



**UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE CIÊNCIAS EXATAS
DEPARTAMENTO DE ESTATÍSTICA
CURSO DE ESTATÍSTICA**

**LUCAS BOZZA
TIAGO TEIXEIRA**

**TESTE ADAPTATIVO COMPUTADORIZADO UNIDIMENSIONAL COM USO DO
PACOTE MIRT CAT**

**CURITIBA
2015**

**LUCAS BOZZA
TIAGO TEIXEIRA**

**TESTE ADAPTATIVO COMPUTADORIZADO UNIDIMENSIONAL COM USO DO
PACOTE MIRT CAT**

Trabalho de Conclusão de Curso apresentado à disciplina CE078, Laboratório de Estatística, do Curso de Graduação em Estatística da Universidade Federal do Paraná, como requisito parcial para obtenção do título de Estatístico.

Orientador: Ms. Adilson dos Anjos

**CURITIBA
2015**

AGRADECIMENTOS

Ao professor Adilson dos Anjos, por confiar e nos orientar durante a execução deste trabalho.

À nossa banca, César Augusto Taconeli e Rhuan Gabriel C. de Lima, por aceitarem participar e nos orientar para a entrega final deste trabalho.

Aos nossos familiares, que sempre nos apoiaram durante todos os anos de nossa graduação.

Aos nossos colegas de curso que de alguma forma contribuíram para nossa formação e o desenvolvimento deste trabalho.

À todos o nosso muito obrigado!

RESUMO

Teste Adaptativo Computadorizado Unidimensional com Uso do Pacote MirtCAT

Um dos problemas decorrentes da teoria clássica dos testes (TCT) é que ela não considera o nível de dificuldade do teste para gerar os escores dos respondentes. Para corrigir tal problema, a teoria da resposta aos itens (TRI) surge como uma forma de aperfeiçoar a estimação dos escores dos respondentes, permitindo, por meio de seus modelos, uma análise tanto dos respondentes quanto dos itens dos testes. Os testes adaptativos computadorizados (TACs) foram desenvolvidos para tornar a aplicação dos testes uma forma mais dinâmica e individualizada em relação aos testes da teoria clássica e neste trabalho são descritas diversas possibilidades para aplicar um TAC. Também é descrito como utilizar o pacote *mirtCAT* para a criação de diferentes TACs utilizando para a exemplificação um banco de dados real, e um banco de dados simulado é utilizado para demonstrar o comportamento de diferentes métodos de estimação da habilidade do respondente e de critérios de seleção dos itens. É desenvolvido também um modelo de apresentação dos resultados de um TAC por meio do pacote *shiny* para facilitar a análise do teste pelo julgador.

Palavras-chave: Teste adaptativo computadorizado. TAC. Pacote *mirtCAT*. Pacote *shiny*. Teoria da resposta ao item. TRI.

LISTA DE FIGURAS

FIGURA 1 - FLUXOGRAMA GERAL DE UM TAC UNIDIMENSIONAL	3
FIGURA 2 - CURVA CARACTERÍSTICA DO ITEM	16
FIGURA 3 - EXEMPLO DE UMA PLANILHA COM ITENS, SEUS PARÂMETROS E OPÇÕES DE RESPOSTAS	36
FIGURA 4 - <i>FIRSTPAGE</i> DE UM TAC COM TAMANHO FIXO	41
FIGURA 5 - INFORMAÇÕES PESSOAIS DE UM TESTE	41
FIGURA 6 – ÚLTIMA PAGINA ANTES DO INÍCIO DE UM TESTE	42
FIGURA 7 – FORMA DE APRESENTAÇÃO DAS PERGUNTAS EM UM TESTE....	42
FIGURA 8 - <i>LASTPAGE</i> DE UM TAC COM TAMANHO FIXO.....	43
FIGURA 9 - <i>FIRSTPAGE</i> DE UM TAC COM CLASSIFICAÇÃO	45
FIGURA 10 - <i>LASTPAGE</i> DE UM TAC COM CLASSIFICAÇÃO COM O RESPONDENTE APROVADO	46
FIGURA 11 - <i>LASTPAGE</i> DE UM TAC COM CLASSIFICAÇÃO COM O RESPONDENTE REPROVADO	46
FIGURA 12 - <i>FIRSTPAGE</i> DE TESTES NÃO ADAPTATIVOS	49
FIGURA 13 - <i>LASTPAGE</i> DE TESTES NÃO ADAPTATIVOS	49
FIGURA 14 – APRESENTAÇÃO DOS RESULTADOS DO TESTE	54
FIGURA 15 – APRESENTAÇÃO DOS RESULTADOS DO TESTE COM UM RESPONDENTE CLASSIFICADO COMO <i>NO DECISION</i>	55
FIGURA 16 – APRESENTAÇÃO DOS RESULTADOS DO TESTE COM UM RESPONDENTE CLASSIFICADO COMO <i>BELOW CUTOFF</i>	55
FIGURA 17 – APRESENTAÇÃO DOS RESULTADOS DO TESTE COM UM RESPONDENTE CLASSIFICADO COMO <i>ABOVE CUTOFF</i>	56

LISTA DE QUADROS

QUADRO 1 – RESPOSTAS SIMULADAS DE UM RESPONDENTE COM <i>THETA</i> IGUAL A 1	19
QUADRO 2 – RESPOSTAS SIMULADAS DE RESPONDENTES COM DIFERENTES <i>THETAS</i>	20
QUADRO 3 – DESCRIÇÃO DAS FUNÇÕES EM UM TAC	32
QUADRO 4 – DADOS DE UM TESTE APÓS IMPORTÁ-LOS AO R.....	36
QUADRO 5 – OBJETO DO MODELO UTILIZADO PELA FUNÇÃO <i>MIRTCAT</i>	37
QUADRO 6 – PREPARAÇÃO DO TESTE	37
QUADRO 7 – TAC COM OS DADOS SOBRE FUTEBOL COM TAMANHO FIXO ..	38
QUADRO 8 – CÓDIGOS PARA GERAR A APLICAÇÃO EM <i>SHINY</i>	53

LISTA DE TABELAS

TABELA 1 - HABILIDADES ESTIMADAS E ITENS RESPONDIDOS EM TESTES SIMULADOS COM DIFERENTES ARGUMENTOS <i>CRITERIA</i> E <i>METHOD</i>	51
TABELA 2 - HABILIDADES ESTIMADAS E ITENS RESPONDIDOS EM TESTES SIMULADOS COM DIFERENTES ARGUMENTOS <i>THETA</i>	52

SUMÁRIO

1 INTRODUÇÃO	1
2 REVISÃO DE LITERATURA	3
2.1 TESTE ADAPTATIVO COMPUTADORIZADO	3
2.1.1 Banco de itens.....	3
2.1.2 Critério inicial do teste	5
2.1.3 Critério de seleção do item.....	5
2.1.4 Critério de exposição do item	6
2.1.4.1 Critério de exposição <i>Sympson-Hetter</i>	7
2.1.5 Estimativa da habilidade.....	9
2.1.6 Critério de parada do teste	10
2.1.7 Vantagens e desvantagens na aplicação de um TAC.....	11
3 MATERIAL E MÉTODOS	13
3.1 CONJUNTO DE DADOS.....	13
3.1.1 Conjunto de dados futebol.....	13
3.1.2 Conjunto de dados simulados	14
3.2 MODELO LOGÍSTICO UNIDIMENSIONAL DE 3 PARÂMETROS.....	14
3.3 RECURSOS COMPUTACIONAIS	16
3.3.1 Pacote <i>mirtCAT</i>	16
3.3.1.1 Função <i>createShinyGUI</i>	17
3.3.1.2 Função <i>findNextItem</i>	17
3.3.1.3 Função <i>generate.mirt_object</i>	18
3.3.1.3.1 Argumento <i>parameters</i>	18
3.3.1.3.2 Argumento <i>itemtype</i>	18
3.3.1.4 Função <i>generate_pattern</i>	19

3.3.1.5 Função <i>mirtCAT</i>	20
3.3.1.5.1 Argumento <i>df</i>	20
3.3.1.5.2 Argumento <i>mo</i>	21
3.3.1.5.3 Argumento <i>method</i>	21
3.3.1.5.4 Argumento <i>criteria</i>	22
3.3.1.5.5 Argumento <i>start_item</i>	22
3.3.1.5.6 Argumento <i>local_pattern</i>	22
3.3.1.5.7 Argumento <i>design_elements</i>	23
3.3.1.5.8 Argumento <i>design</i>	23
3.3.1.5.9 Argumento <i>shinyGUI</i>	27
3.3.1.5.10 Argumento <i>preCAT</i>	30
3.3.1.5.11 Retornos da função <i>mirtCAT</i>	30
3.3.1.6 Função <i>mirtCAT_preamble</i>	31
3.3.1.7 Função <i>updateDesign</i>	31
3.3.2 Pacote <i>shiny</i>	33
4 RESULTADOS E DISCUSSÃO.....	35
4.1 TAC COM OS DADOS SOBRE FUTEBOL COM TAMANHO FIXO	37
4.2 TAC COM CLASSIFICAÇÃO	43
4.3 TESTES NÃO ADAPTATIVOS.....	47
4.4 TAC COM BANCO DE ITENS SIMULADO	50
4.5 APRESENTAÇÃO DOS RESULTADOS	53
5 CONSIDERAÇÕES FINAIS	57
REFERÊNCIAS.....	58

1 INTRODUÇÃO

Baseada em escolhas, a teoria clássica do teste (TCT) utiliza escores brutos ou ponderados, que são a soma das respostas corretas, em suas avaliações para determinar a habilidade de cada respondente no teste aplicado (MOREIRA JUNIOR, 2011). Muitos tipos de exames, bem como vestibulares e concursos públicos, utilizam desses escores como critério de seleção para seus candidatos, de tal forma que quanto maior for o escore do respondente, mais qualificado este está e, conseqüentemente, será o respondente selecionado no processo.

