}{\partial \delta_j} \left[\frac{y_{ij} - \pi_{ij}}{\pi_{ij}(1 - \pi_{ij})} \right] \right\} \quad (2)
\end{aligned}$$

O Modelo Logístico de 3 Parâmetros será usado para calcular π_{ij} , tendo em vista que os demais modelos são casos particulares deste modelo, assim:

$$\pi_{ij} = P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j) = c_i + (1 - c_i) \frac{1}{1 + e^{-D(\theta_i - b_j)}}$$

Lembrando que:

$$\frac{\pi_{ij} - c_j}{1 - c_j} = \left(\frac{1}{1 + e^{-D\alpha_j(\theta_i - b_j)}} \right) \text{ e } (1 - \pi_{ij}) = \frac{(1 - c_j)e^{-D\alpha_j(\theta_i - b_j)}}{1 + e^{-D\alpha_j(\theta_i - b_j)}}$$

Tem-se:

$$\frac{\partial(\pi_{ij})}{\partial \alpha_j} = \frac{D(1 - c_j)(\theta_i - b_j)e^{-D\alpha_j(\theta_i - b_j)}}{(1 + e^{-D\alpha_j(\theta_i - b_j)})^2} = \frac{D(\theta_i - b_j)(1 - \pi_{ij})(\pi_{ij} - c_j)}{(1 - c_j)} \quad (3)$$

$$\frac{\partial(\pi_{ij})}{\partial b_j} = \frac{-D\alpha_j(1 - c_j)e^{-D\alpha_j(\theta_i - b_j)}}{(1 + e^{-D\alpha_j(\theta_i - b_j)})^2} = -\frac{D\alpha_j(1 - \pi_{ij})(\pi_{ij} - c_j)}{(1 - c_j)} \quad (4)$$

$$\frac{\partial(\pi_{ij})}{\partial c_j} = 1 - \frac{1}{1 + e^{-D\alpha_j(\theta_i - b_j)}} = 1 - \frac{\pi_{ij} - c_j}{1 - c_j} = \frac{1 - c_j - \pi_{ij} + c_j}{1 - c_j} = \frac{1 - \pi_{ij}}{1 - c_j} \quad (5)$$

Utilizando as expressões (3), (4) e (5) em (2) obtêm-se:

$$\begin{aligned}
\frac{\partial l(\Delta Y_{ij})}{\partial \alpha_j} &= \sum_{i=1}^n \left\{ \frac{\partial(\pi_{ij})}{\partial \alpha_j} \left[\frac{y_{ij} - \pi_{ij}}{\pi_{ij}(1 - \pi_{ij})} \right] \right\} = \sum_{i=1}^n \left\{ \frac{D(\theta_i - b_j)(1 - \pi_{ij})(\pi_{ij} - c_j)}{(1 - c_j)} \left[\frac{y_{ij} - \pi_{ij}}{\pi_{ij}(1 - \pi_{ij})} \right] \right\} \\
&= \frac{D}{(1 - c_j)} \sum_{i=1}^n \frac{(\theta_i - b_j)(y_{ij} - \pi_{ij})(\pi_{ij} - c_j)}{\pi_{ij}} \quad (6)
\end{aligned}$$

$$\frac{\partial l(\Delta Y_{ij})}{\partial b_j} = \sum_{i=1}^n \left\{ \frac{\partial(\pi_{ij})}{\partial b_j} \left[\frac{y_{ij} - \pi_{ij}}{\pi_{ij}(1 - \pi_{ij})} \right] \right\}$$

$$\begin{aligned}
&= \sum_{i=1}^n \left\{ -\frac{D\alpha_j(1-\pi_{ij})(\pi_{ij}-c_j)}{(1-c_j)} \left[\frac{y_{ij}-\pi_{ij}}{\pi_{ij}(1-\pi_{ij})} \right] \right\} \\
&= -\frac{D\alpha_j}{(1-c_j)} \sum_{i=1}^n \left\{ \frac{(\pi_{ij}-c_j)(y_{ij}-\pi_{ij})}{\pi_{ij}} \right\} \quad (7)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial l(\Delta_i Y_{ij})}{\partial c_j} &= \sum_{i=1}^n \left\{ \frac{\partial(\pi_{ij})}{\partial c_j} \left[\frac{y_{ij}-\pi_{ij}}{\pi_{ij}(1-\pi_{ij})} \right] \right\} \\
&= \sum_{i=1}^n \left\{ \frac{1-\pi_{ij}}{1-c_j} \left[\frac{y_{ij}-\pi_{ij}}{\pi_{ij}(1-\pi_{ij})} \right] \right\} = \frac{1}{1-c_j} \sum_{i=1}^n \left\{ \left[\frac{y_{ij}-\pi_{ij}}{\pi_{ij}} \right] \right\} \quad (8)
\end{aligned}$$

Assim, as raízes das equações de estimação dos parâmetros α , b e c são encontradas igualando $\frac{\partial l(\Delta_i Y_{ij})}{\partial c_j}$:

$$\alpha_j: \frac{D}{(1-c_j)} \sum_{i=1}^n \frac{(\theta_i - b_j)(y_{ij} - \pi_{ij})(\pi_{ij} - c_j)}{\pi_{ij}} = 0$$

$$b_j: -\frac{D\alpha_j}{(1-c_j)} \sum_{i=1}^n \left\{ \frac{(\pi_{ij}-c_j)(y_{ij}-\pi_{ij})}{\pi_{ij}} \right\} = 0$$

$$c_j: \frac{1}{1-c_j} \sum_{i=1}^n \left\{ \left[\frac{y_{ij}-\pi_{ij}}{\pi_{ij}} \right] \right\} = 0$$

Essas equações não possuem soluções explícitas, desta forma, para obter as estimativas de máxima verossimilhança, será necessário o uso de algum método iterativo.

2.8.1.1.2 Algoritmo Newton-Raphson

O método Newton-Raphson é um processo iterativo utilizado em muitos programas computacionais e tem como objetivo estimar raízes de uma equação.

Como mencionado anteriormente:

$\delta_j = (\alpha_j, b_j, c_j)'$ é vetor de parâmetros do item j . Seja $\tilde{\delta}_j^{(0)} = (\alpha_j^{(0)}, b_j^{(0)}, c_j^{(0)})'$ o vetor com valores iniciais dos parâmetros, assim uma estimativa atualizada de δ_j será: $\tilde{\delta}_j^{(1)} = \tilde{\delta}_j^{(0)} + \varepsilon \tilde{\delta}_j^{(0)}$, assim:

$$\tilde{\alpha}_j^{(1)} = \tilde{\alpha}_j^{(0)} + \varepsilon \tilde{\alpha}_j^{(0)}$$

$$\tilde{b}_j^{(1)} = \tilde{b}_j^{(0)} + \varepsilon \tilde{b}_j^{(0)}$$

$$\tilde{c}_j^{(1)} = \tilde{c}_j^{(0)} + \varepsilon \tilde{c}_j^{(0)}$$

Em que $\varepsilon \tilde{\alpha}_j^{(0)}$, $\varepsilon \tilde{b}_j^{(0)}$, $\varepsilon \tilde{c}_j^{(0)}$ são os erros de aproximações. A expansão em série de Taylor da log-verossimilhança em torno de $\tilde{\delta}_j^{(0)}$ é dada por:

$$\frac{\partial l(\Delta_i Y_{ij})}{\partial \alpha_j} = \frac{\partial l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial \alpha_j} + \varepsilon \tilde{\alpha}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial \alpha_j^2} + \varepsilon \tilde{b}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial \alpha_j \partial b_j} + \varepsilon \tilde{c}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial \alpha_j \partial c_j} + R_{\alpha_j}(\tilde{\delta}_j^{(0)})$$

$$\frac{\partial l(\Delta_i Y_{ij})}{\partial b_j} = \frac{\partial l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial b_j} + \varepsilon \tilde{\alpha}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial b_j^2} + \varepsilon \tilde{b}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial b_j \partial \alpha_j} + \varepsilon \tilde{c}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)} | Y_{ij})}{\partial b_j \partial c_j} + R_{b_j}(\tilde{\delta}_j^{(0)})$$

$$\begin{aligned} \frac{\partial l(\Delta; Y_{ij})}{\partial \sigma_j} &= \frac{\partial l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma_j} + \tilde{\alpha}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma_j^2} + \tilde{\delta}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma_j \partial \alpha_j} + \\ &\tilde{\varepsilon}_j^{(0)} \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma_j \partial b_j} + R_{\sigma_j}(\tilde{\delta}_j^{(0)}) \end{aligned}$$

As raízes das equações de verossimilhança são encontradas quando $\frac{\partial l(\Delta; Y_{ij})}{\partial \delta_j} = 0$, ou seja:

$$\frac{\partial l(\Delta; Y_{ij})}{\partial \alpha_j} = \frac{\partial l(\Delta; Y_{ij})}{\partial b_j} = \frac{\partial l(\Delta; Y_{ij})}{\partial \sigma_j} = 0$$

Para simplificação de notação, denotaremos:

$$\begin{aligned} L_1 &= \frac{\partial l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \alpha_j} & L_{11} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \alpha_j^2} & L_{12} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \alpha_j \partial b_j} \\ L_{13} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \alpha_j \partial \sigma_j} & & & & \\ L_2 &= \frac{\partial l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial b_j} & L_{21} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial b_j^2} & L_{22} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial b_j \partial \alpha_j} \\ L_{23} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial b_j \partial \sigma_j} & & & & \\ L_3 &= \frac{\partial l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma} & L_{31} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma^2} & L_{32} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma_j \partial \alpha_j} \\ L_{33} &= \frac{\partial^2 l(\tilde{\delta}_j^{(0)}; Y_{ij})}{\partial \sigma_j \partial b_j} & & & & \end{aligned}$$

Desprezando os restos $R_{\alpha_j}(\tilde{\delta}_j^{(0)})$, $R_{b_j}(\tilde{\delta}_j^{(0)})$ e $R_{\sigma_j}(\tilde{\delta}_j^{(0)})$, têm-se:

$$0 = L_1 + L_{11} s\hat{\alpha}_j^{(0)} + L_{12} s\hat{b}_j^{(0)} + L_{13} s\hat{c}_j^{(0)}$$

$$-L_1 = L_{11} s\hat{\alpha}_j^{(0)} + L_{12} s\hat{b}_j^{(0)} + L_{13} s\hat{c}_j^{(0)}$$

$$0 = L_2 + L_{21} s\hat{\alpha}_j^{(0)} + L_{22} s\hat{b}_j^{(0)} + L_{23} s\hat{c}_j^{(0)}$$

$$-L_2 = L_{21} s\hat{\alpha}_j^{(0)} + L_{22} s\hat{b}_j^{(0)} + L_{23} s\hat{c}_j^{(0)}$$

$$0 = L_3 + L_{31} s\hat{\alpha}_j^{(0)} + L_{32} s\hat{b}_j^{(0)} + L_{33} s\hat{c}_j^{(0)}$$

$$-L_3 = L_{31} s\hat{\alpha}_j^{(0)} + L_{32} s\hat{b}_j^{(0)} + L_{33} s\hat{c}_j^{(0)}$$

Na forma matricial:

$$-\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} s\hat{\alpha}_j^{(0)} \\ s\hat{b}_j^{(0)} \\ s\hat{c}_j^{(0)} \end{bmatrix}$$

$$\begin{bmatrix} s\hat{\alpha}_j^{(0)} \\ s\hat{b}_j^{(0)} \\ s\hat{c}_j^{(0)} \end{bmatrix} = -\begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix}^{-1} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix}$$

Como: $\begin{bmatrix} s\hat{\alpha}_j^{(0)} \\ s\hat{b}_j^{(0)} \\ s\hat{c}_j^{(0)} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_j^{(1)} \\ \hat{b}_j^{(1)} \\ \hat{c}_j^{(1)} \end{bmatrix} - \begin{bmatrix} \hat{\alpha}_j^{(0)} \\ \hat{b}_j^{(0)} \\ \hat{c}_j^{(0)} \end{bmatrix}$ têm-se que:

$$\begin{bmatrix} \hat{\alpha}_j^{(1)} \\ \hat{\beta}_j^{(1)} \\ \hat{c}_j^{(1)} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_j^{(0)} \\ \hat{\beta}_j^{(0)} \\ \hat{c}_j^{(0)} \end{bmatrix} - \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix}^{-1} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \quad (8)$$

Esse procedimento fornece $\delta_j^{(1)}$, para obter $\delta_j^{(2)}$ o mesmo procedimento será utilizado, porém, tendo $\delta_j^{(1)}$ como valor inicial. Da mesma forma para obter $\delta_j^{(3)}$, utilizamos o mesmo procedimento, tendo $\delta_j^{(2)}$ como valor inicial e assim por diante. Esse processo é repetido até que algum critério para encerrá-lo, seja atingido, um critério, por exemplo, é de que o erro em uma iteração seja menor que 0,0001. O número de iterações até a obtenção desse critério será p, ou seja:

$$\varepsilon \delta_j^{(p)} = \delta_j^{(p)} - \delta_j^{(p-1)} < 0,0001$$

As estimativas iniciais para $\hat{\delta}_j^{(0)} = (\alpha_j^{(0)}, \beta_j^{(0)}, c_j^{(0)})'$, podem ser obtidas por:

$$a_j^{(0)} : \sigma_{TY_j} = \frac{\alpha_j}{\sqrt{1+\alpha_j^2}}$$

Sendo σ_{TY_j} a correlação bisserial entre a nota obtida pelo(s) examinados(s) e suas respectivas respostas no item j.

$$b_j^{(0)} : P_j = \phi(-v_j), v_j = b_j \sigma_{TY_j}$$

Sendo P_j é a proporção de respostas corretas no item j, por exemplo:
 Suponha que um item respondido por 50 examinados tenha 10 respostas corretas
 assim, $P_j = \frac{10}{50}$

$$c_j(\theta) = \frac{1}{q_j}$$

Sendo q_j o número de alternativas do item j.

Na matriz 8, os valores de L_{11} , L_{21} e L_{31} são obtidos através das expressões (6), (7) e (8) respectivamente. A obtenção de L_{11} , L_{12} , ..., L_{33} é dada por: $\frac{\partial^2 l(\Delta F_{ij})}{\partial \delta_j \partial \delta_j}$.

2.8.1.1.3 Estimação das habilidades dos examinados quando os parâmetros dos itens são conhecidos

As suposições dessa estimação são:

- i. Independência entre as respostas de examinados diferentes.
- ii. Independência entre as respostas de um mesmo examinado a diferentes itens (independência local).