Porém, um problema da TCT é que o escore gerado ao fim do teste não depende somente da habilidade do respondente, mas também do nível de dificuldade do teste, o que implica dizer que respondentes com habilidade no nível médio terão seu escore inflacionado em provas mais fáceis e deflacionado em provas mais difíceis, prejudicando tanto a classificação dos respondentes quanto a análise dos julgadores (MOREIRA JUNIOR, 2011).

Como forma de corrigir este problema da TCT, algumas teorias e métodos foram surgindo e a teoria da resposta ao item (TRI) vem ganhando espaço a cada dia que passa entre os concursos e avaliações educacionais (ANDRADE; TAVARES; VALLE, 2000), pois mantém o foco da análise nos itens e na habilidade do respondente, e ainda dispõe de modelos que permitem ao julgador avaliar populações em provas com nenhuma ou apenas algumas questões em comum e ainda assim mensurar diferenças entre os candidatos, medindo por uma escala, previamente determinada, o quanto aquele respondente possui de conhecimento sobre o traço latente a ser avaliado, ou seja, o assunto abordado no teste.

A TRI contempla um conjunto de modelos que podem conter apenas um parâmetro até modelos com vários parâmetros associados. O modelo da TRI que tem sido aplicado com maior frequência é o modelo logístico unidimensional de três parâmetros (a, b, c) (3PL): um parâmetro que mensura a discriminação do item (a), um parâmetro que descreve a dificuldade do item (b), e outro que representa a probabilidade de acerto casual (c) (ANDRADE; TAVARES; VALLE, 2000).

Utilizando modelos da TRI, os testes adaptativos computadorizados, ou simplesmente TACs, foram elaborados para tornar a aplicação de provas e questionários mais dinâmica e individualizada. O TAC, também, possibilita uma seleção de respondentes mais rápida e poderosa que os testes tradicionais por conta de suas técnicas apuradas, que possibilitam ao julgador observar o que cada respondente conhece sobre o teste aplicado sem, em determinados casos, precisar que o mesmo responda a todas as questões propostas no questionário.

Com base nisso, os objetivos do trabalho são, demonstrar a aplicação de testes adaptativos computadorizados no R (R DEVELOPMENT CORE TEAM, 2015), por meio do RStudio (RSTUDIO, 2015) e do pacote *mirtCAT* (CHALMERS, 2015a), em diferentes cenários de uma forma simples e de fácil entendimento ao usuário, também demonstrar o comportamento de diferentes critérios de seleção dos itens e métodos de estimativa da habilidade em um determinado banco de itens e construir uma aplicação utilizando o pacote *shiny* (CHANG et al., 2015) para a apresentação final dos resultados dos respondentes.

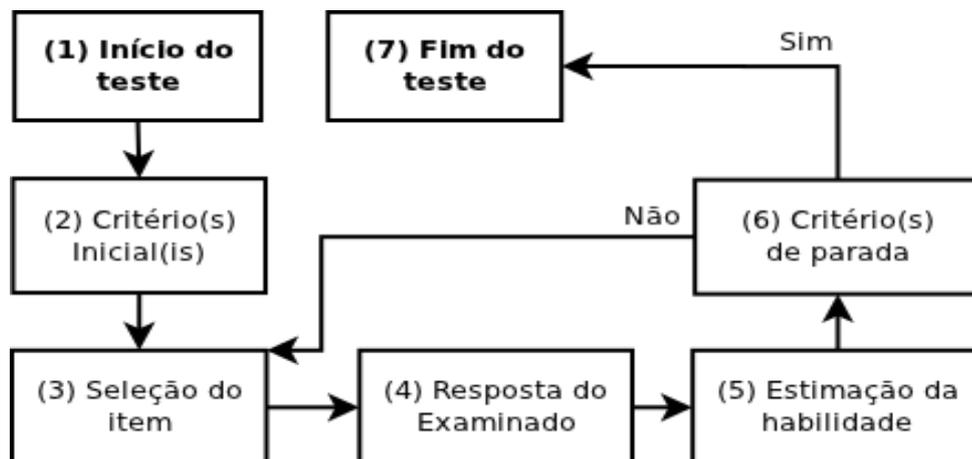
2 REVISÃO DE LITERATURA

2.1 TESTE ADAPTATIVO COMPUTADORIZADO

Um TAC é composto basicamente por quatro componentes: o banco de itens, o critério inicial do teste, o critério de seleção dos itens e o critério de parada do teste (GONÇALVES, 2013).

Na FIGURA 1, apresenta-se um fluxograma com os passos necessários para a realização de um TAC unidimensional, onde a habilidade é estimada em apenas um traço latente.

FIGURA 1 - FLUXOGRAMA GERAL DE UM TAC UNIDIMENSIONAL



Fonte: GONÇALVES, 2013.

2.1.1 Banco de itens

A TRI possui vários formatos de itens que podem ser utilizados em um TAC, como dicotômicos, politômicos, escala gradual, entre outros. A definição do tipo de item afetará diretamente na escolha do modelo de resposta a ser utilizado. Por

exemplo, quando deseja-se medir a habilidade do respondente, o formato mais utilizado é o de itens com resposta de múltipla escolha (MOREIRA JUNIOR, 2011).

Os itens precisam ser previamente calibrados, e, para tal, utilizam-se modelos da TRI para estimação de seus parâmetros. Por exemplo, quando utilizado o modelo logístico unidimensional de três parâmetros, 3PL, estimam-se a discriminação do item (a), dificuldade do item (b) e a probabilidade de acerto casual (c) (RICARTE, 2013). Para garantir uma boa estimação da habilidade dos respondentes, o TAC necessita de um banco de itens composto por um número elevado de itens em todos os níveis de habilidade, ou seja, possua itens de nível fácil, médio e difícil, pois com isso, independentemente da habilidade do respondente, a sua estimativa tende a ter um erro padrão pequeno (MOREIRA JUNIOR, 2011). Os itens precisam conter qualidade pedagógica e psicométrica, e por isso devem ser desenvolvidos, ou ao menos revisados, por especialistas no assunto abordado (LABARRÈRE; DA-SILVA; COSTA, 2011).

Apesar de, na prática, o banco de itens usualmente possuir entre 7 e 10 vezes o tamanho do teste, não há uma definição clara quanto a quantidade de itens necessária para a realização do TAC e tampouco a distribuição da habilidade no teste, podendo haver casos em que se considere a média de habilidade, conhecida *a priori*, da população, outros em que a maioria dos itens possam ser respondidos por 60% a 70% dos respondentes, e também casos em que os itens devam seguir uma distribuição uniforme de probabilidade, por exemplo (MOREIRA JUNIOR, 2011).

O banco de itens precisa ser revisado periodicamente para manter seu nível de precisão. Para tal, a inclusão e a exclusão de itens do banco de itens pode ser feita constantemente. Quando se inclui um novo item não calibrado em um TAC, esse item não é utilizado no processo de estimação da habilidade do respondente, mas a resposta dada a esse item é utilizada no processo de calibração deste novo item. Também é possível inserir itens já calibrados ao banco de itens e estes serão utilizados diretamente no teste, sem necessitar de nova calibração. Os novos itens serão incluídos na escala existente por meio de um método de ligação (MOREIRA JUNIOR, 2011).

2.1.2 Critério inicial do teste

Para dar início ao teste, o TAC necessita de uma estimativa da habilidade do respondente. Na maioria dos casos não há informação alguma sobre a habilidade do respondente, então define-se *a priori* uma habilidade, θ , dentro de uma determinada escala, como por exemplo a escala de uma distribuição Normal, com média zero e desvio padrão um, para a população. Pode-se iniciar o processo, selecionando um item do banco de itens do nível de habilidade médio, por exemplo, e a partir de cada resposta informada estima-se a sua habilidade, $\hat{\theta}$, e apresenta-se uma nova questão com a habilidade atualizada do respondente (RICARTE, 2013). A escolha do nível do primeiro item pode ocorrer de diversas formas, como seleção aleatória, a média da distribuição *a priori*, o nível de corte se existir, um item de nível baixo para motivar o respondente, pode-se pedir, também, que esse selecione o seu nível inicial ou até mesmo aplicar um pequeno teste comum a todos os respondentes para estimar o nível de habilidade do item a ser apresentado, chamado de pré TAC.

Ressalta-se que a qualidade do critério inicial muda de acordo com o objetivo do julgador e do método a ser utilizado na estimação da habilidade. Quanto maior a diferença entre o nível do respondente e o nível da primeira pergunta, mais perguntas deverão ser respondidas para finalizar o processo de estimação com um baixo erro padrão associado. Isto afeta o resultado final de testes cujos critérios de parada envolvem tamanho ou tempo, por exemplo (MOREIRA JUNIOR, 2011).