Seja $\mathbf{Y}_{ij} = (y_{11}, y_{12}, \dots, y_{1j}, y_{21}, \dots, y_{2j}, \dots, y_{nj})$ o vetor contendo todas as respostas dos n examinados aos J itens e $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ o vetor de habilidades dos n examinados. Devido às independências citadas anteriormente, temos:

(por ii)

$$f(\mathbf{Y}_{1j} = y_{11}, y_{12}, \dots, y_{1j} | \theta = \theta_1; \delta_j)$$

$$= f(y_{11} | \theta = \theta_1; \delta_1) \cdot f(y_{12} | \theta = \theta_1; \delta_2) \dots f(y_{1j} | \theta = \theta_1; \delta_j)$$

$$= \prod_{j=1}^J f(y_{1j} | \theta = \theta_1; \delta_j)$$

(por ii)

$$f(\mathbf{Y}_{2j} = y_{21}, y_{22}, \dots, y_{2j} | \theta = \theta_2; \delta_j)$$

$$= f(y_{21} | \theta = \theta_2; \delta_1) \cdot f(y_{22} | \theta = \theta_2; \delta_2) \dots f(y_{2j} | \theta = \theta_2; \delta_j)$$

$$= \prod_{j=1}^J f(y_{2j} | \theta = \theta_2; \delta_j)$$

:

(por ii)

$$f(\mathbf{Y}_{nj} = y_{n1}, y_{n2}, \dots, y_{nj} | \theta = \theta_n; \delta_j)$$

$$= f(y_{n1} | \theta = \theta_n; \delta_1) \cdot f(y_{n2} | \theta = \theta_n; \delta_2) \dots f(y_{nj} | \theta = \theta_n; \delta_j)$$

$$= \prod_{j=1}^J f(y_{nj} | \theta = \theta_n; \delta_j)$$

Assim:

(por i e por ii)

$$f(\mathbf{Y}_{ij} = y_{i1}, y_{i2}, \dots, y_{ij}, y_{21}, \dots, y_{2j}, \dots, y_{nj} | \theta; \Delta)$$

$$= f(y_{11} | \theta = \theta_1; \delta_1) \dots f(y_{nj} | \theta = \theta_n; \delta_j) = \prod_{j=1}^J f(y_{1j} | \theta = \theta_1; \delta_j)$$

$$\dots \prod_{j=1}^J f(y_{nj} | \theta = \theta_n; \delta_j) = \prod_{i=1}^n \prod_{j=1}^J f(y_{ij} | \theta = \theta_i; \delta_j)$$

Dessa forma, a função de verossimilhança de θ , tendo em vista que os parâmetros dos itens são conhecidos é dada por:

$$L(\theta; \mathbf{Y}_{ij}) = \prod_{i=1}^n \prod_{j=1}^J f(y_{ij} | \theta = \theta_i; \delta_j)$$

Lembrando que:

$$f(\mathbf{Y}_{ij} = y_{ij} | \theta = \theta_i; \delta_j) = P(\mathbf{Y}_{ij} = y_{ij} | \theta = \theta_i; \delta_j)$$

Como Y_{ij} tem distribuição de Bernoulli (assume os valores 0 ou 1):

$$\begin{aligned}
& P(Y_{ij} = y_{ij} | \Theta = \theta_i; \delta_j) \\
&= P(Y_{ij} = 1 | \Theta = \theta_i; \delta_j)^{y_{ij}} P(Y_{ij} = 0 | \Theta = \theta_i; \delta_j)^{1-y_{ij}} \\
&= \pi_{ij}^{y_{ij}} (1 - \pi_{ij})^{1-y_{ij}}
\end{aligned}$$

Assim a função de verossimilhança pode ser escrita como:

$$\begin{aligned}
L(\Theta; \mathbf{Y}_{ij}) &= \prod_{i=1}^n \prod_{j=1}^J f(Y_{ij} = y_{ij} | \Theta = \theta_i; \delta_j) \\
&= \prod_{i=1}^n \prod_{j=1}^J \pi_{ij}^{y_{ij}} (1 - \pi_{ij})^{1-y_{ij}}
\end{aligned}$$

Aplicando o logaritmo a ambos os membros, teremos a log-verossimilhança.

$$l(\Theta; \mathbf{Y}_{ij}) = \sum_{i=1}^n \sum_{j=1}^J \{y_{ij} \log \pi_{ij} + (1 - y_{ij}) \log(1 - \pi_{ij})\}$$

Os estimadores de máxima verossimilhança de δ_j , $j=1, \dots, J$ são encontrados como raízes da equação de verossimilhança.

$$l'(\Theta; \mathbf{Y}_{ij}) = \frac{\partial l(\Theta; \mathbf{Y}_{ij})}{\partial \theta_i} = 0 \quad j=1, \dots, J$$

Sendo:

$$\begin{aligned}
\frac{\partial l(\Theta; \mathbf{Y}_{ij})}{\partial \theta_i} &= \sum_{j=1}^J \left\{ y_{ij} \frac{\partial(\log \pi_{ij})}{\partial \theta_i} + (1 - y_{ij}) \frac{\partial(\log(1 - \pi_{ij}))}{\partial \theta_i} \right\} \\
&= \sum_{j=1}^J \left\{ y_{ij} \frac{1}{\pi_{ij}} \frac{\partial(\pi_{ij})}{\partial \theta_i} - (1 - y_{ij}) \frac{1}{(1 - \pi_{ij})} \frac{\partial(\pi_{ij})}{\partial \theta_i} \right\} \\
&= \sum_{j=1}^J \left\{ \frac{\partial(\pi_{ij})}{\partial \theta_i} \left[y_{ij} \frac{1}{\pi_{ij}} - (1 - y_{ij}) \frac{1}{(1 - \pi_{ij})} \right] \right\} = \sum_{j=1}^J \left\{ \frac{\partial(\pi_{ij})}{\partial \theta_i} \left[\frac{y_{ij} - \pi_{ij}}{\pi_{ij}(1 - \pi_{ij})} \right] \right\} \quad (9)
\end{aligned}$$

O Modelo Logístico de 3 Parâmetros será usado para calcular π_{ij} , tendo em vista que os demais modelos são casos particulares deste modelo.

Lembrando que:

$$\frac{\pi_{ij} - c_j}{1 - c_j} = \left(\frac{1}{1 + e^{-D\alpha_j(\theta_i - \beta_j)}} \right) \text{ e } (1 - \pi_{ij}) = \frac{(1 - c_j)e^{-D\alpha_j(\theta_i - \beta_j)}}{1 + e^{-D\alpha_j(\theta_i - \beta_j)}}$$

$$\text{Temos: } \frac{\partial(\pi_{ij})}{\partial \theta_i} = \frac{D(1 - c_j)\alpha_j e^{-D\alpha_j(\theta_i - \beta_j)}}{(1 + e^{-D\alpha_j(\theta_i - \beta_j)})^2} = D\alpha_j \left(\frac{\pi_{ij} - c_j}{1 - c_j} \right) (1 - \pi_{ij})$$

Dessa forma:

$$\begin{aligned} \frac{\partial l(\theta; Y_{ij})}{\partial \theta_i} &= \sum_{j=1}^J \left\{ D a_j \left(\frac{\pi_{ij} - c_j}{1 - c_j} \right) (1 - \pi_{ij}) \left[\frac{y_{ij} - \pi_{ij}}{\pi_{ij}(1 - \pi_{ij})} \right] \right\} \\ &= \sum_{j=1}^J \left\{ D a_j \left(\frac{\pi_{ij} - c_j}{1 - c_j} \right) \left(\frac{y_{ij} - \pi_{ij}}{\pi_{ij}} \right) \right\} \end{aligned}$$

Assim, a raiz da equação de estimação para θ_i é encontrada igualando $\frac{\partial l(\theta; Y_{ij})}{\partial \theta_i}$ a zero.

$$\theta_i: D \sum_{j=1}^J \left\{ a_j \left(\frac{\pi_{ij} - c_j}{1 - c_j} \right) \left(\frac{y_{ij} - \pi_{ij}}{\pi_{ij}} \right) \right\} = 0$$

Essa equação não possui solução explícita, desta forma, para obter as estimativas de máxima verossimilhança será necessário o uso de algum método iterativo.

2.8.1.1.4 Algoritmo de Newton-Raphson

Como mencionado anteriormente, o método Newton-Raphson é um processo iterativo utilizado em muitos programas computacionais e tem como objetivo estimar raízes de uma equação.

$\theta = (\theta_1, \theta_2, \dots, \theta_n)$ é vetor de habilidades dos n examinados. Seja $\bar{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_n^{(0)})'$ o vetor com valores iniciais das habilidades, assim uma estimativa atualizada de θ será: $\bar{\theta}^{(1)} = \bar{\theta}^{(0)} + \delta \bar{\theta}^{(0)}$, assim:

$$\begin{aligned}\bar{\theta}_1^{(1)} &= \bar{\theta}_1^{(0)} + \varepsilon \bar{\theta}_1^{(0)} \\ \bar{\theta}_2^{(1)} &= \bar{\theta}_2^{(0)} + \varepsilon \bar{\theta}_2^{(0)} \\ &\vdots \\ \bar{\theta}_n^{(1)} &= \bar{\theta}_n^{(0)} + \varepsilon \bar{\theta}_n^{(0)}\end{aligned}$$

Em que $\varepsilon \bar{\theta}_1^{(0)}$, $\varepsilon \bar{\theta}_2^{(0)}$, ..., $\varepsilon \bar{\theta}_n^{(0)}$ são erros de aproximações. Os procedimentos para a estimação das habilidades, supondo os parâmetros dos itens conhecidos são os mesmos citados pelo algoritmo de Newton-Raphson (em 2.8.1.1.2).

2.8.2 Algoritmo EM utilizado no ICL

O nome EM vem de Esperança e Maximização, o algoritmo EM utilizado no ICL faz a discretização da variável latente θ em K conhecidos discretos valores q_k , $k=1, \dots, K$ com desconhecidas probabilidades π_k , $k=1, \dots, K$. Ou seja, π_k é a proporção real de pessoas em cada classe de habilidade. Os valores q_k são escolhidos na inicialização do processo e determinam a escala da variável latente. Todos os detalhes do algoritmo EM utilizado no ICL estão em WOODRUFF, D. J., & HANSON, B. A. (1997).

2.8.3 Sobre o ICL

O ICL (IRT Command Language) é um software criado por Brad Hanson, que pode estimar os parâmetros dos modelos logísticos de 1, 2 e 3 parâmetros para itens dicotômicos, e também os parâmetros dos modelos de crédito parcial e crédito parcial generalizado para itens politômicos, em todos os

modelos pode-se considerar um ou mais grupos de examinados. O ICL pode estimar os parâmetros de itens e habilidades dos examinados pelos métodos de máxima verossimilhança e Bayesianos, sendo utilizado o algoritmo EM para computar os parâmetros do modelo utilizado. O algoritmo EM utilizado pelo ICL pode ser encontrado em WOODRUFF E HANSON (1997); HANSON (1998).

Existem versões do software para Linux, Windows e Macintosh. O ICL é um software livre que faz parte do projeto Ciências Sociais de Medição e está hospedado no site da SourceFourge, sendo que seu desenvolvimento é gerido pelo site e permite a participação de qualquer pessoa interessada no seu desenvolvimento. O ICL é escrito na linguagem C++ e utiliza o ETIRM (Estimation Toolkit for Item Response Models- que é uma ferramenta para a estimativa de modelos da Teoria de Resposta ao Item). Informações sobre o ICL, a última versão, o código fonte podem ser obtidas pelo site (<http://www.b-a-h.com/software/irt/icl/>), disponível em 14/02/2012.

Mead, Morris & Blitz (2007) fizeram uma comparação entre o BILOG-MG, um software muito utilizado e o ICL. Tal trabalho comparou diversas características dos softwares como: manual, métodos de estimação, opções de estimação, possibilidade de escolha das distribuições *a priori*, informações produzidas pelo software como o ajuste do modelo, dentre outras. Sobre alguns aspectos concluiu-se ser o BILOG mais vantajoso que o ICL e em outros o ICL apresentou ser mais vantajoso, porém na precisão das estimações dos parâmetros, ambos os softwares tiveram resultados muito parecidos.

Jurich & Goodman (2009) também fizeram uma comparação entre o ICL e o software PARSCALE e concluíram que os dois softwares estimam os parâmetros da TRI de forma muito próxima.

O ICL é muito flexível com relação aos critérios utilizados para realizar as estimações, como a escolha de em quantos intervalos será dividida a variável

latente para a discretização, as distribuições das priores, os critérios de convergência, a quantidade de grupos respondentes e o modelo utilizado.

3 METODOLOGIA

Foram realizadas simulações, com o objetivo de estudar o comportamento das estimações do ICL, para o modelo logístico de 3 parâmetros. O processo simulativo contou com um programa escrito em Python e decorreu da seguinte forma:

- 1- Foram atribuídas distribuições para cada um dos parâmetros a , b , c e θ .
- 2- A partir das distribuições dos parâmetros, o programa em Python fez o sorteio dos valores para cada um dos parâmetros a , b , c e θ .
- 3- A partir dos valores sorteados, foi calculada a probabilidade de acerto para cada examinado, a cada item, através do modelo logístico de 3 parâmetros.
- 4- O programa sorteou um valor da distribuição uniforme (0,1).
- 5- O valor sorteado da uniforme (0,1), foi comparado com a probabilidade de acerto calculada. Se a probabilidade de acerto fosse maior que o valor sorteado, a resposta era considerada certa, caso contrário era como errada. Através deste processo construiu-se o conjunto de respostas dos examinados aos itens, ou seja, obteve-se o vetor \mathbf{Y}_{ij} .

O exemplo a seguir ilustra o processo de simulação de um teste com 20 itens e 20 examinados, os valores sorteados de $\delta_j = (a_j, b_j, c_j)'$ e $\Theta_i = (\theta_1, \theta_2, \dots, \theta_{20})'$ são:

Tabela 1 Exemplo Contendo Valores Sorteados de Parâmetros de 20 Itens.

Item	a_i	b_i	c_i
1	0,7670	1,0882	0,1053
2	1,0709	1,6952	0,2269
3	0,3977	-0,2393	0,2216
4	0,8344	0,3292	0,2219
5	0,2455	-0,0250	0,2209
6	0,2973	1,0074	0,2630
7	1,0448	0,3178	0,1457
8	1,0808	0,2417	0,2360
9	0,2673	2,7798	0,2361
10	0,2769	1,2707	0,2592
11	1,2585	-0,2322	0,3158
12	0,7731	0,1617	0,2307
13	0,4325	-0,6199	0,2717
14	0,2159	-0,0684	0,2268
15	0,3698	0,1520	0,2405
16	0,1998	0,4235	0,1869
17	0,7403	-0,8100	0,2644
18	0,6988	-0,5284	0,2300
19	1,1879	-1,1781	0,2270
20	0,5069	0,4960	0,2429

Tabela 2 Exemplo de Habilidades Sorteadas de 15 Examinados.

Examinados	θ_j
1	0,7231
2	0,2087
3	0,2856
4	-0,4056
5	0,1332
6	-0,7901
7	0,5332
8	0,1332
9	-0,2430
10	0,4469
11	0,5332
12	-0,2430
13	-0,6842
14	-0,1658
15	-0,4056

Dessa forma, a probabilidade do examinado 1 acertar o item 1 é dada por:

$$P(Y_{11} = 1 | \theta = \theta_1; \delta_1) = c_1 + (1 - c_1) \frac{1}{1 + e^{B(-a(\theta_1 - b_1))}}, \text{ tem-se que:}$$

$$\delta_1 = (0,7670, 1,0882, 0,1053) \text{ e } \theta_1 = 0,7231. \text{ Logo:}$$

$$P(Y_{11} = 1 | \theta = 0,7231; \delta_1) =$$

$$0,1053 + (1 - 0,1053) \frac{1}{1 + e^{1,0882(-0,7670(0,7231 - 1,0882))}} = 0,4480$$

A probabilidade calculada de acerto do examinado 1 ao item 1 é 0,4480. Suponha-se que o valor sorteado da uniforme (0,1) seja 0,70, dessa forma, a resposta dada pelo examinado será errada, ou seja, será atribuído o valor 0. Agora se ocorresse o valor 0,3, no sorteio da uniforme (0,1), a resposta dada pelo examinado seria correta, logo seria atribuído o valor 1. Dessa forma, construiu-se uma tabela de zeros e uns contendo a resposta de todos os examinados aos itens, como mostra a tabela a seguir.