2.1.3 Critério de seleção do item

Depois de iniciado o seu processo e apresentada a primeira pergunta, o TAC precisa de um critério de seleção dos itens seguintes, pré-definido, para selecionar o próximo item a ser ministrado ao respondente (MOREIRA JUNIOR, 2011).

Como possível critério para a seleção dos próximos itens tem-se o da Máxima Informação (MI) que utiliza a medida de Informação de Fisher para o seu cálculo. Esse

é o critério mais utilizado (RICARTE, 2013). Pode-se utilizar também o critério da Máxima Informação Global, que utiliza como medida a Informação de *Kullback-Leibler* (KL), ideal quando a amostra das respostas do examinando ainda é pequena, pois o MI pode ser pouco eficiente se a estimativa inicial da habilidade não for próxima ao valor verdadeiro como quando, por exemplo, a habilidade do respondente for igual a três e o valor inicial igual a zero, o que acarretará em um maior número de itens aplicados e dependendo do critério de parada do teste, afetará a precisão da estimativa da habilidade (LABARRÈRE; DA-SILVA; COSTA, 2011). Um terceiro critério utilizado é a Informação a *Posteriori* Ponderada, que considera o valor da função de informação do teste em relação a *posteriori* da habilidade (RICARTE, 2013).

Um problema que pode ser encontrado quando se inicia um TAC utilizando, por exemplo, uma estimativa média da habilidade para todos os respondentes e a MI como critério de seleção dos itens, é que todos os respondentes responderão ao mesmo item inicial, e como o segundo item provém de acerto ou erro do primeiro item, pessoas que realizarem o teste em outras oportunidades poderão ter conhecimento das primeiras perguntas que irão ser submetidas, fazendo com que o erro das estimativas seja inflacionado (LABARRÈRE; DA-SILVA; COSTA, 2011). Este problema pode ser resolvido utilizando um critério de exposição dos itens.

2.1.4 Critério de exposição do item

Um dos problemas que decorrem da aplicação de um TAC, é que alguns itens podem ser superexpostos em detrimento a todo o restante do banco de itens. Não é incomum que entre 15% e 20% das questões do banco de itens representem mais de 50% dos itens de um teste. Isso ocorre porque esses itens são os que melhor discriminam os respondentes (WAINER, 2000). Estas questões podem passar a ser conhecidas, causando um viés positivo no erro estimado de quem possui tal conhecimento prévio (MOREIRA JUNIOR, 2011; MOREIRA et al., 2013). Para evitar esta possível falta de segurança do teste, em casos onde os itens devem ser sigilosos, podem-se aplicar métodos que limitem a taxa de exposição de cada item, ou seja, a

proporção de vezes em que determinado item aparece nos testes (MOREIRA JUNIOR, 2011). O procedimento *Sympson-Hetter*, descrito no item 2.1.4.1, calcula parâmetros de exposição do item para controlar probabilisticamente a frequência com que um item é selecionado.

Essas limitações também têm como objetivo aumentar a exposição de itens pouco expostos, pois se o item está em um banco de itens previamente elaborado e calibrado, deve-se tentar garantir que este item tenha uma taxa razoável de exposição, garantindo maior variedade nos diferentes testes aplicados com o mesmo banco (COSTA, 2009).

Outra vantagem em limitar a exposição de certos itens, é poder controlar o conteúdo dos testes, podendo, por exemplo, conferir a um respondente um teste com muitas questões de um assunto e poucas de outro, ou balancear o teste, forçando a apresentação de itens com assuntos ainda não apresentados ou menos apresentados, dependendo do interesse do julgador (MOREIRA JUNIOR, 2011).

Como as limitações implicam em perda de informação do teste, é importante que os itens do banco sejam de qualidade, para que quando determinado item não possa ser administrado devido à alguma restrição, o processo de seleção possa escolher um novo item com, também, alto poder de informação, mesmo que não seja o item que melhor discrimine os respondentes (LABARRÈRE; DA-SILVA; COSTA, 2011).

2.1.4.1 Critério de exposição *Sympson-Hetter* (LABARRÈRE; DA-SILVA; COSTA, 2011).

O algoritmo de *Sympson-Hetter* pode ser descrito por 8 passos necessários para sua aplicação, são eles:

- 1) Especificar a taxa de exposição máxima esperada de um item no teste (r), ou seja, qual é o percentual máximo de testes em que se espera que o item apareça;

- 2) Gerar uma tabela dos itens, separando-os por nível de habilidade e ordenando-os por sua informação;
- 3) Criar um vetor com tamanho igual a quantidade de itens, I , com todos os elementos iguais a 1. Este vetor representa o parâmetro de exposição, que corresponde a probabilidade do item ser apresentado (A), dado que foi selecionado (S). É representado por $P(A|S)$;

Os próximos passos são aplicados iterativamente até o critério de parada ser satisfeito:

- 4) Simular um TAC para uma amostra aleatória de respondentes, identificar o item i do TAC que possui informação mais próxima da estimativa de habilidade do respondente simulado. Gerar um número x pseudoaleatório da distribuição Uniforme $(0,1)$. Administrar o item i , se x for menor ou igual $P_i(A|S)$. Mesmo que o item i não seja apresentado ao respondente, ele não poderá mais ser selecionado para o teste deste respondente;
- 5) Observar a quantidade de vezes em que cada item é selecionado (NS) e a quantidade de vezes em que cada item é apresentado (NA) em todas as amostras aplicadas. Após o fim das simulações calcula-se para cada item a probabilidade de o item ser selecionado ($P(S) = (NS/N)$) e a probabilidade de o item ser apresentado ($P(A) = (NA/N)$), onde N representa o número de respondentes na simulação;
- 6) Com o valor de r , informado no passo 1, e com a $P(S)$, recalcula-se os valores de $P_i(A|S)$, como:

$$P_i(A|S) = \begin{cases} \frac{r}{P(S)} & \text{se } P(S) > r. \\ 1 & \text{se } P(S) \leq r. \end{cases}$$

- 7) Para um TAC de tamanho n , deve-se assegurar que ao menos n itens tenham nova $P_i(A|S)=1$, desta maneira, esses itens serão sempre apresentados, pois o número gerado aleatoriamente será sempre menor ou igual a 1. Quando esse requisito não é satisfeito, devem-se igualar os n maiores valores de $P_i(A|S)$ a 1. Desta maneira, assegura-se que n itens serão apresentados ao respondente antes do banco de itens ser completamente selecionado;

- 8) Calculados os novos valores de $P_i(A|S)$ e utilizando a mesma amostra de respondentes, refazem-se os passos 4 a 7 até que o máximo valor de $P(A)$ aproxime-se do limite de r , e, então oscile em sucessivas simulações.

2.1.5 Estimativa da habilidade

A habilidade do respondente pode ser calculada de diversas maneiras em um TAC, seja por métodos de estimativas pontuais ou bayesianas. Nos métodos de estimativas pontuais, usualmente encontra-se o Método da Máxima Verossimilhança (MV), calculada por meio de um método iterativo, como, por exemplo, o Newton-Raphson ou o *Scoring* de Fisher. Já nos métodos de estimativas bayesianas, podem ser encontradas a estimação pelo máximo da *posteriori* (MAP) ou a estimação pela média da *posteriori* (EAP) (ANDRADE; TAVARES; VALLE, 2000).

Os estimadores de MV das habilidades convergem para uma distribuição normal com média igual ao verdadeiro valor das habilidades e variância igual a inversa da Matriz Hessiana, no caso do processo iterativo de Fisher (OLIVEIRA FILHO; MENEZES, 2006).

Algumas diferenças observadas entre os métodos bayesianos e os métodos de estimativas pontuais nos mostram que os métodos da Máxima Verossimilhança apresentam maior erro padrão, menor eficiência e necessitam de um maior tempo para os cálculos computacionais em relação aos métodos bayesianos. Porém apresentam menor viés e menor correlação entre os valores estimados e os parâmetros (MOREIRA JUNIOR, 2011). Com isso, depende-se do interesse do julgador para decidir qual o método de estimação a ser utilizado.

Para calcular as estimativas das habilidades pelo método EAP, utilizam-se pontos de quadratura para aproximar a distribuição *a priori* das habilidades de cada respondente. Ao se utilizar este método, obtém-se uma estimativa da distribuição de habilidade da população de respondentes, na forma de uma distribuição discreta, dada pelos pontos de quadratura. Esta distribuição é obtida acumulando-se as

densidades da *posteriori* de todos os respondentes em cada ponto de quadratura. As somas são então normalizadas para se obter as probabilidades estimadas em cada ponto, e são fornecidos a média e o desvio-padrão para essa distribuição estimada (SANTOS, 2008).

O método de estimação EAP tem a vantagem de ser calculado diretamente, não precisando da aplicação de métodos iterativos (ANDRADE; TAVARES; VALLE, 2000).

2.1.6 Critério de parada do teste

Os procedimentos de seleção e aplicação dos itens no teste são feitos repetidamente, até que algum critério de parada seja satisfeito ou que todos os itens sejam selecionados (COSTA, 2009).

O critério de parada do teste está diretamente relacionado com o objetivo do teste, características dos itens e as restrições operacionais. Os diferentes critérios de parada permitem que o teste seja de tamanho fixo ou variável. Não necessariamente, precisa-se adotar apenas um critério de parada para o teste, sendo encerrado quando um, todos ou uma combinação desses ocorrer (MOREIRA JUNIOR, 2011).

Um critério de parada para um teste de tamanho fixo possível de ser utilizado é o número máximo de itens respondido pelo respondente. Nestes casos a tendência é que o erro padrão das estimativas das habilidades dos respondentes nos extremos da escala de habilidade seja inflacionada, pois em alguns casos o respondente com habilidade alocada em um dos extremos necessitaria de um número elevado de itens para ter uma boa estimativa de sua habilidade (MOREIRA JUNIOR, 2011; RICARTE, 2013).

Outro possível critério para um teste de tamanho fixo poderia ser o tempo de aplicação do teste. Porém, neste caso, um respondente mais vagaroso poderia ser prejudicado (LABARRÈRE; DA-SILVA; COSTA, 2011).

Estes critérios para testes de tamanho fixo também são indicados em casos onde o banco de itens seja de tamanho pequeno ou médio.

Já para os casos em que o teste seja de tamanho variável, utiliza-se normalmente um valor fixo no erro padrão da estimativa da habilidade do respondente. Neste caso, todas as habilidades serão estimadas com erro padrão equivalente, e com isso a comparação entre os respondentes poderá se dar de forma direta, ou seja, o respondente com maior habilidade estaria alocado em primeiro lugar, por exemplo, em um ranking (MOREIRA JUNIOR, 2011). O problema deste critério ocorre em casos onde o banco de itens não é suficientemente grande, pois podem acontecer casos em que o erro padrão associado a estimativa da habilidade do respondente não convirja ao valor esperado para o erro especificado, com isso a estimativa da habilidade deixa de ser comparável diretamente com as outras.

Outro critério possível em testes de tamanho variável, seria o teste reconhecer um comportamento inadequado do respondente, como, por exemplo, responder todas as questões com a mesma alternativa, responder de forma muito rápida ou muito devagar os itens (MOREIRA JUNIOR, 2011). Pode-se utilizar, também, esse tipo de critério quando deseja-se comparar a habilidade do respondente a algum tipo de ponto de corte (classificado/desclassificado, reprovado/aprovado). Assim, se o intervalo de confiança da habilidade estiver acima do ponto de corte este respondente estará, por exemplo, classificado.

2.1.7 Vantagens e desvantagens na aplicação de um TAC

Os testes adaptativos possuem vantagens em relação aos testes da teoria clássica dos testes, tais como (COSTA, 2009; WAINER, 2000):

- a) Redução no tempo e no tamanho do teste mantendo o mesmo nível de confiança;
- b) Controle sobre a exposição dos itens no teste;

- c) Flexibilidade na aplicação das provas, pois permite que os respondentes respondam a provas em dias diferentes ou de forma simultânea em qualquer lugar do mundo;
- d) Rapidez e precisão na correção;
- e) Possibilidade de gerar relatórios com o resultado logo após o fim do teste de cada respondente;
- f) Possibilidade de adicionar itens a um TAC que já foi utilizado;
- g) Utilização de itens em diferentes formatos, tais como gráficos, multimídia, aberto, entre outros;
- h) Segurança, evitando possíveis danos físicos, perda ou extravio do teste e podendo criptografar os dados;
- i) Produtividade dos respondentes, pois estes respondem itens com um nível de habilidade apropriado, não perdendo tempo em itens por eles considerados muito fáceis ou muito difíceis.

Porém, os testes adaptativos possuem certas desvantagens em relação aos testes tradicionais, como por exemplo (COSTA, 2009):

- a) Necessita-se de um banco de itens suficientemente grande e calibrado para se manter o nível de precisão em todos os testes aplicados;
- b) Atualização constante do banco de itens o que demanda recursos humanos e financeiros.

3 MATERIAL E MÉTODOS

3.1 CONJUNTO DE DADOS

3.1.1 Conjunto de dados futebol

O conjunto de itens e a escala para interpretação da habilidade do respondente, a serem utilizados, foram aplicados na disciplina de Teorias de Avaliação do curso de Estatística da Universidade Federal do Paraná, e continha 80 questões com o tema futebol, as quais foram respondidas por 710 pessoas.

Para a execução da metodologia do TAC, foi utilizado este banco de itens, que após ser calibrado com o modelo logístico unidimensional de três parâmetros (3PL), ficou com 66 itens. Cada item contém cinco opções de resposta, em que apenas uma é a correta. Este banco de itens foi utilizado com a finalidade de demonstrar o funcionamento de um TAC.

A escala gerada para a interpretação da estimativa da habilidade do respondente contém cinco níveis de habilidade e a interpretação para tais níveis é da seguinte forma (FORBELLONE; INÁCIO, 2013):

- a) Respondentes com habilidade menor que -1 conhecem sobre símbolos e mascotes de times nacionais, e também sobre os jogadores que fizeram parte da seleção brasileira;
- b) Respondentes com habilidade entre -1 e 0, conhecem sobre o que foi descrito no item anterior e também sobre títulos da seleção brasileira, sobre as seleções que conquistaram a Copa do Mundo, sobre técnicos brasileiros que atuaram em outras seleções nacionais e possuem um pouco conhecimento sobre times internacionais;
- c) Respondentes com habilidade entre 0 e 1, conhecem sobre o que foi descrito nos itens anteriores e também possuem conhecimento mais

aprofundado sobre conquistas de times nacionais, conhecem sobre os principais jogos da seleção brasileira e conhecem em termos gerais sobre times internacionais;

- d) Respondentes com habilidade entre 1 e 2, conhecem sobre o que foi descrito nos itens anteriores e também possuem conhecimento mais aprofundado de conquistas de times e seleções internacionais;
- e) Respondentes com habilidade maior que 2, conhecem sobre tudo o que foi descrito nos itens anteriores e também possuem conhecimento mais aprofundado sobre seleções internacionais.

3.1.2 Conjunto de dados simulados

Foi simulado pelos autores um banco de itens com mil itens de múltipla escolha e o modelo unidimensional de três parâmetros associado aos itens, com isso foram simulados os três parâmetros relacionados a cada um dos mil itens. Esses itens foram utilizados para demonstrar a performance, no pacote *mirtCAT* (CHALMERS, 2015a), entre os métodos de estimação da habilidade dos respondentes e também dos critérios de seleção dos itens. Foram simulados cem padrões de respostas para cada nível de habilidade presente no banco de itens e foi observada a precisão da estimativa da habilidade média dos respondentes em cada cenário e a quantidade média de itens necessária para tal estimativa utilizando um erro padrão menor ou igual a 0,2 para todos os pares de métodos de estimativa e critérios de seleção dos itens.

3.2 MODELO LOGÍSTICO UNIDIMENSIONAL DE 3 PARÂMETROS

Sobre o pacote *mirtCAT* (CHALMERS, 2015a) foram utilizados métodos relativos à Teoria de Resposta ao Item, aplicado ao modelo logístico unidimensional de três parâmetros. (1)

$$P(U_{ij} = 1|\theta_j) = c_i + (1 - c_i) \frac{1}{1 + e^{-Da_i(\theta_j + b_i)}}, \quad (1)$$

com $i = 1, 2, \dots, l$, e $j = 1, 2, \dots, n$, onde:

U_{ij} é uma variável dicotômica que assume os valores 1, quando o respondente j responde corretamente ao item i e 0 quando o respondente j não responde corretamente ao item i ;

θ_j representa a habilidade do j -ésimo respondente;

$P(U_{ij} = 1|\theta_j)$, chamada de Função de Resposta do Item (FRI), é a probabilidade de um respondente j com habilidade θ_j responder corretamente ao item i ;

c_i é o parâmetro do item que representa a probabilidade de um respondente com baixa habilidade responder corretamente o item i . Também é referido como a probabilidade de acerto casual;

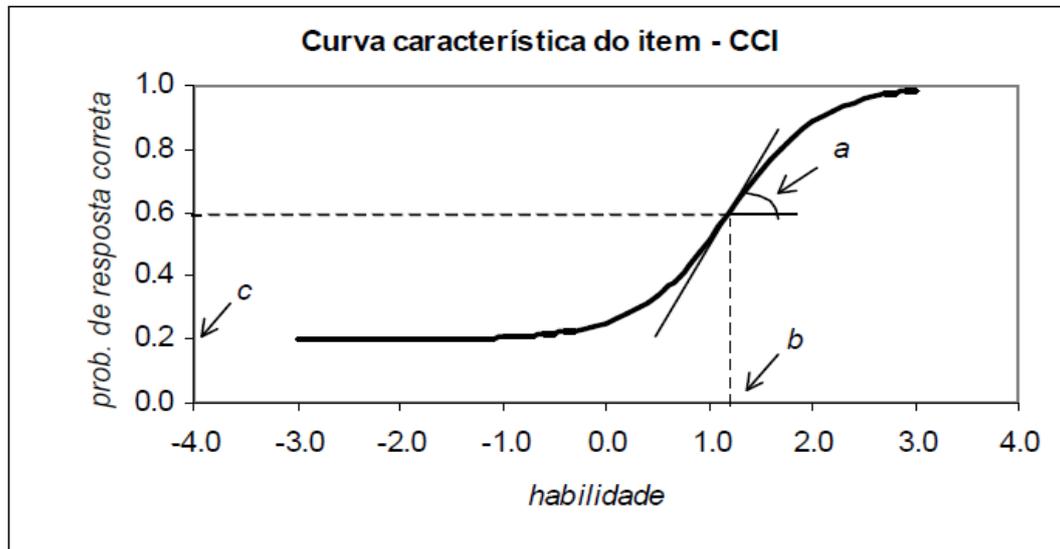
b_i é o parâmetro de dificuldade (ou de posição) do item i , medido na mesma escala da habilidade;

a_i é o parâmetro de discriminação do item i (ou de inclinação da curva), com valor proporcional à inclinação da Curva Característica do Item no ponto b_i ;

D é um fator escala, constante e igual a 1. Utiliza-se o valor 1,7 quando se deseja que a função logística forneça resultados semelhantes ao da função ogiva normal.