Tabela 3 Tabela de respostas de 20 Examinados a 20 Itens.

<u>exam.</u>																				
<u>itens</u>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0
4	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1
5	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	0	0	0
6	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1
7	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
8	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0
9	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	1
10	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0
11	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0
12	1	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
13	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
14	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1
15	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0
17	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1

18	1	0	1	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0
19	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1
20	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1

Finalmente, utilizando as respostas dos examinados aos itens, ou seja, a tabela de zero e uns, construída conforme descrito anteriormente, obteveram-se com o ICL, as estimativas para os parâmetros a , b , c e θ , sendo que o parâmetro θ foi estimado por dois métodos: verossimilhança e Bayesiano.

Todos os itens analisados possuíam cinco alternativas. Para cada conjunto de parâmetros de itens e habilidades dos examinados (como os da tabela 1), foram simuladas as respostas dos examinados para o teste 1000 vezes, ou seja, foram construídas 1000 tabelas como a 2.

Como os valores verdadeiros dos parâmetros eram conhecidos, realizou-se a comparação entre os verdadeiros valores e os estimados pelo ICL, através do erro quadrático médio e viés, que são medidas capazes de comparar o quanto os valores estimados, estão próximos dos verdadeiros.

Nas estimações feitas pelo ICL, usou-se a configuração do padrão do software, exceto o intervalo da variável latente que foi de -6 a 6, o critério de convergência que foi de 10000 iterações ou um erro menor que 0,000001 e a quantidade de intervalos de divisão da variável latente, que foi 200.

3.1 Software utilizado para realizar as simulações

Para realizar as simulações, foi desenvolvido um programa em Python, os scripts do programa encontram-se no Anexo A. Sem tal programa, a realização das simulações era inviável, pois o ICL necessita de dois códigos distintos para realizar a estimação dos parâmetros dos itens (calibração) e a

habilidades, ou seja, nele esse processo é feito em duas etapas, primeiro estima-se os parâmetros dos itens e depois utilizando esses valores, estima-se a habilidade dos examinados. Tendo em vista que fizemos milhares de simulações, fazer esse processo manualmente seria extremamente trabalhoso e demorado.

O programa desenvolvido realiza os seguintes processos:

- 1- Sorteio dos parâmetros dos itens e habilidades dos examinados de acordo com as distribuições pré-determinadas.
- 2- Construção das respostas dos examinados ao teste.
- 3- Recuperação dos parâmetros dos itens e habilidades dos examinados através do ICL.
- 4- Cálculo do erro quadrático médio e viés entre os valores sorteados e os valores estimados pelo ICL. Tais valores são apresentados em tabelas que contém também a média dos valores estimados para cada conjunto de parâmetros de itens e habilidade dos examinados, juntamente com o valor real sorteado.
- 5- O programa ainda possibilita a mudança dos valores das médias e variância dos parâmetros dos itens e habilidade dos examinados.

Em algumas simulações observaram-se alguns erros ocorridos na estimação feita pelo ICL, assim o programa excluiu esses erros de todas as análises estatísticas, sendo que em todas as simulações mais de 99% das simulações foram válidas.

3.2 Situações analisadas

O comportamento do ICL foi analisado frente aos seguintes fatores:

- a) Número de itens
- b) Número de examinados
- c) Variação dos valores do parâmetro de discriminação dos itens.

d) Variação dos valores de dificuldade dos itens.

Para estudar o comportamento do software, foi realizado um experimento em um esquema fatorial onde, aumentava-se o número de itens e o número de examinados, em 3 situações diferentes, conforme mostra a tabela a seguir.

Tabela 4 Tabela com esquema de análise dos fatores, sendo que S1, S2 e S3 representam as 3 situações analisadas.

Itens						
Examinados	40		80		120	
40	40 Itens e 40 Examinados	S1	80 Itens e 40 Examinados	S1	120 Itens e 40 Examinados	S1
		S2		S2		S2
		S3		S3		S3
80	40 Itens e 80 Examinados	S1	80 Itens e 80 Examinados	S1	120 Itens e 80 Examinados	S1
		S2		S2		S2
		S3		S3		S3
120	40 Itens e 120 Examinados	S1	80 Itens e 120 Examinados	S1	120 Itens e 120 Examinados	S1

A análise dos fatores foi realizada em 3 situações distintas, sendo que, em cada situação, os testes continham as combinações entre 40, 80 e 120 itens e 40, 80 e 120 examinados. A escolha de tais valores tentou reproduzir situações reais de números examinados e de itens.

1ª Situação:

Nesta situação, os parâmetros a foram sorteados a partir de uma distribuição log-normal com média zero e desvio padrão 0.1. Foi usada tal distribuição visto que, 99% de seus valores estão entre 0,7 e 1,3 logo, estão na faixa de uma discriminação moderada, ou seja, são curvas que discriminam em uma maior faixa de habilidade.

Para o sorteio dos parâmetros b e θ , foi utilizada a distribuição normal, o uso de tal distribuição se deve ao fato de esperar-se que, à medida que se aumenta a quantidade de itens de um teste, os valores de dificuldade tenham essa distribuição.

Para o sorteio do parâmetro c , atribui-se a esse parâmetro uma distribuição normal (0,2;0,0025). A média de 0,2 deve-se à análise clássica onde, os examinados com baixa habilidade possuem uma probabilidade de acerto de $\frac{1}{N}$, onde N é o número de itens do teste. Em todas as simulações, os testes continham 5 itens, logo a probabilidade de acerto era $\frac{1}{5} = 0,2$. Já a pequena variância de 0,0025 reflete o fato de que, em um item bem elaborado, não há a exclusão de nenhuma alternativa pelo examinado, o que poderia aumentar a probabilidade de acerto de examinados com baixa habilidade. Logo, espera-se que a probabilidade de acerto casual fique bem próximo de $\frac{1}{N}$, com pouca variação.

2ª Situação:

Nesta situação, o objetivo era verificar se uma maior variação na discriminação dos itens poderia afetar a qualidade das estimações feitas pelo software, visto que, curvas com maiores valores do parâmetro a , discriminam

muito bem os examinados, mas em um pequeno intervalo. Para isso, atribuiu-se ao parâmetro a uma distribuição log-normal com média zero e variância 0,5 que possui 91% de seus valores até 2 e 98% até 3, ou seja, 1 com esta distribuição obteve-se mais curvas com alta discriminação. As distribuições para b , c e θ foram as mesmas que na situação 1.

3ª Situação:

Esta situação teve como objetivo verificar o comportamento do software frente a maiores valores do parâmetro b , por isso atribuiu-se à esse parâmetro uma distribuição normal com média zero e variância 1,5. Para estudar o comportamento do parâmetro b , sem possíveis interferências do parâmetro a , diminuiu-se a variação de a , atribuindo para esse parâmetro uma distribuição normal com média 1 e variância 0,001 na qual 99% dos valores estão entre 0,92 e 1,07. As distribuições de c e θ foram as mesmas da primeira situação.

4 RESULTADOS E DISCUSSÃO

A seguir, encontram-se os gráficos de Erro Quadrático Médio e Viés das estimativas feitas pelo ICL para cada parâmetro e os valores verdadeiros dos parâmetros, previamente conhecidos. O Viés e o Erro Quadrático Médio foram calculados a partir de:

$$\text{Viés Médio} = \frac{\sum_{i=1}^n \sum_{k=1}^k (\beta_i - \hat{\beta}_{ik})}{n \cdot k}$$

$$\text{Erro Quadrático Médio} = \frac{\sum_{i=1}^n \sum_{k=1}^k (\beta_i - \hat{\beta}_{ik})^2}{n \cdot k}$$

Sendo, β valor verdadeiro do parâmetro, $\hat{\beta}$ o valor estimado pelo ICL, k o número de repetições válidas em cada simulação, ou seja, se não houver nenhuma perda $k=1000$ e n , é o número de itens do teste, caso o parâmetro estimado seja a, b ou c . Se o parâmetro estimado for θ , n é o número de respondentes, mas em ambos os casos $n=40, 80$ ou 120 .

4.1 Resultados da situação 1

Lembramos que neste caso o parâmetro a tem distribuição log-normal com média 0 e desvio padrão 0,1; b e θ têm distribuição normal com média zero e variância 1 e c tem distribuição normal com média 0,2 variância 0,0025.

A) Resultados para o parâmetro α :

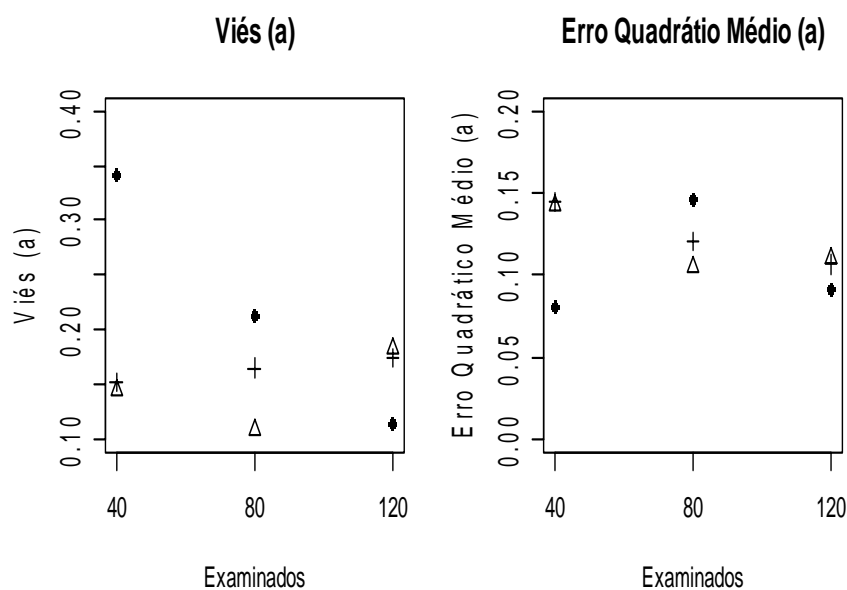


Figura 11 Gráfico do viés e erro quadrático médio do parâmetro α , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens

Observa-se que, para o parâmetro α , nas condições estudadas, o ICL subestima em média 0,0835, o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de α . Além disso, através do gráfico do erro quadrático médio e do viés, conclui-se que, o aumento do número de examinados não influenciou a estimação realizada pelo software para este parâmetro.

B) Resultados para o parâmetro b :

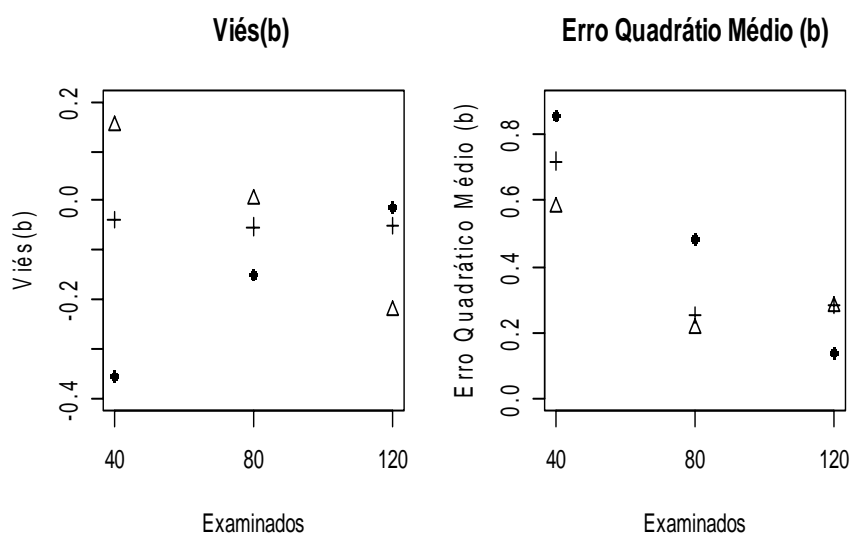


Figura 12 Gráfico do viés e do erro quadrático médio para o parâmetro b , sendo que os símbolos representam respectivamente: ● 40 itens, Δ 80 itens e + 120 itens

Observa-se que, para o parâmetro b , nas condições estudadas, o ICL superestima em média, 0,0701 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de b . Além disso, através do gráfico do erro quadrático médio conclui-se que, o aumento do número de examinados tornou mais precisas as estimações do software, pois observou-se

que os valores do erro quadrático médio diminuíram à medida que o número de examinados aumentou.

Já no gráfico do viés, não se observou nenhuma regularidade com o aumento do número de examinados.

C) Resultados para o parâmetro c:

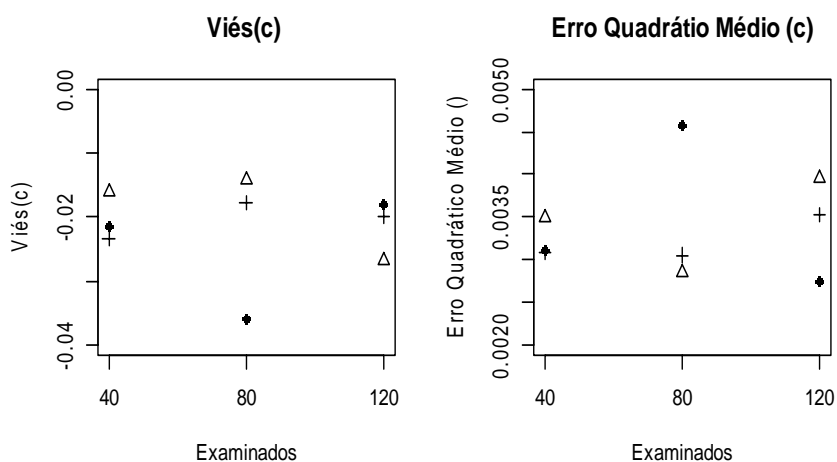


Figura 13 Gráfico do viés e erro quadrático médio para o parâmetro c, sendo que os símbolos representam respectivamente: ● 40 itens, Δ 80 itens e + 120 itens

Observa-se que, para o parâmetro c , nas condições estudadas, o ICL superestima em média 0,0215 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de c . Além disso, através dos gráficos do erro quadrático médio e viés, não se observou qualquer influência devida ao aumento do número de examinados.

D) Resultados para o parâmetro θ estimado por verossimilhança

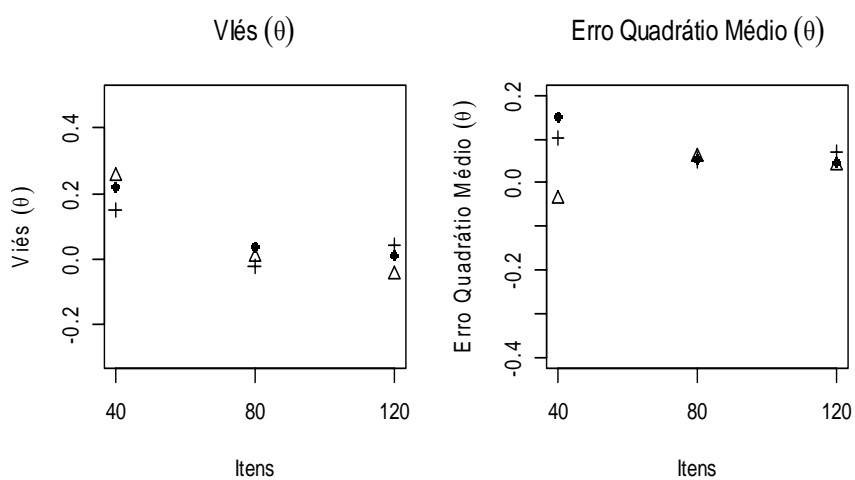


Figura 14 Gráfico do viés e do erro quadrático médio para o parâmetro θ , estimado por verossimilhança, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados

Observa-se que, para o parâmetro θ , nas condições estudadas, o ICL subestima em média 0,0718 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de θ . Além disso, analisando os gráficos do erro quadrático médio e viés, conclui-se que o aumento do número de itens aumentou a precisão das estimativas feitas pelo software.

E) Resultados para o parâmetro θ estimado pelo método bayesiano

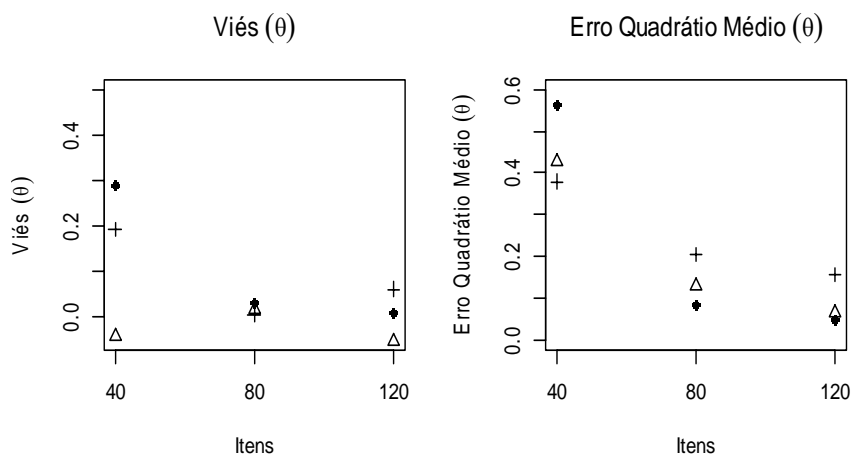


Figura 15 Gráfico do viés e do erro quadrático médio para o parâmetro θ pelo método Bayesiano, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados

Observa-se que, para o parâmetro θ , nas condições estudadas, o ICL subestima em média 0,0584 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de θ . Além disso, através do gráfico do viés e do erro quadrático médio, conclui-se que o aumento do número de itens aumentou a precisão das estimativas feitas pelo software, visto que, tanto o viés

quanto o erro quadrático médio diminuíram à medida que aumentaram o número de itens.

Em comparação com o método de estimação por verossimilhança, apesar do viés médio do bayesiano ter sido menor, podemos notar que a variação do método Bayesiano é maior, tanto no gráfico do viés, quanto no do erro quadrático médio, por isso, podemos afirmar que, o método Bayesiano tem menor precisão nas estimativas que o de verossimilhança, para as condições estudadas.

F) Considerações gerais da situação 1

Segundo Andrade, Tavares & Valle (2000), os estimadores de máxima verossimilhança dos parâmetros dos itens e da habilidade dos examinados são não- viciados quando o número de examinados e itens cresce.

Na situação 1 para a estimação de θ , tanto pelo método Bayesiano, quanto por verossimilhança, verificou-se que, à medida que o número de itens aumentava, o viés e o erro quadrático médio diminuíram. No entanto, com relação às estimativas dos parâmetros, não se observou isso, somente houve mais precisão das estimativas no parâmetro b , onde se observou uma diminuição do erro quadrático médio.

Tal fato pode ter ocorrido, devido ao tamanho da amostra tomada, pois trabalhos como o de Nunes (2005) mostram que, em amostras a partir de 200 examinados e itens, conseguem-se valores próximos dos verdadeiros para os parâmetros θ e b , sendo que para o parâmetro a , foram obtidos os piores resultados com esse tamanho de amostra. O resumo da situação 1 está nas Tabelas a seguir:

Tabela 5 Resumo dos resultados da situação 1.

Parâmetro	Considerações
Estimado	
α	Foi subestimado em média 0,0835 e não mostrou melhores resultados com o aumento do número de examinados, pois nem o viés nem o erro quadrático médio diminuíram.
b	Foi superestimado em média, 0,0701 e à medida que o número de examinados aumentou, houve diminuição do erro quadrático médio, porém o viés não alterou.
c	Foi superestimado em média, 0,0215 e não mostrou melhores resultados com o aumento do número de examinados, pois o viés e o erro quadrático médio não diminuíram.
θ por verossimilhança	Foi subestimado em média 0,0718 e à medida que o número de itens aumentou, o erro quadrático médio e o viés diminuíram. O intervalo do erro quadrático médio foi menor que método Bayesiano.

θ pelo método Bayesiano	Foi subestimado em média 0,0584 e apresentou menor erro quadrático médio e viés, à medida que aumentaram a quantidade de itens, porém os valores foram maiores que por verossimilhança.
--------------------------------	---

Tabela 6 Comparação entre o viés e a amplitude dos valores sorteados na situação 1.

Parâmetro	Valores máximos e mínimos sorteados	Amplitude	Viés	Varição do viés em relação à amplitude
a	Mín= 0,6955 Máx=1,4733	0,7778	0,0835	10,7354%
b	Mín= -2,5129 Máx=2,1086	4,6215	0,0701	1,5166%
c	Mín= 0,0226 Máx=0,3864	0,3638	0,0215	5,9098%
θ por verossim.	Mín=-2,4025 Máx= 2,2156	4,6181	0,0718	1,5547%
θ pelo método Bayesiano	Mín= -2,4025 Máx=2,2156	4,6181	0,0584	1,2646%

O tamanho do viés comparado com a amplitude de variação dos dados foi muito baixo para os parâmetros b e θ , visto que na estatística frequentemente considera-se que, um erro menor que 5%, é aceitável. Dessa forma, nas condições estudadas, o ICL estima com boa precisão, os parâmetros b e θ , já os parâmetros a e c foram estimados com baixa precisão.

4.2 Resultados da situação 2

O objetivo desta situação foi verificar o comportamento do ICL frente a valores maiores e mais variáveis do parâmetro a , por isso, se atribuiu para este parâmetro uma distribuição log-normal com média zero e variância 0,5. Os demais parâmetros tiveram a mesma distribuição da primeira situação.

A) Resultados para o parâmetro a :

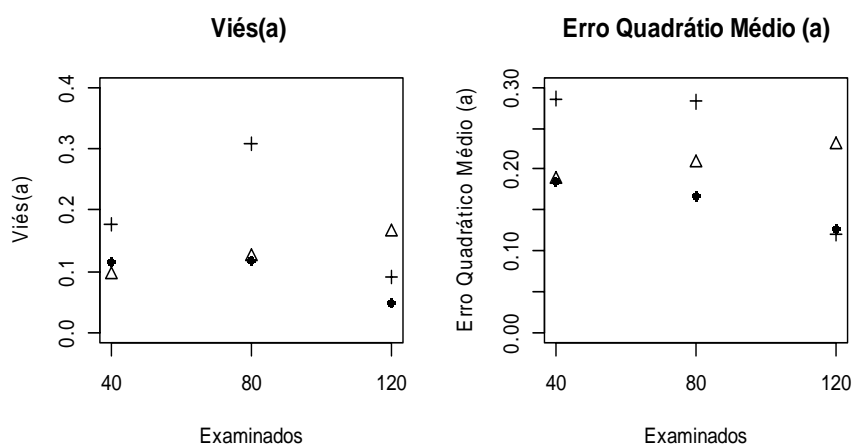


Figura 16 Gráfico do viés e erro quadrático médio do parâmetro a, sendo que os símbolos representam respectivamente: ● 40 itens, Δ 80 itens e + 120 itens

Observa-se que, para o parâmetro a , nas condições estudadas, o ICL subestima em média 0,1373 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de a . Além disso, através do gráfico do viés e do erro quadrático médio, conclui-se que o aumento do número de examinados não influenciou as estimativas feitas pelo ICL.

B) Resultados do parâmetro b :

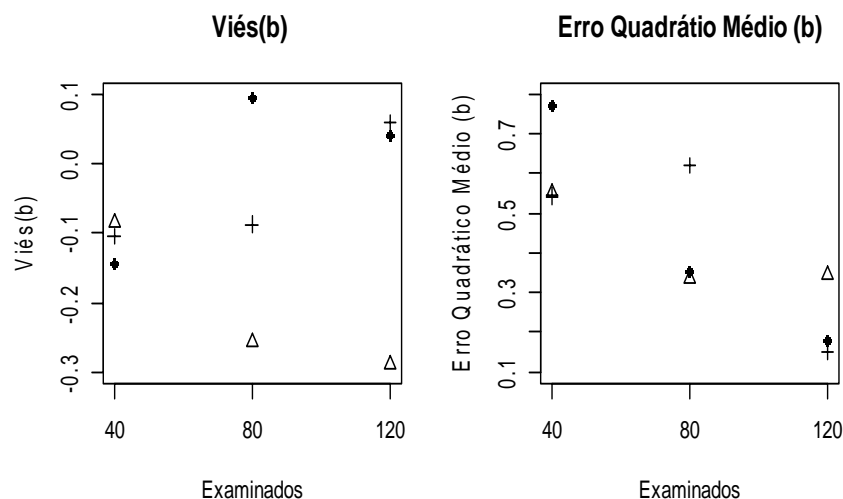


Figura 17 Gráfico do viés e do erro quadrático médio para o parâmetro b, sendo que os símbolos representam respectivamente: ● 40 itens, Δ 80 itens e + 120 itens

Observa-se que, para o parâmetro b , nas condições estudadas, o ICL superestima em média 0,0859 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de b . Além disso, através do gráfico do erro quadrático médio, conclui-se que o aumento do número de examinados aumentou a precisão das estimativas feitas pelo software, pois o erro quadrático médio diminuiu. No gráfico do viés, não se observou nenhuma regularidade com o aumento de examinados.

C) Resultados do parâmetro C:

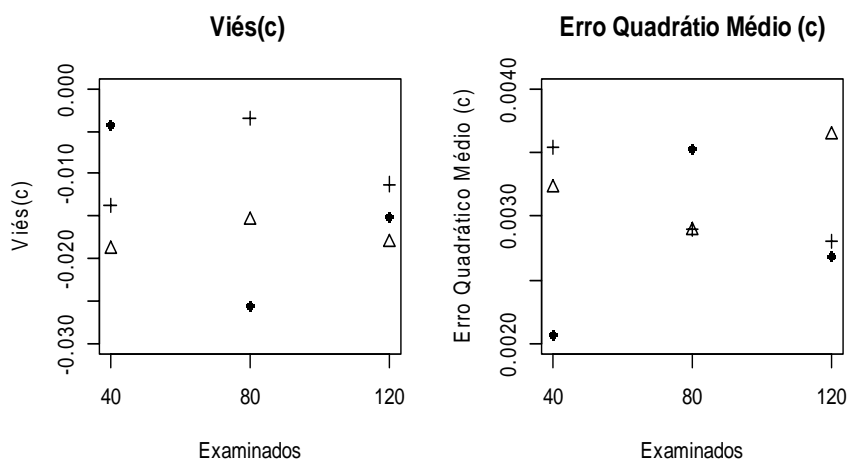


Figura 18 Gráfico do viés e do erro quadrático médio para o parâmetro c , sendo que os símbolos representam respectivamente: ● 40 itens, Δ 80 itens e + 120 itens

Observa-se que, para o parâmetro ϵ , nas condições estudadas, o ICL superestima em média 0,0140 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de ϵ . Através dos gráficos do erro quadrático médio e do viés, conclui-se que o aumento do número de examinados não influenciou a precisão das estimativas feitas pelo software.

D) Resultados para θ estimado por verossimilhança:

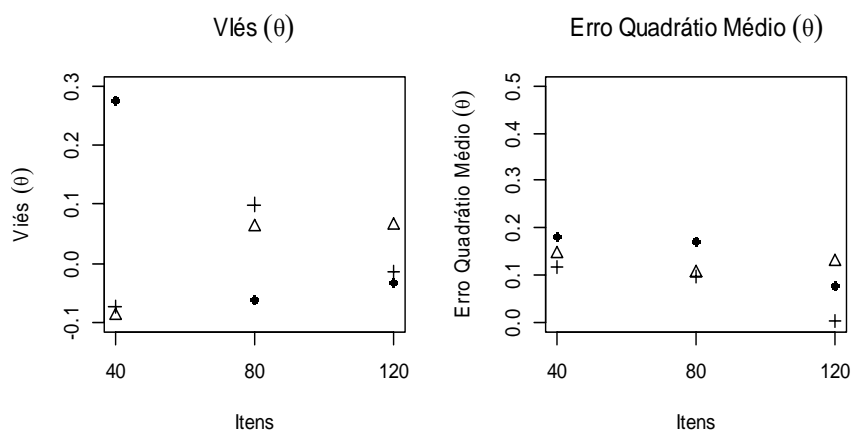


Figura 19 Gráfico do viés e erro quadrático médio do parâmetro θ estimado por verossimilhança, sendo que os símbolos representam respectivamente: ● 40 examinados, Δ 80 examinados e + 120 examinados

Observa-se que, para o parâmetro θ , nas condições estudadas, o ICL superestima em média 0,0089 o parâmetro verdadeiro, esse valor foi encontrado através da média de todos os vieses de θ . Além disso, através dos gráficos do erro quadrático médio e do viés, conclui-se que, com o aumento do número de itens o erro quadrático médio e o viés diminuíram.

E) Resultados para θ estimado pelo método Bayesiano:

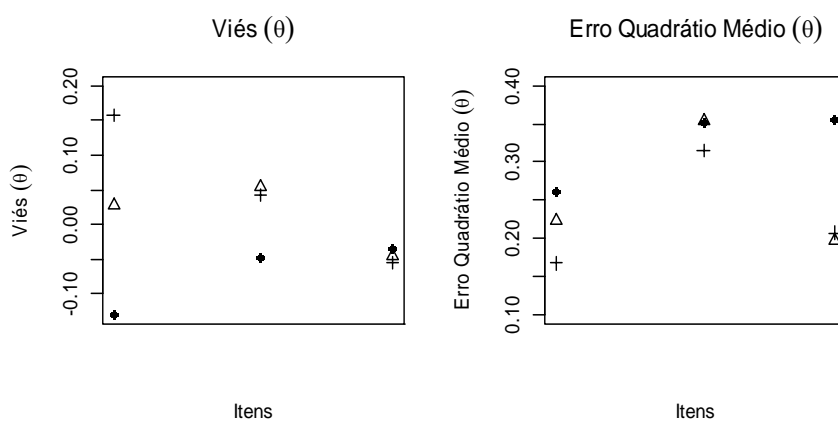


Figura 20 Gráfico do viés e do erro quadrático médio para o parâmetro θ estimado pelo método Bayesiano, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados

Observa-se que, o parâmetro θ estimado pelo método Bayesiano, nas condições estudadas, foi em média subestimado pelo ICL em 0,0062, esse valor foi encontrado através da média de todos os vieses de θ . Além disso, através do gráfico do viés, observa-se que com o aumento do número de itens o viés diminuiu, porém não observou-se diminuição no erro quadrático médio com o aumento de itens.