Os parâmetros do modelo são ilustrados pela FIGURA 2, conhecida como a Curva Característica do Item (CCI).

FIGURA 2 - CURVA CARACTERÍSTICA DO ITEM



Fonte: ANDRADE; TAVARES; VALLE, 2000.

3.3 RECURSOS COMPUTACIONAIS

Os procedimentos do TAC foram aplicados no R (R DEVELOPMENT CORE TEAM, 2015) por meio do pacote *mirtCAT* (CHALMERS, 2015a) que possui suas funções destinadas a esta metodologia. Os resultados finais foram apresentados em uma aplicação desenvolvida no pacote *shiny* (CHANG et al., 2015).

3.3.1 Pacote *mirtCAT* (CHALMERS, 2015a)

O pacote *mirtCAT* do R (R DEVELOPMENT CORE TEAM, 2015) foi desenvolvido para a criação de testes adaptativos e não adaptativos para avaliações educacionais e psicológicas, por meio do pacote *shiny* (CHANG et al., 2015). O *mirtCAT* (CHALMERS, 2015a) possibilita ao julgador a geração de TACs unidimensionais ou multidimensionais através de modelos da TRI e também questionários através da teoria clássica para coletar as respostas diretamente no R.

O TAC mais simples possível de ser aplicado necessita somente do modelo gerado pela função *generate.mirt_object*, e do *dataframe* com todos os itens, suas respostas, suas opções de resposta e o tipo de questão a ser apresentada, ou seja, questões abertas, de múltipla escolha, gráficas, entre outras. Todos os outros argumentos obrigatórios para o processo do TAC possuem uma forma padrão e, caso não seja especificado nada a eles, esses padrões são utilizados durante o processo. Estes argumentos podem ser alterados e servem para o usuário definir como será o teste, por exemplo, se o mesmo será adaptativo ou não adaptativo e/ou será unidimensional ou multidimensional.

O pacote possui apenas sete funções e o uso das mesmas se dá de forma direta ou, em alguns casos, indireta, pois o pacote não necessita do uso de todas as funções para a aplicação de um TAC. As funções e seus principais argumentos são descritos a seguir:

3.3.1.1 Função *createShinyGUI*

Esta função retorna os resultados da estrutura GUI, chamando o *shinyApp*. A princípio, esta função só é útil quando a aplicação GUI está armazenada em domínio público. Essa função deve ser chamada apenas após a função *mirtCAT_Preamble* ser executada.

```
>createShinyGUI<-function() {return(shinyApp(ui=ui(),server=server)) }
```

3.3.1.2 Função *findNextItem*

A função *findNextItem* retorna um vetor numérico com a próxima questão que deverá ser apresentada ao respondente. Esta função é utilizada de forma indireta pela função *mirtCAT*, atrelada ao argumento de critério de seleção dos itens utilizado.

3.3.1.3 Função *generate.mirt_object*

A função *generate.mirt_object* é utilizada para gerar um objeto padrão com os parâmetros dos itens conhecidos e este objeto é utilizado pela função *mirtCAT* para a aplicação do teste. A função necessita de um *dataframe* com os parâmetros estimados pela TRI e o modelo que será utilizado no processo do TAC. O objeto resultante da função é um dos argumentos obrigatórios da função *mirtCAT*.

Os argumentos da função são: *parameters*, *itemtype*, *latent_means*, *latent_covariance*, *key* e *min_category*. Os dois primeiros argumentos são obrigatórios para a função funcionar, os demais são opcionais.

3.3.1.3.1 Argumento *parameters*

O argumento chamado de *parameters* recebe uma matriz ou um *dataframe* com os parâmetros definidos pela TRI, onde cada coluna representa o(s) parâmetro(s) associado(s) a cada item em sua respectiva linha. A quantidade de colunas desta matriz, ou *dataframe*, depende do modelo associado aos itens e o número de linhas é igual ao número de itens do teste. No modelo unidimensional de três parâmetros, a matriz, ou *dataframe*, terá três colunas, uma representando o parâmetro de discriminação (a_1), outra representando o negativo da multiplicação entre os parâmetros de discriminação e dificuldade (d), e a última coluna representa a probabilidade de acerto casual (g).

3.3.1.3.2 Argumento *itemtype*

O argumento chamado de *itemtype*, recebe o modelo associado aos itens. Pode-se utilizar diferentes tipos de modelos como, por exemplo, o modelo logístico

unidimensional de dois parâmetros (2PL), o modelo logístico unidimensional de três parâmetros (3PL), o modelo logístico unidimensional de quatro parâmetros (4PL), o modelo de *Rash* e o modelo de escala gradual (*Graded*).

```
> pars <- data.frame(a1=a, d=d, g=g)
> mod <- generate.mirt_object(parameters=pars, itemtype='3PL')
```

3.3.1.4 Função *generate_pattern*

A função *generate_pattern* é utilizada para criar simulações de respostas dos respondentes. Os argumentos da função são *mo*, *Theta* e *df*, onde os dois primeiros são obrigatórios. O argumento *mo* recebe o objeto gerado pela função *generate.mirt_object*, o *Theta* recebe um valor numérico ou vetor vertical de valores, indicando os valores das habilidades dos supostos respondentes. Já *df* recebe o *dataframe* com as perguntas, respostas, opções de respostas e o tipo dos itens.

Se for declarado na função o argumento *df*, o argumento *Theta* deverá receber apenas um valor, e não um vetor. Com o argumento *df* presente, o retorno da função é um vetor horizontal tipo caractere com *n* colunas, onde *n* é o número de itens banco de itens. A coluna *i* possui a resposta simulada que alguém com nível de habilidade *theta* daria para o item *i*. No QUADRO 1, observa-se como usar a função com o argumento *df* e o objeto retornado por ela, em um exemplo onde o tamanho do banco de itens é dez e cada item possui cinco opções de resposta.

QUADRO 1 – RESPOSTAS SIMULADAS DE UM RESPONDENTE COM *THETA* IGUAL A 1

```
> pat<- generate_pattern(mo = mod, Theta = 1,df = df)
> pat
[1] "Opção 2" "Opção 1" "Opção 1" "Opção 5" "Opção 2" "Opção 4" "Opção 1" [
9] "Opção 5" "Opção 1" "Opção 3"
```

Se o argumento *df* não estiver presente na função, é possível atribuir um vetor indicando diversos *thetas* para o argumento *Theta*. Com apenas os argumentos obrigatórios da função, esta retornará uma matriz de dimensões *jxn*, onde *j* é o número de *thetas* passado ao argumento *Theta* e *n* é o número de itens do banco de dados. Os valores da matriz, 0 ou 1, indicam se, na simulação, um respondente com

determinado θ erraria (0) ou acertaria (1) determinado item. O quadro abaixo mostra simulações de respostas para três respondentes com diferentes níveis de habilidade ($\theta_1 = -0,7$, $\theta_2 = 0$ e $\theta_3 = 0,7$). O banco de itens possui dez perguntas que estão em ordem crescente de dificuldade.

QUADRO 2 – RESPOSTAS SIMULADAS DE RESPONDENTES COM DIFERENTES θ ETAS.

```
> pat <- generate_pattern(mo = mod, Theta = cbind(c(-0.7,0,0.7)))
> pat
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	0	1	0	0	1	0	0	0	0
[2,]	1	1	0	0	1	0	0	1	0	0
[3,]	1	1	1	1	1	1	0	0	1	0

3.3.1.5 Função *mirtCAT*

A função *mirtCAT* inicia e aplica todos os passos necessários de um TAC, essa função possui vários argumentos destinados a diferentes tipos de testes e aplicações, conforme descritos a seguir.

3.3.1.5.1 Argumento *df*

O argumento *df* recebe um *dataframe* com ao menos os itens do teste (*question*), e o tipo de item (*type*) que será apresentado. Pode-se também adicionar ao *dataframe* uma coluna com as respostas dos itens (*answer*), e, no caso de testes de múltipla escolha, as opções possíveis de respostas que serão apresentadas ao usuário (*option*).

Os argumentos *question*, *answer* e *option* recebem vetores com os itens do teste, as respostas para os itens e as opções de resposta, respectivamente. O argumento *type* recebe o tipo de item a ser apresentado. Os possíveis tipos de itens são:

- a) *radio*: Quando neste formato, as opções de respostas são dispostas verticalmente, podendo ser selecionada apenas uma resposta;
- b) *radio_inline*: Opção semelhante à anterior, com a diferença de que os itens são dispostos horizontalmente;
- c) *Select*: As opções se encontram em uma caixa de seleção, devendo ser escolhida uma dessas;
- d) *Text*: não são apresentadas opções de resposta, apenas uma caixa de texto para o respondente digitar a resposta que julga ser a correta.