Novamente nesta situação, verifica-se a existência de valores maiores de erro quadrático médio pelo método Bayesiano que por Verossimilhança, logo, assim como para na situação 1, conclui-se que as estimativas por Verossimilhança são mais precisas.

F) Considerações gerais da situação 2:

O objetivo dessa situação foi observar o comportamento das estimativas do software frente a maiores valores do parâmetro α . O resumo da situação 2 está nas tabelas a seguir:

Tabela 7 Resumo da Situação 2.

Parâmetro	Considerações
Estimado	
α	Foi subestimado em média 0,1373, o viés e o erro quadrático médio não diminuíram com o aumento do número de examinados. Em comparação com a situação 1, apresentou maior viés médio e maiores valores de erro quadrático médio.
b	Foi superestimado em média 0,0859 e à medida que o número de examinados aumentou, houve diminuição do erro

	quadrático médio, porém isso não aconteceu no viés.
c	Foi superestimado em média, 0,0140 e não mostrou melhores resultados com o aumento do número de examinados.
θ por verossim.	Foi subestimado em média 0,0089, o erro quadrático médio e viés diminuíram à medida que aumentaram a quantidade de itens, teve menor erro quadrático médio que a estimação pelo método Bayesiano.
θ pelo método Bayesiano	Foi subestimado em média 0,0062, o viés diminuiu à medida que aumentaram a quantidade de itens porém, o erro quadrático médio não diminuiu como na situação 1, os valores de erro quadrático médio foram maiores que por verossimilhança.

Tabela 8 Comparação entre o viés e a amplitude dos valores sorteados na situação 2.

Parâmetro	Valores máximos e mínimos sorteados	Amplitude	Viés	Varição do viés em relação à amplitude.
a	Mín=0,2431 Máx=3,323 3	3,0802	0,1373	4,4575%
b	Mín= -2,5815 Máx=3,1370	5,7185	0,0859	1,5021%
c	Mín= 0,0536 Máx=0,3399	0,2863	0,0140	4,8900%
θ por verossim.	Mín= -4,1805 Máx= 2,4824	6,6629	0,0089	0,1336%

θ	pelo	Mín= -4,1805	6,6629	0,0062	0,0930%
método		Máx=2,4824			
Bayesiano					

A variação do viés médio, comparada à amplitude de variação dos valores sorteados foi menor, para todos os parâmetros, se comparado à situação 1, destacando-se que, a variação foi menor que 5% para todos os parâmetros.

Assim como na situação 1, os vieses dos parâmetros de itens não diminuíram com o aumento do número de examinados, já nos parâmetros θ , isso foi observado. Porém, Nunes (2005) menciona que, tal comportamento é observado para amostras de tamanho 200 e o maior valor amostrado nesta situação foi de 120, ou seja, talvez a amostra tenha sido pequena para a observação desta regularidade.

4.3 Resultados da situação 3

Esta situação teve como objetivo verificar o comportamento das estimações do ICL frente à maior variação do parâmetro b . Para isso, atribuiu-se a esse parâmetro uma distribuição normal com média zero variância 1,5. Para estudar o comportamento do parâmetro b , sem possíveis interferências do parâmetro a , diminuiu-se a variação de a , atribuindo para esse parâmetro uma distribuição normal com média 1 e variância 0,001 na qual 99% dos valores estão entre 0,92 e 1,07. A distribuição de c continuou normal com média 0,2 e variância 0,0025.

A) Resultados para o parâmetro α .

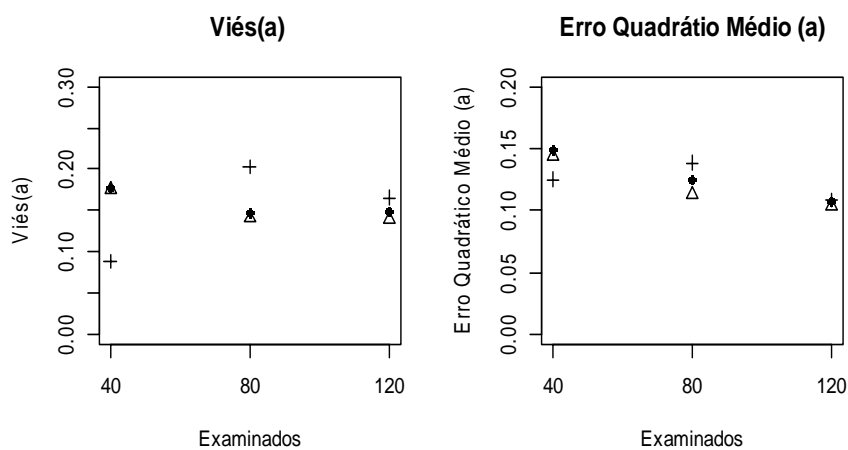


Figura 21 Gráfico do Viés e do Erro Quadrático Médio para o parâmetro α , sendo que os símbolos representam respectivamente: ● 40 itens, Δ 80 itens e + 120 itens

Observa-se que, o ICL subestima o parâmetro α 0,1536 em média, nas condições estudadas, esse valor foi encontrado através da média de todos os vieses de α . Além disso, através do gráfico do erro quadrático médio, conclui-se que, o aumento do número de examinados tornou os resultados do software mais precisos, já no gráfico do viés o aumento de examinados não diminuiu os vieses.

B) Resultados do parâmetro β .

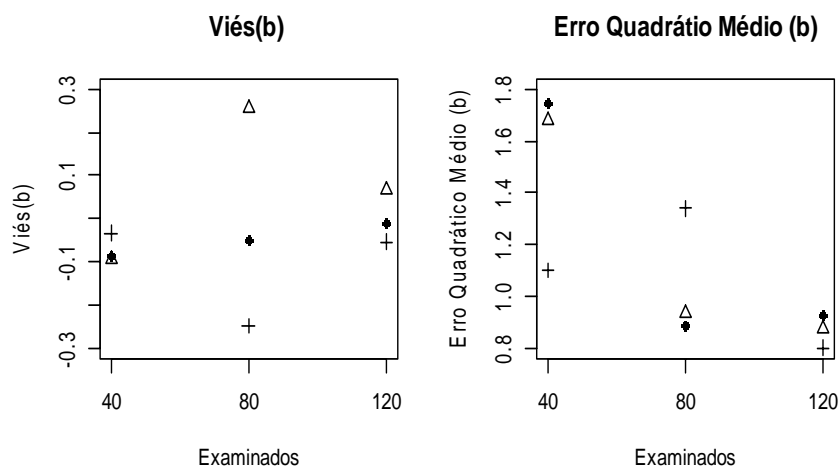


Figura 22 Gráfico do Viés e do Erro Quadrático Médio para o parâmetro b , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens

Observa-se que, o ICL superestima o parâmetro b em média 0,0288 nas condições estudadas, esse valor foi encontrado através da média de todos os vieses de b . Além disso, através dos gráficos do viés e do erro quadrático médio, não se observa que o aumento do número de examinados tornou os resultados do software mais precisos.

C) Resultados do parâmetro c .

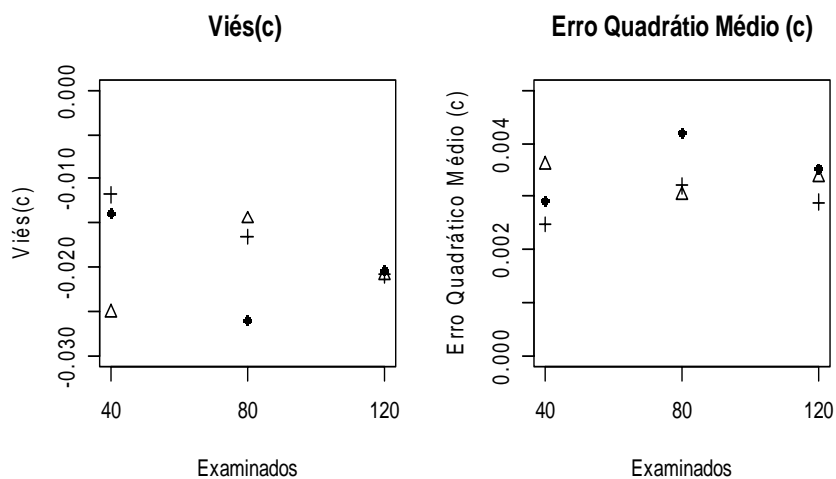


Figura 23 Gráficos do viés e do erro quadrático médio para o parâmetro c , sendo que os símbolos representam respectivamente: \bullet 40 itens, Δ 80 itens e $+$ 120 itens

Observa-se que o ICL superestima em média o parâmetro c em 0,0189, esse resultado foi obtido através da média de todos os vieses. Além disso, através dos gráficos do erro quadrático médio e do viés, conclui-se que, com o aumento do número de examinados o viés e o erro quadrático médio não diminuíram.

D) Resultados do parâmetro θ estimado por verossimilhança:

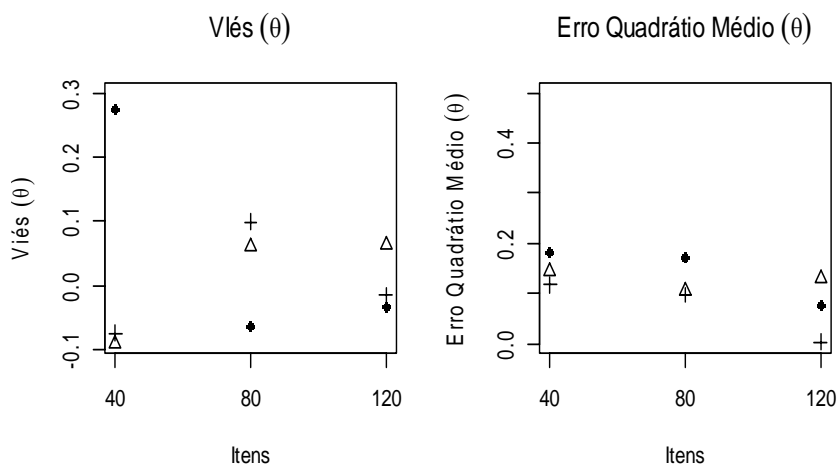


Figura 24 Gráfico do viés e erro quadrático médio para o parâmetro θ estimado por verossimilhança, sendo que os símbolos representam respectivamente: ● 40 examinados, Δ 80 examinados e + 120 examinados

Observa-se que o ICL superestima em média o parâmetro θ em 0,0311, esse resultado foi obtido através da média de todos os vieses. Além disso, conclui-se que, com o aumento do número de itens, o viés e o erro quadrático médio diminuiram.

E) Resultados para o parâmetro θ estimado pelo método Bayesiano.

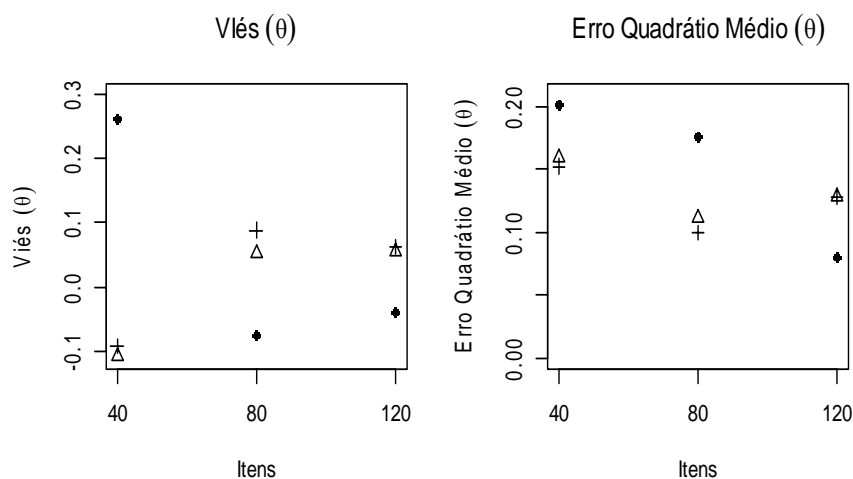


Figura 25 Gráfico do viés e do erro quadrático médio para o parâmetro θ Estimado pelo método Bayesiano, sendo que os símbolos representam respectivamente: \bullet 40 examinados, Δ 80 examinados e $+$ 120 examinados

Observa-se que o ICL superestima em média o parâmetro θ em 0,0255, esse resultado foi obtido através da média de todos os vieses. Além disso, com o aumento do número de itens, o erro quadrático médio e viés diminuiram.

Em comparação com a estimação de θ por verossimilhança, observamos que a variação do erro quadrático médio é menor, logo as estimativas por verossimilhança são mais precisas.

F) Considerações Gerais da Situação 3.

Nesta situação, o aumento do número de itens diminuiu o erro quadrático médio e o viés nas estimações de θ , já nos parâmetros a , b e c não se observou que o aumento de examinados diminuiu o viés. O método de

verossimilhança se mostrou mais preciso que o Bayesiano, assim como nas situações 1 e 2. O resumo da situação 3 está nas tabelas a seguir.

Tabela 9 Resumo da situação 3.

Parâmetro Estimado	Considerações
a	Foi subestimado em média 0,1536, com o aumento do número de examinados, o erro quadrático médio diminuiu, porém o viés não.
b	Foi superestimado em média 0,0288 e à medida que o número de examinados aumentou, não houve diminuição nem do erro quadrático médio nem do viés.
c	Foi superestimado em média 0,0189 e o aumento do número de examinados não diminuiu o erro quadrático médio nem o viés.
θ por verossimilhança	Foi subestimado em média 0,0311, o erro quadrático médio e o viés diminuíram à medida que aumentaram a quantidade de itens, teve menor erro quadrático médio que a estimação pelo método Bayesiano.
θ pelo método Bayesiano	Foi subestimado em média 0,0255, o erro quadrático médio e viés diminuíram à medida que aumentaram a quantidade de itens. Os valores do erro quadrático foram maiores que por verossimilhança.

Tabela 10 Comparação entre o viés e a amplitude dos valores sorteados na situação 3

Parâmetro	Valores máximos e mínimos sorteados	Amplitude	Viés	Variação do viés em relação à amplitude
α	Mín= 0,091 Máx=1,084 2	1,7132	0,1536	8,9600%
b	Mín= -3,7792 Máx= 4,4247	8,2039	0,0288	0,3510%
c	Mín= 0,0577 Máx=0,3417	0,284	0,0189	6,6540%
θ por verossim.	Mín= -1,8909 Máx= 1,4360	3,3269	0,03112	0,9354%
θ pelo método Bayesiano	Mín= -4,1805 Máx=2,4824	6,6629	0,0255	0.3827%

O tamanho do viés comparado com a amplitude de variação dos dados foi menor que 1% para os parâmetros b e θ . Já para os parâmetros α e c , obtiveram-se valores maiores que 5%, assim como na primeira situação. Dessa forma, conclui-se que, para esta situação, os valores de α e c tiveram altos erros na estimação.

Tabela 11 Vieses encontrados para os parâmetros a , b , c e θ nas 3 situações estudadas.

S	Viés do parâmetro a	Viés do parâmetro b	Viés do parâmetro c	Viés do parâmetro θ por Verossim.	Viés do parâmetro θ por Bayesiana
1	0,0835	-0,0701	-0,0215	0,0718	0,0584
2	0,1374	-0,0859	-0,0140	0,0089	0,0062
3	0,1536	-0,0289	-0,0189	0,0311	0,0255

A estimação de todos os parâmetros pelo ICL mostrou valores de viés, porém não podemos concluir que o software é viesado, pois não houve controle do gerador de números aleatórios utilizado, sendo, portanto, necessários mais estudos neste sentido.

5 CONCLUSÃO

Conclui-se que, com o aumento do número de itens, os valores de viés e erro quadrático médio diminuíram na estimação das habilidades dos examinados, tanto pelo método de máxima verossimilhança quanto pelo Bayesiano.

A estimação por verossimilhança, em todas as três situações, mostrou-se mais precisa que pelo método Bayesiano, visto que, os intervalos de variação dos erros quadráticos médios foram menores.

Não se observou que os valores de viés e erro quadrático médio, para os parâmetros a , b e c , diminuíram à medida que se aumentou o número de examinados, tal fato pode ter ocorrido devido ao pequeno tamanho da amostra analisada.

Os melhores resultados foram obtidos na segunda situação, onde se observou uma maior variação do parâmetro a . Os parâmetros θ e b tiveram baixos erros na estimação em todas as situações estudadas. A maior variação de b não influencia as estimativas do ICL.

Dessa forma, a partir dos dados analisados, conclui-se que o ICL, nas situações analisadas, mostrou-se um software que produz estimativas para os parâmetros muito próximas das verdadeiras, sendo que os melhores resultados são obtidos na estimação dos parâmetros θ e b .

REFERÊNCIAS

- ANDRADE, D.F.; TAVARES, H.R.; VALLE, R.C. **Teoria de Resposta ao Item**: conceitos e aplicações. Minas Gerais: SINAPE, 2000.
- BAKER, F.B. **Item Response Theory - Parameter Estimation Techniques**. New York: Marcel Dekker, Inc, 2001.
- BIRNBAUM, A. **Some latent trait models and their use in inferring an examinee's ability**. In statistical Theories of Mental Test Scores, F. M. Lord and M. R. Novick (Eds.). Reading, M.A.: Addison-Wesley, 1968.
- BRAGION, M.L.L.de. **Análise combinada de exames vestibulares da Universidade Federal de Lavras usando a teoria de resposta ao item. 2010**. 187 p. Tese (Doutorado em Estatística e Experimentação Agropecuária), Universidade Federal de Lavras, Lavras, 2010.
- HAMBLETON, R.K.; SWAMINATHAN, H.; ROGERS, H.J. **Fundamentals of Item Response Theory**. Newbury Park: Sage Publications, 2001.
- HALEY, D.C. **Estimation of the dosage mortality relationship when the dose is subject to error**. Technical Report N° 15, August 29, 1952. Stanford, Calif.: Contract N° ONR-25140, Applied Mathematics and Statistics Laboratory, Stanford University.
- HANSON, B.A. **IRT Parameter Estimation using the EM Algorithm**. 1998. Disponível em: <<http://www.b-a-h.com/papers/note9801.html>>. Acesso em: 14 fev. 2012.
- HANSON, B.A. **IRT Command Language Manual**. 2002. Disponível em: <<http://www.b-a-h.com/software/irt/icl/>>. Acesso em: 14 fev. 2012.
- INEP - Guia de Elaboração e Revisão de Itens, v.1, 2010.

JURICH & GOODMAN. **A Comparison of IRT Parameter Recovery in Mixed Format Examinations Using PARSALE and ICL.** Illinois Institute of Technology, 2009.

MEAD, MORRIS & BLITZ. Open-source IRT: **A comparison of BILOG-MG and ICL features and item parameter recovery.** Illinois Institute of Technology, 2007.

MOREIRA JR., F. J. Aplicações da teoria da resposta ao item no Brasil. **Rev. Bras. Biom.**, v.28, n.4, São Paulo, 2010, p.137-170.

LORD, F. M.; NOVICK, M. R. **Statistical Theories of Mental Test Score.** Reading: Addison-Wesley, 1968.

LORD, F. M. **Applications of Item Response Theory to Practical Testing Problems.** Hillsdale: Lawrence Erlbaum Associates, 1980.

NUNES, C. H. **Impacto do Tamanho da Amostra na Calibração de Itens e Estimativa de Escores por Teoria de Resposta ao Item.** Aval. Psicol, v. 4v, n. 2, nov. 2005.

PASQUALI, L. **Fundamentos da Teoria de Resposta ao Item.** Avaliação Psicológica, 2003.

WOODRUFF, D. J.; HANSON, B. A. **Estimation of item response models using the EM algorithm for finite mixtures.** Paper presented at the Annual Meeting of the Psychometric Society (Gatlinburg, Tennessee, June). 1997. Disponível em: <http://www.b-a-h.com/papers/paper9701.html>. Acesso em: 14 fev. 2012.

ANEXOS

Sripts do Programa em Phyton Elaborado para as Simulações.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

#
#     Simulacao do software icl com a finalidade de
#     atestar o mesmo
#
#     File: testarICL.py
#
#
#     Copyright 2012 Thiago Dias Simao
#     <thiago_dias@computacao.ufla.br>
#
#     This program is free software; you can redistribute
#     it and/or modify
#     it under the terms of the GNU General Public
#     License as published by
#     the Free Software Foundation; either version 2 of
#     the License, or
#     (at your option) any later version.
#
#     This program is distributed in the hope that it
#     will be useful,
#     but WITHOUT ANY WARRANTY; without even the implied
#     warranty of
#     MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE. See the
#     GNU General Public License for more details.
#
#     You should have received a copy of the GNU General
#     Public License
#     along with this program; if not, write to the Free
#     Software
#     Foundation, Inc., 51 Franklin Street, Fifth Floor,
#     Boston,
#     MA 02110-1301, USA.

import random,math,os,time
```

```

from random import lognormvariate, normalvariate, random,
uniform
from threading import Thread, Lock
from scipy import stats
import multiprocessing

def rodarICL(k,i,contN):
    #
    # USA O ICL PARA ESTIMAR OS PARAMETRO A,B,C
    #
    # k: numero de testes
    # i: numero de itens de cada teste
    for contK in range(k):
        #aux1 = str(contK)
        # Texto do arquivo TCL
        texto = []
        texto.append("# Estima parâmetros de item usando
dados em
aux/dadosObservados/simulacao_teste_"+str(contK)+".dat")
        texto.append("# Lab. de Inovação em Tecnologias
Educaionais - UFLA (2011)")
        texto.append("# Arquivo de saída
aux/logs/simulacao_"+str(contN)+"_teste_" +
str(contK)+".log")
        texto.append("output -log_file
aux/logs/simulacao_"+str(contN)+"_teste_" +
str(contK)+".log")
        texto.append("# "+ str(i)+" itens dicotômicos")
        texto.append("allocate_items_dist "+ str(i) + " -
num_latent_dist_points 200")
        texto.append("# Ler tabela de respostas do arquivo
aux/dadosObservados/simulacao_"+str(contN)+"_teste_" +
str(contK)+".dat")
        texto.append("# Cada linha contem a resposta aos")
        texto.append("# "+ str(i)+" itens para cada aluno
nas colunas 1-"+ str(i)+".")
        texto.append("read_examinees
aux/dadosObservados/simulacao_"+str(contN)+"_teste_"+str(co
ntK)+".dat "+ str(i)+"il")
        texto.append("# Computar valores iniciais para
estimativa dos parâmetros do item")
        texto.append("starting_values_dichotomous")
        texto.append("# Perform EM iterations for computing
item parameter estimates.")
        texto.append("# Maximum of 50 EM iterations.")
        texto.append("EM_steps -max_iter 1000000 -crit
0.00000001")

```

```

        texto.append("# Print item parameter estimates and
discrete latent")
        texto.append("# variable distribution.")
        texto.append("print -item_param -latent_dist")
        texto.append("# Release memory allocated to hold
items and the")
        texto.append("# latent variable distribution")
        texto.append("release_items_dist")

        # Criar arquivo TCL
        file1 = open("simulacao_"+str(contN)+".tcl",'w')
        # escrever no arquivo file1
        for linha in texto:
            file1.write(linha)
            file1.write("\n")
        file1.close()

        # Rodar arquivo TCL
        os.system("./iclsh simulacao_"+str(contN)+".tcl >
lixo")
        #os.system("./iclsh simulacao_"+str(contN)+".tcl ")

        os.system("rm simulacao_"+str(contN)+".tcl")

def rodarTheta(k, i, contN):
    #
    # USA O ICL PARA ESTIMAR O THETA
    #
    # k: numero de testes
    # i: numero de itens do teste
    for contK in range(k):
        #aux1 = str(contK)
        # Texto do arquivo TCL
        texto = []
        texto.append("output -log_file
aux/logs/simulacao_"+str(contN)+"_saidaTheta.log")
        texto.append("allocate_items_dist "+ str(i))
        texto.append("read_examinees
aux/dadosObservados/simulacao_"+str(contN)+"_teste_"+str(co
ntK)+".dat "+ str(i)+"il")
        texto.append("read_item_param
aux/logs/simulacao_"+str(contN)+"_theta_"+str(contK)+".par
")
        #texto.append("read_item_param
aux/logs/simulacao_"+str(contN)+"_theta_"+str(contK)+".par
"+ str(i)+"il "+ str(i))

```

```

        #texto.append("read_item_param
aux/logs/simulacao_"+str(contN)+"_theta_"+str(contK)+".par
"+ str(i))
        texto.append("set estep [new_estep]")
        texto.append("estep_compute $estep 1 1")
        texto.append("delete_estep $estep")
        texto.append("set eapfile [open
aux/logs/simulacao_"+str(contN)+"_theta_" +
str(contK)+".log w]")
        texto.append("for {set i 1} {$i <= [num_examinees]}
{incr i} {")
        texto.append("  set resp [examinee_responses $i]")
        texto.append("  set numcorrect 0")
        texto.append("  foreach r $resp {")
        texto.append("    if {$r > 0} then {incr
numcorrect}")
        texto.append("  }")
        texto.append("  set eap [examinee_posterior_mean
$i]")
        texto.append("  set mle [examinee_theta_MLE $i -6.0
6.0]")
        texto.append("  puts $eapfile [format
\\%.6f\\t%.6f\\t%d\\ $eap $mle $numcorrect]")
        texto.append("}")
        texto.append("close $eapfile")
        texto.append("release_items_dist")

        # Criar arquivo TCL
        file1 =
open("theta_simulacao_"+str(contN)+".tcl",'w')
        # escrever no arquivo file1
        for linha in texto:
            file1.write(linha)
            file1.write("\n")
        file1.close()

        # Rodar arquivo TCL
        os.system("./iclsh
theta_simulacao_"+str(contN)+".tcl")
        os.system("rm theta_simulacao_"+str(contN)+".tcl")

def main():
    #criar pastas para arquivar saidas
    nome = str(raw_input("Digite o nome da simulacao: "))

```

```

while os.path.exists(nome):
    print("Ja existe um pasta com esse nome
"+"["+red("!!")+"]")
    nome = str(raw_input("Digite um novo nome: "))

os.system("rm -rf aux")
os.system("mkdir aux")
os.system("mkdir aux/dadosObservados")
os.system("mkdir aux/erro")
os.system("mkdir aux/erro2")
os.system("mkdir aux/logs")
os.system("mkdir aux/histogramas")
os.system("mkdir aux/saida")

# i: numero de itens dos testes
i = int(raw_input("Numero de itens: "))

# j: número de alunos
j = int(raw_input("Numero de alunos: "))
#j = 40

# k: número de testes
k = 1
print("Número de testes: " + str(k))

# l: numero de alternativas de cada item
#l = int(raw_input("Numero de alternativas: "))
l=5

# n: numero de repeticoes
n = int(raw_input("Digite o numero de repeticoes: "))

# definir parâmetros da distribuições
# Alpha
muA = float(raw_input("MuA: "))
sigmaA = float(raw_input("SigmaA: "))
#muA = 0
#sigmaA = 0.5
print("SigmaA: " + str(sigmaA))

# Beta
muB = float(raw_input("MuB: "))
sigmaB = float(raw_input("SigmaB: "))
#muB = 1.5
#sigmaB = 0.5

# Gama

```



```

muC = float (1) / 1
sigmaC = 0.05

# muT: mu para calculo do Theta
muT = float(raw_input("Media da distribuicao(theta):
"))
#muT = 0
print("MuT: " + str(muT))

# sigmaT: sigma para calculo do Theta
sigmaT = float(raw_input("Variancia da
distribuicao(theta): "))
#sigmaT = 1
print("SigmaT: " + str(sigmaT))

# numErros: numero de iteracoes com erro que serao
descartadas no fim
#print("Num Erros: " + str(numErros))
global numErros
numErros = 0
#semaforo para evitar condicao de corrida sobre a
variavel numErros
semaphore = Lock()
# numThreads: numero de threads que serao executadas em
paralelo
# para melhor aproveitamento dos nucleos do processador
# obs: sugere-se que seja igual ao num de nucleos do
processador
numThreads = multiprocessing.cpu_count() + 2
numThreads = 1
print("Num Threads: " + str(numThreads))
#marcar tempo inicial da simulacao
#para analise de eficiencia do codigo
tempo1 = time.clock()
tempo2 = time.time()

# Gerar parametro alfa, beta, gama e theta
alpha = []
beta = []
gama = []
#gerar(sortear) parametros a,b,c para o teste
for contK in range(k):
    alpha.append([])
    beta.append([])
    gama.append([])
    #gerar parametros para cada item i do teste k

```

```

# Retorno à distribuição normal para beta
# lognormal para alfa
for contI in range(i):
    alpha[contK].append(lognormvariate(muA,sigmaA))
    beta[contK].append(normalvariate(muB,sigmaB))
    gama[contK].append(normalvariate(muC,sigmaC))
#gerar(sortear) parametro theta
global theta
resortearAlunos = 0
if not(resortearAlunos):
    theta = []
    # gerar parametros para cada aluno j
    for contJ in range(j):
        # theta[j]: é a habilidade do indivíduo j
        #             definida pela normal padronizada de
muT e sigmaT
        theta.append(normalvariate(muT,sigmaT))
    #- Gravar dados sorteados nos arquivos de saida -
    arquivoTheta = open("aux/saida/theta.csv",'w')
    for contJ in range(j):
        arquivoTheta.write(str(theta[contJ]) + "\n")
    arquivoTheta.close

#Gravar dados sorteados
arquivoAlpha = open("aux/saida/alpha.csv",'w')
arquivoBeta = open("aux/saida/beta.csv",'w')
arquivoGama = open("aux/saida/gama.csv",'w')
for contI in range(i):
    auxAlpha = ""
    auxBeta = ""
    auxGama = ""
    for contK in range(k):
        auxAlpha += str(alpha[contK][contI])+ ";"
        auxBeta += str(beta[contK][contI])+ ";"
        auxGama += str(gama[contK][contI])+ ";"
    auxAlpha += "\n"
    auxBeta += "\n"
    auxGama += "\n"
    arquivoAlpha.write(auxAlpha)
    arquivoBeta.write(auxBeta)
    arquivoGama.write(auxGama)
arquivoAlpha.close()
arquivoBeta.close()
arquivoGama.close()

```