```
> df <- data.frame(question=questions, answer=answer,
+ option=choices, Type='radio')
> TAC <- mirtCAT(df=df, ...)
```

3.3.1.5.2 Argumento *mo*

O argumento *mo* recebe o objeto gerado pela função *generate.mirt_object* descrita anteriormente.

```
> TAC <- mirtCAT(mo=mod, ...)
```

3.3.1.5.3 Argumento *method*

O argumento *method* recebe o método que será utilizado na estimação da habilidade do respondente. O padrão do pacote é o método média a *posteriori* 'MAP'. Pode-se também utilizar o método da esperança a *posteriori* 'EAP' e o método da máxima verossimilhança 'ML', como alternativa em testes unidimensionais.

```
> TAC <- mirtCAT(method='EAP', ...)
```

3.3.1.5.4 Argumento *criteria*

O argumento *criteria* recebe o método de seleção dos itens que será utilizado. Para testes não adaptativos, utiliza-se o critério chamado 'seq', que apresentará os itens de forma sequencial. Para testes adaptativos unidimensionais, o pacote possui vários critérios disponíveis, como o da máxima informação 'MI', o mínimo esperado pela variância a *posteriori*, 'MEPV', método da verossimilhança ponderada 'MLWI', o da máxima informação ponderada a *posteriori* 'MPWI', o da máxima informação esperada 'MEI', o da integração baseada na medida de *Kullback-Leibler* 'KL', e os de integrações baseada na medida de *Kullback-Leibler* ponderada 'IKLn' e 'IKLPn'.

```
> TAC <- mirtCAT(criteria='MI', ...)
```

3.3.1.5.5 Argumento *start_item*

O argumento chamado *start_item* recebe o item a ser primeiramente apresentado e possui duas opções: a primeira representa o número do item que se deseja iniciar o teste e a segunda opção é o primeiro item ser selecionado por algum critério de seleção dos itens.

```
> TAC <- mirtCAT(start_item='MI', ...)
```

3.3.1.5.6 Argumento *local_pattern*

O argumento *local_pattern* recebe um vetor de texto ou numérico com as respostas padrões para a aplicação de um TAC sem gerar a interface *GUI*. Esta opção requer um vetor completo de respostas, conforme os itens foram propostos no modelo. Para utilização desse argumento na forma de vetor de texto, o argumento *df* é obrigatório, e o texto usado como resposta de cada um dos itens deve ser correspondente a uma alternativa daquele item. Para utilizar um vetor numérico, o argumento *df* não pode ser inserido, ou deve receber *NULL*, e os valores devem ser

0, para indicar que aquele item será respondido errado, caso selecionado, ou 1, para indicar acerto do item, da mesma forma.

```
# Exemplo onde se deseja que a resposta dada ao item 1 do banco de
# itens seja "Opção 2", ao item 2 # seja "Opção 5", e assim por
# diante.
> resp.dadas<-c("Opção 2", "Opção 5", "Opção 1", "Opção 3", "Opção
+ 2", "Opção 4",...)
> resp <- mirtCAT(df = df, local_pattern = resp.dadas, ... )

# Exemplo onde se deseja que a resposta dada ao item 1 do banco de
# itens esteja certa, o item 2 esteja errada, e assim por diante
> resp.dadas <- c(1, 0, 1, 1, 0, 1, 0, 1, 1,...)
> resp <- mirtCAT( df = NULL, local_pattern = resp.dadas, ... )
```

3.3.1.5.7 Argumento *design_elements*

O argumento *design_elements* é um argumento lógico, que quando utilizado retorna um objeto que contém os elementos do teste e do respondente.

3.3.1.5.8 Argumento *design*

O argumento chamado de *design* representa uma lista de argumentos que serão utilizados no TAC:

a) Argumento *min_SEM*

O argumento representa o erro padrão da estimativa da habilidade do respondente, este pode ser utilizado como critério de parada para o teste. O padrão do pacote é 0.3, ou seja, quando a estimativa da habilidade atinge um erro padrão menor que 0.3, o teste é encerrado. O argumento recebe um valor entre zero e um.

```
> TAC <- mirtCAT(design=list(min_SEM=0.25, ...),...)
```

b) Argumento *delta_thetas*

O argumento recebe um valor que será utilizado na comparação da estimativa da habilidade do respondente entre duas questões consecutivas para um critério de parada para o teste. Por exemplo, se a diferença entre estimativa da habilidade do respondente na questão n com a questão $n-1$ for menor que o valor que o argumento *delta_thetas* recebeu, o teste será encerrado. Se o *theta* na questão n for igual a 1,35 e o *theta* na questão $n-1$ for igual a 1,32 e o valor de *delta_thetas* passado a função for igual a 0,05, o teste finaliza, pois 0,03 é menor que 0,05.

```
> TAC <- mirtCAT(design=list(delta_thetas=0.05, ...),...)
```

c) Argumento *thetas.start*

O argumento recebe a estimativa do *theta* inicial do item. O padrão é 0, neste caso todos os respondentes iniciarão o teste com um item de *theta* próximo ou igual a 0.

```
> TAC <- mirtCAT(design=list(thetas.start=1, ...),...)
```

d) Argumento *min_items*

O argumento *min_items* recebe a quantidade mínima de itens que será apresentada ao respondente, o padrão do pacote é um item.

```
> TAC <- mirtCAT(design=list(min_items=5, ...),...)
```

e) Argumento *max_items*

O argumento *max_items* recebe a quantidade máxima de itens que será apresentada ao respondente.

```
> TAC <- mirtCAT(design=list(max_items=20, ...),...)
```

f) Argumento *classify*

O argumento *classify* recebe um vetor numérico que indica valores de corte para uma classificação, utilizando o valor estimado da habilidade do respondente e considerando valores acima e abaixo dos indicados para a classificação.

```
> TAC <- mirtCAT(design=list(classify=c(-1,1), ...),...)
```

g) Argumento *classify_CI*

O argumento *classify_CI* recebe um vetor numérico, com valores entre 0 e 1, indicando o valor de z que será utilizado na criação dos intervalos de confiança na classificação. Por exemplo, se indicado o valor de 0,95, o intervalo de confiança terá 95% de confiança.

```
> TAC <- mirtCAT(design=list(classify_CI=0.90, ...),...)
```

h) Argumento *exposure*

O argumento *exposure* recebe um vetor com a quantidade máxima de exposição para cada item. Existem duas opções para este argumento: a primeira é um vetor com números maiores que um, que representam a quantidade de vezes que este item poderá aparecer durante a aplicação do TAC aos respondentes. Se o vetor receber números entre 0 e 1 para a exposição dos itens, o método de controle de exposição *Sympson-Hetter* será aplicado.

```
> TAC <- mirtCAT(design=list(exposure=rep(0.3,n),...),...)  
# com n igual ao número de itens
```

i) Argumento *constraints*

O argumento *constraints* recebe uma lista de restrições na seleção dos itens para cada particular item do TAC. As opções possíveis são listadas a seguir:

i. Argumento not_scored

O argumento *not_scored* recebe um vetor com os itens que podem ser selecionados, porém não serão utilizados na estimação da habilidade do respondente, pois ainda estão em fase de estimação de seus parâmetros ou testes para TACs futuros.

```
> TAC <-  
+ mirtCAT(design=list(constraints=list(not_scored=c(1,3,5,10),...),...)  
# Os itens 1,3,5,10 não serão utilizados na estimativa da habilidade  
#dos respondentes.
```

ii. Argumento independent

O argumento *independent* recebe itens que nunca devem ser apresentados em uma mesma sessão de TAC, pois possuem características muito parecidas.

```
> TAC <-
+ mirtCAT(design=list(constraints=list(independent=c(1,3),...),...))
# Os itens 1,3 não serão utilizados no mesmo TAC.
```

iii. Argumento *ordered*

O argumento *ordered* recebe um grupo particular de itens que devem ser apresentados ordenadamente após algum item especificado ser selecionado.

```
> TAC <- mirtCAT(design=list(constraints=list(ordered=c(3,1,7),...),...))
# Caso o item um, o três ou o sete seja selecionado, será
# apresentado ao respondente o item três, seguido do item um e
# depois do sete, independentemente de as respostas dadas estarem
# certas.
```

iv. Argumento *unordered*

O argumento *unordered*, como o *ordered*, recebe o mesmo grupo particular de itens, sendo, porém, apresentados de forma aleatória dentro do grupo.

```
> TAC <-
mirtCAT(design=list(constraints=list(unordered=c(2,4,8),...),...))
# Caso o item dois, o quatro ou o oito seja selecionado, esses três
# itens serão apresentados sequencialmente, em ordem aleatória,
# independentemente das respostas escolhidas.
```

j) Argumento *KL_delta*

Argumento válido quando *criteria* recebe KL ou KLn, é o valor de delta usado no cálculo desses dois critérios. O padrão do pacote é 0,1.

```
> TAC <- mirtCAT(design=list(criteria=KL, KL_delta=0,15,...))
```

k) Argumento *content*

Este argumento recebe um vetor de tamanho igual ao do número de itens, indicando qual o assunto abordado pelo item. Deve ser usado junto com o argumento *content_prop*.

l) Argumento *content_prop*

Vetor numérico nominal que indica a proporção de itens dos assuntos inseridos no argumento *content* que aparecerão no teste. A soma das proporções deve somar um.

Abaixo, pode-se observar um exemplo com um banco de 60 itens, onde os 20 primeiros são sobre português, seguidos de 20 de matemática e 20 de inglês. No

exemplo, deseja-se que metade dos itens apresentados no teste sejam de matemática, 25% sejam de português e 25% de inglês.

```
> TAC <- mirtCAT(design=list(content=c(rep("português",20),rep(
+"matemática",20),rep("inglês",20)),content_prop=c("matemática"=0.50
+"português"= 0.25,"inglês"=0.25),...))
```

3.3.1.5.9 Argumento *shinyGUI*

O argumento *shinyGUI* recebe uma lista de argumentos que serão utilizados no TAC e altera diretamente a interface gerada pelo shiny.

```
> shinyGUI <- list(title = title, authors = authors,
+ instructions=instructions, demographics = demographics,
+ demographics_inputIDs = c("Nome", "Sexo","Idade"), firstpage =
+ firstpage, lastpage = lastpage, temp_file=temp)
```

a. Argumento *title*

O argumento *title* recebe um vetor de texto com o título do teste.

```
> TAC <- mirtCAT(shinyGUI=list(title='Teste Adaptativo
+ Computadorizado',...),...)
```

b. Argumento *authors*

O argumento *authors* recebe um vetor de texto com o(s) nome(s) do(s) autor(es) do TAC.

```
> TAC <- mirtCAT(shinyGUI=list(authors='Lucas Bozza e Tiago
+ Teixeira',...),...)
```

c. Argumento *instructions*

O argumento *instructions* recebe três partes de um vetor de texto com instruções ao respondente de como usar o TAC.

```
> TAC <- mirtCAT(shinyGUI=list(instructions=c('Instruções',
+ 'Pressione o botão abaixo','Botão'),...),...)
```

d. Argumento *firstpage*

O argumento *firstpage* recebe uma lista de texto com informações que serão apresentadas na primeira página do teste.

```
> TAC <- mirtCAT(shinyGUI=list(firstpage=list(h3('TAC'),h5(Responda
+ as questões propostas'),...),...))
```

Se uma lista vazia for passada neste argumento, a primeira página não será apresentada.

e. Argumento *demographics*

O argumento *demographics* recebe um campo que coleta as informações pessoais dos respondentes que o pesquisador deseja conhecer, como por exemplo: nome, idade, sexo, entre outros.

```
>Demographics<-list(textInput(inputId="Nome",label="Insira seu nome
+e sobrenome:",value=""),selectInput(inputId="Sexo",label
+"Sexo",choices=c("Masculino","Feminino"),selected=""),numericInput(
+"Idade","Idade",value=18,min=5,max=111))
>TAC<-mirtCAT(shinyGUI=list(demographics=demographics,...),...)
```

f. Argumento *demographics_inputIDs*

O argumento *demographics_inputIDs* torna-se obrigatório na presença do argumento *demographics*, com os nomes dos campos que serão pedidos aos respondentes, como nome, idade, sexo, entre outros.

```
> demographics_inputsIDs <- c("Nome", "Sexo", "Idade")
> TAC <- mirtCAT(shinyGUI=list(demographics_inputsIDs=
+ demographics_inputIDs, ...),...)
```

g. Argumento *max_time*

O argumento *max_time* recebe o tempo máximo em segundos para responder ao teste.

```
> TAC <- mirtCAT(shinyGUI=list(max_time=120,...),...)
```

h. Argumento *temp_file*

O argumento *temp_file* recebe um vetor de texto indicando onde o arquivo com extensão *.rds* gerado pela função *mirtCAT* será armazenado temporariamente. Este argumento é utilizado somente durante a execução do TAC para salvar as respostas,

sendo atualizado a cada resposta fornecida pelo respondente. O *temp_file* evita a perda de informação quando algum problema acontece durante a execução do TAC, como por exemplo, uma queda de energia ou impossibilidade de o respondente terminar a prova naquele instante.

```
> temp <- paste0(getwd(), 'temp_file.rds')
> TAC <- mirtCAT(shinyGUI=list(temp_file=temp,...), ...)
```

i. Argumento *resume_file*

O argumento *resume_file* recebe um vetor indicando onde o arquivo temporário gerado pela função *temp_file* foi salvo. Este argumento permite ao respondente retornar ao teste exatamente de onde ele parou anteriormente, no item com a última resposta dada antes de parar.

```
> TAC <- mirtCAT(shinyGUI=list(resume_file=temp,...), ...)
```

j. Argumento *lastpage*

O argumento *lastpage* recebe uma função com as últimas informações a serem apresentadas ao respondente, logo após o fim do processo do TAC.

```
> lastpage <- function(person) {
+ nome <- as.vector(person$demographics$Nome)
+ theta <- as.vector(person$thetas)
+ return(list(h3(sprintf("Obrigado %s. Sua estimativa de habilidade é
+ %.2f.", nome, theta))))}
> TAC <- mirtCAT(shinyGUI=list(lastpage=lastpage,...), ...)
```

k. Argumento *forced_choice*

O argumento *forced_choice* recebe um teste lógico deixando a escolha por uma resposta como obrigatória ou facultativa, este argumento é utilizado somente em testes não adaptativos. O padrão é verdadeiro, ou seja, o respondente necessariamente precisa escolher dentre uma das alternativas possíveis em um teste de múltipla escolha.

```
> TAC <- mirtCAT(shinyGUI=list(forced_choice=FALSE,...), ...)
```

3.3.1.5.10 Argumento *preCAT*

O argumento *preCAT* recebe uma lista com um teste a ser aplicado antes de a sessão do TAC ser iniciada. O *preCAT* pode ser realizado para se obter uma estimativa da habilidade do respondente para iniciar o seu teste. É necessário informar ao argumento os seguintes critérios descritos anteriormente: *min_items*, *max_items*, *criteria* e *method*.