```
#####
###
# Funcao para repetir simulacao com os indices
especificados #

#####
###
def rodarIntervalo(intervalo):
    for contN in intervalo:
        print("Simulacao: "+str(contN))
        global theta

        #se for resortejar o grupo de alunos a cada
iteracao gera-se um novo theta aqui
        if resortejarAlunos:
            theta = []
            # gerar parametros para cada aluno j
            for contJ in range(j):
                # theta[j]: é a habilidade do indivíduo
j
                #                definida pela normal
padronizada de  $\mu_T$  e  $\sigma_T$ 
                theta.append(normalvariate( $\mu_T$ , $\sigma_T$ ))

            #
            # Gerar as respostas dos alunos(dados
observados)
            #
            D = 1.702
            for contK in range(k):
                Y = [[]]

            # gerar a tabela Y que representa as
respostas dos alunos
            #
            #   coluna = questao
            #   linha = aluno

            # rodar para cada aluno j

            for contJ in range(j):
                Y.append([])
                # gerar as respostas do aluno j para
cada item i do teste k
                for contI in range(i):
```

```

        p = gama[contK][contI] + (
(float(1)-gama[contK][contI]) / (1 + math.e**(-D *
alpha[contK][contI] * (theta[contJ] - beta[contK][contI]) )
))
        aux = random()
        Y[contJ].append(str(int(aux<p)))
        #print(str(Y[contJ][contI])+" | p =
"+str(p)+" | r = "+str(aux))

        # Escrever respostas Y do teste k no
arquivo aux/dadosObservados/simulacao_teste_k.dat
        file1 =
open("aux/dadosObservados/simulacao_"+str(contN)+"_teste_"+
str(contK)+".dat", 'w')
        for contJ in range(j):
            for contI in range(i):
                file1.write(str(Y[contJ][contI]))
                file1.write("\n")
        file1.close()

        # Rodar Simulacao "./iclsh simulacao.tcl"
rodarICL(k,i,contN)

        # Extrair dados do arquivo .log
a = [[]]
b = [[]]
c = [[]]
erro = False
for contK in range(k):
    arquivoLog =
open("aux/logs/simulacao_"+str(contN)+"_teste_" +
str(contK)+".log")
    lines = arquivoLog.readlines()

        # Inserir dados nos vetores a,b,c que
armazenam os valores gerados pelo icl
        a.append([])
        b.append([])
        c.append([])

        arquivoPar =
open("aux/logs/simulacao_"+str(contN)+"_theta_"+str(contK)+
".par", 'w')
        try:
            # Deletar cabeçalho

```

```

        while lines[0] != "(a, b, c for 3PL,
2PL, 1PL; a, b1, b2, ... for GPCM, PCM)\n":
            del lines[0]
        del lines[0]

        #verificar se dados sao validos

        for l in lines[0:i]:
            aux = l.split()
            if (float(aux[1]) > 20) or
(float(aux[2])>20) or (float(aux[3]) > .5) or
(float(aux[3]) < 0):
                erro = True
                print "valor(es) invalido(s)
encontrado(s)" + str(contN)
                print aux
                break
        except:
            erro = True

        #
        # Salva os dados se nao houver erros
        # caso contraio salva valores iguais aos
sorteados
        #
        if not(erro):
            for l in lines[0:i]:
                aux = l.split()
                a[contK].append(float(aux[1]))
                b[contK].append(float(aux[2]))
                c[contK].append(float(aux[3]))
                arquivoPar.write(l)
        else:
            for contI in range(i):

a[contK].append(alpha[contK][contI])
                b[contK].append(beta[contK][contI])
                c[contK].append(gama[contK][contI])
                arquivoPar.write(str(contI+1)+
"+str(alpha[contK][contI])+ " "+ str(beta[contK][contI]) +
" "+ str(gama[contK][contI])+"\n")
                arquivoPar.close

        #chama funcao para gerar theta da iteracao
rodarTheta(k,i,contN)

```

```

#extrair o parametro theta
t1 =[]
t2 =[]
for contK in range(k):
    try:
        arquivoLog =
open("aux/logs/simulacao_"+str(contN)+"_theta_" +
str(contK)+".log")
        lines = arquivoLog.readlines()
    except:
        erro = True
        # Inserir dados no vetor t que armazenam os
valores de theta gerados pelo icl
        t1.append([])
        t2.append([])
        if not(erro):
            for l in lines:
                aux = l.split()
                t1[contK].append(float(aux[0]))
                t2[contK].append(float(aux[1]))
        else:
            for contJ in range(j):
                t1[contK].append(theta[contJ])
                t2[contK].append(theta[contJ])

# verifica se houve erro
# nesse caso incrementa o numErros
if erro:
    print("Ocorreu um problema na iteração "+
str(contN))
    # inserir semaforo para gerar exclusao
mutua
    # evitando que 2 threads acessem a variavel
numErros ao mesmo tempo
    global numErros
    semaphore.acquire() # fecha o semaforo
    #Regiao Critica
    numErros += 1
    semaphore.release() # abre o semaforo

#
# gravar dados nos arquivos de saida
#
arquivoT1 =
open("aux/saida/t1_"+str(contN)+".csv", 'w')
arquivoT2 =
open("aux/saida/t2_"+str(contN)+".csv", 'w')

```

```

for contJ in range(j):
    auxT1 = ""
    auxT2 = ""
    for contK in range(k):
        if not(erro):
            auxT1 = auxT1 +
str(t1[contK][contJ])+ ";"
            auxT2 = auxT2 +
str(t2[contK][contJ]) + ";"
        else:
            auxT1 = auxT1 + str(theta[contJ])+
";"
            auxT2 = auxT2 + str(theta[contJ]) +
";"

            auxT1 += "\n"
            auxT2 += "\n"
            arquivoT1.write(auxT1)
            arquivoT2.write(auxT2)
    arquivoT1.close()
    arquivoT2.close()

arquivoA =
open("aux/saida/a_"+str(contN)+".csv", 'w')
arquivoB =
open("aux/saida/b_"+str(contN)+".csv", 'w')
arquivoC =
open("aux/saida/c_"+str(contN)+".csv", 'w')
for contI in range(i):
    auxA = ""
    auxB = ""
    auxC = ""
    for contK in range(k):
        if not(erro):
            auxA += str(a[contK][contI])+ ";"
            auxB += str(b[contK][contI])+ ";"
            auxC += str(c[contK][contI])+ ";"
        else:
            auxA += str(alpha[contK][contI])+
";"
            auxB += str(beta[contK][contI])+
";"
            auxC += str(gama[contK][contI])+
";"

            auxA += "\n"
            auxB += "\n"
            auxC += "\n"

```

```

        arquivoA.write(auxA)
        arquivoB.write(auxB)
        arquivoC.write(auxC)
    arquivoA.close()
    arquivoB.close()
    arquivoC.close()

#
# Classe que permite que multiplos processos rodem
simultaneamente
# isso permite um aproveitamento melhor do diversos
nucleos da CPU
#
class testit(Thread):
    def __init__(self, intervalo):
        Thread.__init__(self)
        self.intervalo = intervalo
        self.status = -1

    def run(self):
        rodarIntervalo(self.intervalo)

#defini os intervalos de indices que cada processo ira
rodar
testesPorThread = n/numThreads
restantes = n-(numThreads*testesPorThread)
intervalo = [-1]
threadList = []
for contNumTheads in range(numThreads):
    ultimoAnterior = intervalo[-1]
    ultimoAnterior+=1
    if contNumTheads < restantes:
        intervalo =
range(ultimoAnterior,ultimoAnterior+testesPorThread+1)
    else:
        intervalo =
range(ultimoAnterior,ultimoAnterior+testesPorThread)
        current = testit(intervalo)
        threadList.append(current)

# inicia a execucao do processo
current.start()

#espera a execucao de todas as threads para continuar a
execucao do codigo
for t in threadList:

```



```

t.join()

#
# Apos rodar todas as iteracoes inicia-se o calculo das
saidas (vies, EQM, etc...)
#
print(red("-Calcular Saidas-"))

somatorioA = []
somatorioB = []
somatorioC = []
mediaA = []
mediaB = []
mediaC = []
somatorioQuadA = []
somatorioQuadB = []
somatorioQuadC = []
desvioPadraoA = []
desvioPadraoB = []
desvioPadraoC = []
for contK in range(k):
    somatorioA.append([])
    somatorioB.append([])
    somatorioC.append([])
    mediaA.append([])
    mediaB.append([])
    mediaC.append([])
    somatorioQuadA.append([])
    somatorioQuadB.append([])
    somatorioQuadC.append([])
    desvioPadraoA.append([])
    desvioPadraoB.append([])
    desvioPadraoC.append([])
    for contI in range(i):
        somatorioA[contK].append(float(0))
        somatorioB[contK].append(float(0))
        somatorioC[contK].append(float(0))
        mediaA[contK].append(float(0))
        mediaB[contK].append(float(0))
        mediaC[contK].append(float(0))
        somatorioQuadA[contK].append(float(0))
        somatorioQuadB[contK].append(float(0))
        somatorioQuadC[contK].append(float(0))
        desvioPadraoA[contK].append(float(0))
        desvioPadraoB[contK].append(float(0))
        desvioPadraoC[contK].append(float(0))

```

```

    arquivoAlpha =
open("aux/saida/alpha.csv", 'r').readlines()
    arquivoBeta =
open("aux/saida/beta.csv", 'r').readlines()
    arquivoGama =
open("aux/saida/gama.csv", 'r').readlines()
    sA = 0
    sB = 0
    sC = 0

    for contN in range(n):
        arquivoA =
open("aux/saida/a_"+str(contN)+".csv", 'r').readlines()
        arquivoB =
open("aux/saida/b_"+str(contN)+".csv", 'r').readlines()
        arquivoC =
open("aux/saida/c_"+str(contN)+".csv", 'r').readlines()
            for contI in range(i):
                for contK in range(k):
                    somatorioA[contK][contI] +=
float(arquivoA[contI].split(";")[contK])
                    somatorioB[contK][contI] +=
float(arquivoB[contI].split(";")[contK])
                    somatorioC[contK][contI] +=
float(arquivoC[contI].split(";")[contK])
                    sA += (alpha[contK][contI] -
float(arquivoA[contI].split(";")[contK]))**2
                    sB += (beta[contK][contI] -
float(arquivoB[contI].split(";")[contK]))**2
                    sC += (gama[contK][contI] -
float(arquivoC[contI].split(";")[contK]))**2

    #N = numero de iteracoes OK
    N = n-numErros

    if N == 0:
        print "Nenhuma iteracao com sucesso"
        return -1

    sA = sA / (N*i)
    sB = sB / (N*i)
    sC = sC / (N*i)

```

```

    for contI in range(i):
        for contK in range(k):
            #global numErros
            mediaA[contK][contI] =
somatorioA[contK][contI]/n
            mediaB[contK][contI] =
float(somatorioB[contK][contI])/n
            mediaC[contK][contI] =
somatorioC[contK][contI]/n

    for contN in range(n):
        arquivoA =
open("aux/saida/a_"+str(contN)+".csv", 'r').readlines()
        arquivoB =
open("aux/saida/b_"+str(contN)+".csv", 'r').readlines()
        arquivoC =
open("aux/saida/c_"+str(contN)+".csv", 'r').readlines()
        for contI in range(i):
            for contK in range(k):
                somatorioQuadA[contK][contI] +=
(float(arquivoA[contI].split(";")[contK]) -
mediaA[contK][contI] )**2
                somatorioQuadB[contK][contI] +=
(float(arquivoB[contI].split(";")[contK]) -
mediaB[contK][contI] )**2
                somatorioQuadC[contK][contI] +=
(float(arquivoC[contI].split(";")[contK]) -
mediaC[contK][contI] )**2

        for contI in range(i):
            for contK in range(k):

                desvioPadraoA[contK][contI]
= ((float(somatorioQuadA[contK][contI])/(N-1)))**.5
                desvioPadraoB[contK][contI]
= ((float(somatorioQuadB[contK][contI])/(N-1)))**.5
                desvioPadraoC[contK][contI]
= ((float(somatorioQuadC[contK][contI])/(N-1)))**.5

        #
        # Gravar saida nos arquivos Ra, Rb, Rc
        # contendo os dados de cada item do teste
        # São eles
        # indice do Item, dadoEsperado, dadoObservado[n],
media, desvioPadrao, vies

```

```

#
for contK in range(k):
    arquivo = open("aux/Ra_"+str(contK)+".csv",'w')

    #cabeçalho
    linha = "Item;"
    for contN in range(n):
        linha+=("a"+str(contN)+";")
    linha+=
"alpha;mediaA;desvioPadraoA;viesA;erroQuadraticoMedio\n"
    arquivo.write(linha)
    # itens do teste
    for contI in range(i):
        # Indice
        linha = str(contI) + ";"

        # Dados Observados
        for contN in range(n):
            arquivoA =
open("aux/saida/a_"+str(contN)+".csv",'r').readlines()
            linha +=
str(arquivoA[contI].split(";")[contK]).replace('.',',')+";"

            # Dado Esperado
            linha +=
str(round(alpha[contK][contI],6)).replace('.',',')+ ";"

            # Media
            linha +=
str(round(mediaA[contK][contI],6)).replace('.',',')+ ";"

            # Desvio Padrao
            linha +=
str(round(desvioPadraoA[contK][contI],6)).replace('.',',')+
";"

            #ViessomatorioQuadA
            linha += str(round(alpha[contK][contI]-
mediaA[contK][contI],6)).replace('.',',')+ ";"

            # Desvio Padrao
            linha +=
str(round(desvioPadraoA[contK][contI]**2+(alpha[contK][cont
I]-mediaA[contK][contI])**2,6)).replace('.',',')+ ";"

        linha += "\n"
        arquivo.write(linha)

```

```

arquivo.close()

arquivo = open("aux/Rb_"+str(contK)+".csv",'w')

#cabeçalho
linha = "Item;"
for contN in range(n):
    linha+=("b"+str(contN)+";")
linha+=
"beta;mediaB;desvioPadraoB;viesB;erroQuadraticoMedio\n"
arquivo.write(linha)
# itens do teste
for contI in range(i):
    linha = str(contI) + ";"
    for contN in range(n):
        arquivoB =
open("aux/saida/b_"+str(contN)+".csv",'r').readlines()
        linha +=
str(arquivoB[contI].split(";")[contK]).replace('.',',')+";"

        linha +=
str(round(beta[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(mediaB[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoB[contK][contI],6)).replace('.',',')+
";"
        linha += str(round(beta[contK][contI]-
mediaB[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoB[contK][contI]**2+(beta[contK][contI
]-mediaB[contK][contI])**2,6)).replace('.',',')+ ";"
        linha += "\n"
        arquivo.write(linha)
arquivo.close()

arquivo = open("aux/Rc_"+str(contK)+".csv",'w')
#cabeçalhol
linha = "Item;"
for contN in range(n):
    linha+=("c"+str(contN)+";")
linha+=
"gama;mediaC;desvioPadraoC;viesC;erroQuadraticoMedio\n"
arquivo.write(linha)
# itens do teste
for contI in range(i):

```

```

        linha = str(contI) + ";"
        for contN in range(n):
            arquivoA =
open("aux/saida/c_"+str(contN)+".csv", 'r').readlines()
            linha +=
str(arquivoC[contI].split(";")[contK]).replace('.',',')+";"

            linha +=
str(round(gama[contK][contI],6)).replace('.',',')+ ";"
            linha +=
str(round(mediaC[contK][contI],6)).replace('.',',')+ ";"
            linha +=
str(round(desvioPadraoC[contK][contI],6)).replace('.',',')+
";"
            linha += str(round(gama[contK][contI]-
mediaC[contK][contI],6)).replace('.',',')+ ";"
            linha +=
str(round(desvioPadraoC[contK][contI]**2+(gama[contK][contI
]-mediaC[contK][contI])**2,6)).replace('.',',')+ ";"
            linha += "\n"
            arquivo.write(linha)
        arquivo.close()

#####
#Saida Resumida
#####

arquivo =
open("aux/SaidaResumida_"+str(contK)+".csv", 'w')

#cabecalho
linha = "Item;"
linha+=
"alpha;mediaA;desvioPadraoA;viesA;erroQuadraticoMedio;beta;
mediaB;desvioPadraoB;viesB;erroQuadraticoMedio;gama;mediaC;
desvioPadraoC;viesC;erroQuadraticoMedio\n"
arquivo.write(linha)
# itens do teste
for contI in range(i):
    # Indice
    linha = str(contI) + ";"
    # Dado Esperado
    linha +=
str(round(alpha[contK][contI],6)).replace('.',',')+ ";"
    # Media
    linha +=
str(round(mediaA[contK][contI],6)).replace('.',',')+ ";"

```

```

        # Desvio Padrao
        linha +=
str(round(desvioPadraoA[contK][contI],6)).replace('.',',')+
";"
        #Vies
        linha += str(round(alpha[contK][contI]-
mediaA[contK][contI],6)).replace('.',',')+ ";"
        # Erro Quadratico Medio
        linha +=
str(round(desvioPadraoA[contK][contI]**2+(alpha[contK][contI]-
mediaA[contK][contI])**2,6)).replace('.',',')+ ";"

        linha +=
str(round(beta[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(mediaB[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoB[contK][contI],6)).replace('.',',')+
";"
        linha += str(round(beta[contK][contI]-
mediaB[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoB[contK][contI]**2+(beta[contK][contI]-
mediaB[contK][contI])**2,6)).replace('.',',')+ ";"

        linha +=
str(round(gama[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(mediaC[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoC[contK][contI],6)).replace('.',',')+
";"
        linha += str(round(gama[contK][contI]-
mediaC[contK][contI],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoC[contK][contI]**2+(gama[contK][contI]-
mediaC[contK][contI])**2,6)).replace('.',',')+ ";"

        linha += "\n"
        arquivo.write(linha)
        arquivo.close()

#
# Saidas THETA
#

```

```

somatorioT = []
mediaT = []
somatorioQuadT = []
desvioPadraoT = []
for contK in range(k):
    somatorioT.append([])
    mediaT.append([])
    somatorioQuadT.append([])
    desvioPadraoT.append([])
    for contJ in range(j):
        somatorioT[contK].append(float(0))
        mediaT[contK].append(float(0))
        somatorioQuadT[contK].append(float(0))
        desvioPadraoT[contK].append(float(0))

sT1 = 0
for contN in range(n):
    arquivoT =
open("aux/saida/tl_"+str(contN)+".csv",'r').readlines()
    for contJ in range(j):
        for contK in range(k):
            somatorioT[contK][contJ] +=
float(arquivoT[contJ].split(";")[contK])
            sT1 += (theta[contJ] -
float(arquivoT[contJ].split(";")[contK]))**2
        sT1 = sT1 / (N* j)

    for contJ in range(j):
        for contK in range(k):
            mediaT[contK][contJ] =
somatorioT[contK][contJ]/n

    for contN in range(n):
        arquivoT =
open("aux/saida/tl_"+str(contN)+".csv",'r').readlines()
        for contJ in range(j):
            for contK in range(k):
                somatorioQuadT[contK][contJ] +=
(float(arquivoT[contJ].split(";")[contK]) -
mediaT[contK][contJ] )**2

        for contJ in range(j):
            for contK in range(k):
                desvioPadraoT[contK][contJ]
= ((float(somatorioQuadT[contK][contJ])/(N-1)))**.5

```



```

#
# Gravar saida nos arquivos Rt1
# contendo os dados de cada item do teste
# São eles
#   indice do Item, dadoEsperado, dadoObservado[n],
media, desvioPadrao, vies
#
for contK in range(k):
    arquivo = open("aux/Rt1_"+str(contK)+".csv", 'w')

    #cabeçalho
    linha = "Item;"
    for contN in range(n):
        linha+="t"+str(contN)+";"
    linha+=
"theta;mediaT;desvioPadraoT;viesT;erroQuadraticoMedio\n"
    arquivo.write(linha)
    # itens do teste
    for contJ in range(j):
        # Indice
        linha = str(contJ) + ";"

        # Dados Observados
        for contN in range(n):
            arquivoT =
open("aux/saida/t1_"+str(contN)+".csv", 'r').readlines()
            linha +=
str(arquivoT[contJ].split(";")[contK]).replace('.',',')+";"

        # Dado Esperado
        linha +=
str(round(theta[contJ],6)).replace('.',',')+ ";"

        # Media
        linha +=
str(round(mediaT[contK][contJ],6)).replace('.',',')+ ";"

        # Desvio Padrao
        linha +=
str(round(desvioPadraoT[contK][contJ],6)).replace('.',',')+
";"

        #Vies
        linha += str(round(theta[contJ]-
mediaT[contK][contJ],6)).replace('.',',')+ ";"

        #erroQuadraticoMedio

```

```

        linha +=
str(round(desvioPadraoT[contK][contJ]**2+(theta[contJ]-
mediaT[contK][contJ])**2,6)).replace('.',',')+ ";"
        linha += "\n"
        arquivo.write(linha)
    arquivo.close()

somatorioT = []
mediaT = []
somatorioQuadT = []
desvioPadraoT = []
for contK in range(k):
    somatorioT.append([])
    mediaT.append([])
    somatorioQuadT.append([])
    desvioPadraoT.append([])
    for contJ in range(j):
        somatorioT[contK].append(float(0))
        mediaT[contK].append(float(0))
        somatorioQuadT[contK].append(float(0))
        desvioPadraoT[contK].append(float(0))

sT2 = 0
for contN in range(n):
    arquivoT =
open("aux/saida/t2_"+str(contN)+".csv",'r').readlines()
    for contJ in range(j):
        for contK in range(k):
            somatorioT[contK][contJ] +=
float(arquivoT[contJ].split(";")[contK])
            sT2 += (theta[contJ] -
float(arquivoT[contJ].split(";")[contK]))**2
sT2 = sT2 / (N*j)

for contJ in range(j):
    for contK in range(k):
        #global numErros
        mediaT[contK][contJ] =
somatorioT[contK][contJ]/n

for contN in range(n):
    arquivoT =
open("aux/saida/t2_"+str(contN)+".csv",'r').readlines()
    for contJ in range(j):

```

```

        for contK in range(k):
            somatorioQuadT[contK][contJ] +=
(float(arquivoT[contJ].split(";")[contK]) -
mediaT[contK][contJ] )**2

    for contJ in range(j):
        for contK in range(k):
            desvioPadraoT[contK][contJ]
= ((float(somatorioQuadT[contK][contJ])/(N-1))**.5

#
# Gravar saída nos arquivos Rt2
# contendo os dados de cada item do teste
# São eles
# índice do Item, dadoEsperado, dadoObservado[n],
media, desvioPadrao, vies
#
for contK in range(k):
    arquivo = open("aux/Rt2_"+str(contK)+".csv", 'w')

#cabeçalho
linha = "Item;"
for contN in range(n):
    linha+=("t"+str(contN)+";")
linha+=
"theta;mediaT;desvioPadraoT;viesT;erroQuadraticoMedio\n"
arquivo.write(linha)
# itens do teste
for contJ in range(j):
    # Índice
    linha = str(contJ) + ";"

    # Dados Observados
    for contN in range(n):
        arquivoT =
open("aux/saida/t2_"+str(contN)+".csv", 'r').readlines()
        linha +=
str(arquivoT[contJ].split(";")[contK]).replace('.',',')+ ";"

    # Dado Esperado
    linha +=
str(round(theta[contJ],6)).replace('.',',')+ ";"

    # Media
    linha +=
str(round(mediaT[contK][contJ],6)).replace('.',',')+ ";"

```

```

        # Desvio Padrao
        linha +=
str(round(desvioPadraoT[contK][contJ],6)).replace('.',',')+
";"

        #Vies
        linha += str(round(theta[contJ]-
mediaT[contK][contJ],6)).replace('.',',')+ ";"
        linha +=
str(round(desvioPadraoT[contK][contJ]**2+(theta[contJ]-
mediaT[contK][contJ])**2,6)).replace('.',',')+ ";"
        linha += "\n"
        arquivo.write(linha)
    arquivo.close()
#
#Calcular Correlacao
#
# para o calculo da correlacao utilizou-se a funcao
pearsonr da biblioteca stats do pacote scipy do python
#
# no Ubuntu para instalar o pacote basta utilizar o
comando:
# "sudo apt-get install python-scipy"
#

vetA = []
vetAlpha = []
vetB = []
vetBeta = []
vetC = []
vetGama = []
for contN in range(n):
    arquivoA =
open("aux/saida/a_"+str(contN)+".csv",'r').readlines()
    arquivoB =
open("aux/saida/b_"+str(contN)+".csv",'r').readlines()
    arquivoC =
open("aux/saida/c_"+str(contN)+".csv",'r').readlines()
    for contI in range(i):
        for contK in range(k):
            vetA.append(
float(arquivoA[contI].split(";")[contK]))
            vetAlpha.append(alpha[contK][contI])
            vetB.append(
float(arquivoB[contI].split(";")[contK]))
            vetBeta.append(beta[contK][contI])

```

```

        vetC.append(
float(arquivoC[contI].split(";")[contK]))
        vetGama.append(gama[contK][contI])
        correlacaoA = stats.pearsonr(vetA,vetAlpha)
        correlacaoB = stats.pearsonr(vetB,vetBeta)
        correlacaoC = stats.pearsonr(vetC,vetGama)

vetT1= []
vetT2 = []
vetTheta = []
for contN in range(n):
    arquivoT1 =
open("aux/saida/t1_"+str(contN)+".csv",'r').readlines()
    arquivoT2 =
open("aux/saida/t2_"+str(contN)+".csv",'r').readlines()
    for contJ in range(j):
        for contK in range(k):
            vetT1.append(
float(arquivoT1[contJ].split(";")[contK]))
            vetT2.append(
float(arquivoT2[contJ].split(";")[contK]))
            vetTheta.append(theta[contJ])
            correlacaoT1 = stats.pearsonr(vetT1,vetTheta)
            correlacaoT2 = stats.pearsonr(vetT2,vetTheta)

#Gravar Saida Resumidissima

arquivo =
open("aux/resumo_"+str(i)+"itens_"+str(j)+"alunos.csv",'w')
linhas = []
linhas.append("itens;alunos;Correlacao A;Correlacao
B;Correlacao C;Correlacao T1;Correlacao T2;Erro Quadratico
Medio A;Erro Quadratico Medio B;Erro Quadratico Medio
C;Erro Quadratico Medio T1;Erro Quadratico Medio T2;\n")

linhas.append(str(i)+";")
linhas.append(str(j)+";")

linhas.append(str(correlacaoA[0]).replace(".",",")+";")
linhas.append(str(correlacaoB[0]).replace(".",",")+";")
linhas.append(str(correlacaoC[0]).replace(".",",")+";")

linhas.append(str(correlacaoT1[0]).replace(".",",")+";")

linhas.append(str(correlacaoT2[0]).replace(".",",")+";")

```

```

linhas.append(str(sA).replace(".",",")+";")
linhas.append(str(sB).replace(".",",")+";")
linhas.append(str(sC).replace(".",",")+";")
linhas.append(str(sT1).replace(".",",")+";")
linhas.append(str(sT2).replace(".",",")+";\n")

for linha in linhas:
    arquivo.write(linha)
arquivo.close()

#Gravar parametros da simulacao
arquivo = open("aux/parametros.txt", 'w')
linhas = []
linhas.append("Simulacao: "+str(nome)+"\n\n")
linhas.append("Numero de itens: "+str(i)+"\n")
linhas.append("Numero de alunos: "+str(j)+"\n")
linhas.append("Número de testes: "+str(k)+"\n")
linhas.append("Número de alternativas: "+str(l)+"\n")
linhas.append("-----\n")
linhas.append("A como log-normal (boa discriminação
sigmaA ~ 0.25 \n")
    linhas.append("MuA: "+str(muA)+"\n")
    linhas.append("SigmaA:
"+str(sigmaA)+"\n")
        linhas.append("B como normal (difícil <=> muB > 0 \n")
        linhas.append("MuB: "+str(muB)+"\n")
        linhas.append("SigmaB:
"+str(sigmaB)+"\n")
            linhas.append("MuC: "+str(muC)+"\n")
            linhas.append("SigmaC:
"+str(sigmaC)+"\n")
                linhas.append("MuT: "+str(muT)+"\n")
                linhas.append("SigmaT:
"+str(sigmaT)+"\n")
                    linhas.append("-----\n")
                    linhas.append("Iteracoes: "+str(n)+"\n")
                    linhas.append("Num Erros:
"+str(numErros)+"\n")
                        linhas.append("Iteracoes OK: "+str(n-
numErros)+"\n")
                            linhas.append("Num Threads:
"+str(numThreads)+"\n")
                                linhas.append("Correlacao A:
"+str(correlacaoA)+"\n")
                                    linhas.append("Correlacao B:
"+str(correlacaoB)+"\n")

```

```

        linhas.append("Correlacao C:
"+str(correlacaoC)+"\n")
        linhas.append("Correlacao T1:
"+str(correlacaoT1)+"\n")
        linhas.append("Correlacao T2:
"+str(correlacaoT2)+"\n")

    for linha in linhas:
        arquivo.write(linha)
    arquivo.close()

    # Renomear a pasta para o nome da simulacao
    os.system("mv -f aux/ "+nome+"/")
    os.system("rm lixo")

    tempo1 = time.clock() - tempo1
    tempo2 = time.time() - tempo2
    print("tempo processador : "+ '\033[92m' + str(tempo1)
+ '\033[0m')
    print("tempo : "+bold(green(str(tempo2))))

def green(s):
    return('\033[92m' + s + '\033[0m')
def bold(s):
    return('\033[1m' + s + '\033[0m')
def red(s):
    return('\033[91m' + s + '\033[0m')

if __name__ == "__main__":

    #
    # Le o numero de simulacoes a ser executado
    # Interessante quando voce quer rodar diversas
simulacoes
    # com parametros diferentes
    #
    numTestes = int(raw_input("Digite o numero de
simulacoes a ser executado:"))

    tempo2 = time.time()
    for conNumTestes in range(numTestes):
        main()
    tempo2 = time.time() - tempo2
    print("tempo total das "+str(numTestes)+" simulacoes:
"+bold(green(str(tempo2))))

```