```
> TAC <- mirtCAT(preCAT=list(min_items=5, max_items=10,
+ criteria='MI', method='EAP'),...)
```

3.3.1.5.11 Retornos da função *mirtCAT*

Após o TAC ser finalizado é possível obter alguns resultados para um teste individual, são eles em seus respectivos objetos:

- a) Estimativas finais do *theta* e seu erro padrão (*\$final_estimates*);
- b) Vetor numérico com as respostas indicadas pelo respondente (*\$raw_responses*);
- c) Vetor numérico indicando se o item foi respondido corretamente (*\$scored_responses*);
- d) Vetor numérico com a sequência de itens respondidas (*\$items_answered*);
- e) Matriz com as estimativas das habilidades do respondente a cada item respondido (*\$thetas_history*);
- f) Matriz com os erros padrões das estimativas da habilidade do respondente a cada item respondido (*\$thetas_SE_history*);
- g) Vetor numérico com o tempo levado pelo respondente para finalizar cada questão proposta (*\$item_time*);
- h) *Dataframe* com as informações pessoais dos respondentes indicadas no início da aplicação (*\$demographics*);

- i) Classificação final dos respondentes baseada em algum critério pré-definido, linha de corte, para gerar a classificação com os respondentes acima e abaixo deste critério (*\$classification*).

3.3.1.6 Função *mirtCAT_preamble*

A função *mirtCAT_preamble* é uma função utilizada internamente pela função *mirtCAT*, que pode ser usada externamente para melhor usabilidade com interfaces externas. A função *mirtCAT_preamble* deve ser usada com seu argumento *final_fun*, que deve ser chamado após *shiny_GUI* ser terminado e serve para salvar as informações dos respondentes.

3.3.1.7 Função *updateDesign*

É a função que vai atualizar o objeto retornado pela função *findNextItem*. Esse argumento é utilizado na geração de padrões de respostas dos respondentes.

- a) Argumento *x*

Um objeto da classe *mirtCAT_design* que retorna da função *mirtCAT* quando se especifica *design_elements = TRUE*

- b) Argumento *items*

Vetor numérico indicando quais itens selecionar.

- c) Argumento *responses*

Vetor numérico indicando as respostas esperadas para os itens selecionados.

- d) Argumento *Theta*

Vetor indicando o valor do θ a ser colocado. O QUADRO 3 apresenta um resumo do que foi descrito anteriormente, com algumas funções em um exemplo de aplicação de TAC unidimensional de múltipla escolha com n itens utilizando o *mirtCAT* (CHALMERS, 2015a).

QUADRO 3 - DESCRIÇÃO DAS FUNÇÕES EM UM TAC.

(continua)

Função	Argumentos	Códigos R
generate.mirt_object	Parameters and itemtype	<pre>> n <- 100 > a <- seq(1,1.7,n) > b<-seq(-3,3,length.out=n) > g<-rep(0.2,n) > d<--a*b > parameters<-data.frame(a1=a, d=d, g=g) > modelo<- generate.mirt_object(parameters, + type='3PL')</pre>
mirtCAT	Df	<pre>> choice1<-rep(1,n) > choice2<-rep(2,n) > choice3<-rep(3,n) > choice4<-rep(4,n) > choice5<-rep(5,n) > choices <- apply(as.data.frame(cbind(+ choice1,choice2,choice3,choice4,choice5)), + 2,as.character) > questions<-as.character(seq(1,n)) > answer<-as.character(rep(1,n)) > df <- data.frame(Question=questions, + Answer=answer, Option=choices, Type='radio')</pre>
	Mo	> Aplicação<-mirtCAT(df=df, mo=modelo)
	Method	> Aplicação<-mirtCAT(df=df, mo=modelo, + method='EAP')
	Criteria	> Aplicação<-mirtCAT(df=df, mo=modelo, + method='EAP', criteria='MI')
	start_item	> Aplicação<-mirtCAT(df=df, mo=modelo, + method='EAP', criteria='MI',start_item='MI')
	Design	> exposição<-rep(0.3,n) <pre>> Aplicação<-mirtCAT(df=df, mo=modelo, + method='EAP', criteria='MI', + start_item='MI',design=list(min_SEM=0.2, + thetas.start=0, min_items=10, max_items=30, + exposure=exposição, classify=0.7, + classify_CI=0.95))</pre>
	shinyGUI	<pre>> temp<-paste0(getwd(), 'temp_file.rds') > title <- "Perguntas:" > authors <- "Lucas Bozza e Tiago Teixeira" > instructions<-c("Instruções:", "Para iniciar + seu teste, clique no botão abaixo", + "Próximo") > firstpage <- list(h2("TCC"), + h5("Questionário geral sobre futebol.")) > lastpage <- list(h3("Seu teste foi + finalizado com sucesso! Obrigado!")) > demographics <- list(textInput(inputId = + "Nome", label = "Insira seu nome e</pre>

QUADRO 3 - APLICAÇÃO DAS FUNÇÕES EM UM TAC.

(conclusão)

mirtCAT	shinyGUI	<pre>+ sobrenome:", value = ""), + selectInput(inputId = "Sexo", label = "Sexo", + choices = c("Masculino", "Feminino"), + selected = ""), numericInput("Idade", + "Idade", value=18 , min = 5, max = 111)) > shinyGUI <- list(title = title, authors = + authors, instructions=instructions, + demographics = demographics, + demographics_inputIDs = c("Nome", "Sexo", + "Idade"), firstpage = firstpage, lastpage = + lastpage, temp_file=temp) > Aplicação <- mirtCAT(df=df, mo=modelo, + method='EAP', criteria='MI', + start_item='MI', design=list(min_SEM=0.2, + thetas.start=0, min_items=10, max_items=30, + exposure=exposição, classify=0.7,</pre>
	preCAT	<pre>> preCAT<-list(min_items=10, max_items=30, + criteria='MI', method='EAP') > Aplicação<-mirtCAT(df=df, mo=modelo, + method='EAP', criteria='MI', + start_item='MI', design=list(min_SEM=0.2, + thetas.start=0, min_items=10, max_items=30, + exposure=exposição), shinyGUI=shinyGUI, + preCAT=preCAT)</pre>

3.3.2 Pacote *shiny* (CHANG et al., 2015)

O pacote *shiny* (CHANG et al., 2015) é uma poderosa ferramenta para criação de aplicativos que atuam como *framework* com *layouts* interativos, por meio de dois *scripts*, um para a programação do método e outro para a interface final. O pacote pode apresentar, por exemplo, saídas como: mapas e gráficos com a possibilidade de mudança de parâmetros, imagens e questionários com diversas formas de resposta (RSTUDIO, 2015).

Para utilizar o pacote *shiny* (CHANG et al., 2015) faz-se necessário a utilização de dois arquivos, um chamado de *ui.R* e outro de *server.R*. Com o uso dos dois arquivos cria-se o layout da apresentação e indicam-se as informações e análises que o julgador deseja apresentar ao respondente.

Utilizando o *shiny* (CHANG et al., 2015), será construída uma saída para a apresentação dos resultados do teste onde observam-se algumas informações sobre

os respondentes coletadas durante a aplicação dos testes e algumas análises descritivas, de acordo com o interesse do julgador.

4 RESULTADOS E DISCUSSÃO

O pacote *mirtCAT*(CHALMERS, 2015a) proporciona ao usuário uma ampla possibilidade de argumentos para a criação de diferentes testes, aqui chamados de cenários, e são apresentados alguns exemplos destes cenários e suas utilidades. Atualmente, o pacote *mirtCAT* encontra-se disponível com a versão 0.5, porém para este estudo foi utilizada a versão disponível no GitHub (CHALMERS, 2015b), versão 0.5.1.

Apresenta-se, a partir do QUADRO 4, um exemplo de códigos em R (R DEVELOPMENT CORE TEAM, 2015) para preparar os dados, gerar o modelo associado aos itens e construir o *dataframe* a ser utilizado na geração de um TAC. Independente do cenário a ser realizado, utiliza-se a mesma estrutura de códigos, apenas será adaptada a entrada dos dados, dependendo do formato em que os dados serão importados ao R e o modelo associado aos itens.

Para preparar os dados, utilizou-se um exemplo de entrada de dados por meio de um arquivo com extensão .csv que contém suas colunas com os itens, os três parâmetros associados ao modelo, as respostas dos itens e cinco opções de respostas a ser apresentadas ao respondente. Pode-se utilizar, também, dados provenientes de diferentes extensões como, por exemplo, .xls, .xlsx e txt.

Na FIGURA 3 pode-se observar a planilha e como estão dispostos os itens e seus parâmetros.

FIGURA 3 - EXEMPLO DE UMA PLANILHA COM ITENS, SEUS PARÂMETROS E OPÇÕES DE RESPOSTA

The screenshot shows an Excel spreadsheet with the following data:

Questoes	A	B	C	Respostas	Opcao 1	Opcao 2	Opcao 3	Opcao 4	Opcao 5
Questao 1	1,207	1,976	0,1015	1	1	2	3	4	5
Questao 2	1,068	-2,802	0,153	3	1	2	3	4	5
Questao 3	1,373	-1,685	0,1527	5	1	2	3	4	5
Questao 4	1,212	-2,11	0,1881	2	1	2	3	4	5
Questao 5	1,174	0,611	0,1607	4	1	2	3	4	5
Questao 6	1,333	-1,154	0,1289	2	1	2	3	4	5
Questao 7	1,135	2,694	0,1992	1	1	2	3	4	5
Questao 8	1,285	-0,672	0,1838	4	1	2	3	4	5
Questao 9	1,422	0,281	0,0844	3	1	2	3	4	5
Questao 10	1,064	-1,49	0,1173	5	1	2	3	4	5

No QUADRO 4 pode-se observar o código para importar a planilha ao R (R DEVELOPMENT CORE TEAM, 2015), e também os seis primeiros itens deste arquivo.

QUADRO 4 – DADOS DE UM TESTE APÓS IMPORTÁ-LOS AO R

```
> dados <- read.table("C:/Users/Tiago/Desktop/TCC/entradadosdados.csv",
  sep=";", dec=".", quote="\\"", h=T)
> head(dados)
Questoes      A      B      C  Respostas  Opcao.1 Opcao.2 Opcao.3 Opcao.4 Opcao.5
Questao 1  1.207  1.976  0.1015      1         1         2         3         4         5
Questao 2  1.068 -2.802  0.1530      3         1         2         3         4         5
Questao 3  1.373 -1.685  0.1527      5         1         2         3         4         5
Questao 4  1.212 -2.110  0.1881      2         1         2         3         4         5
Questao 5  1.174  0.611  0.1607      4         1         2         3         4         5
Questao 6  1.333 -1.154  0.1289      2         1         2         3         4         5
```

Para gerar o objeto do modelo, utiliza-se a função *generate.mirt_object* do pacote *mirtCAT* (CHALMERS, 2015a) com os parâmetros e o modelo associado aos itens. No QUADRO 5 pode-se observar a sequência de códigos para gerar este objeto do modelo. (d) representa o parâmetro associado à discriminação e dificuldade, utiliza-se a multiplicação entre o negativo do parâmetro discriminação (a) e o parâmetro de dificuldade (b) para parametrizar (d) na escala da distribuição normal que é utilizada na função *generate.mirt_object*. O parâmetro (g) representa a probabilidade de acerto casual.

QUADRO 5 – OBJETO DO MODELO UTILIZADO PELA FUNÇÃO *MIRT*CAT

```
> a <- dados[,2]
> b <- dados[,3]
> c <- dados[,4]
> d <- -a*b
> parametros <- data.frame(a1=a, d=d, g=c)
> objeto <- generate.mirt_object(parameters=parametros, itemtype='3PL')
```

Para gerar o *dataframe* que é utilizado pela função *mirtCAT*, pode-se observar, no QUADRO 6, a sequência de códigos R.

QUADRO 6 - PREPARAÇÃO DO TESTE

```
# Dataframe com os itens do teste, respostas, possíveis escolhas e tipo
# de item.
> Opcao.1 <- dados[,6]
> Opcao.2 <- dados[,7]
> Opcao.3 <- dados[,8]
> Opcao.4 <- dados[,9]
> Opcao.5 <- dados[,10]
> Opcoes <- apply(as.data.frame(cbind(Opcao.1,Opcao.2,Opcao.3,Opcao.4,
+ Opcao.5)),2, as.character)
> Questoes <- dados[,1]
> Respostas <- as.character(dados[,5])
> Base <- data.frame(Question=Questoes, Answer=Respostas, Option=Opcoes,
+ Type='radio')
```

Após a entrada dos dados, geração do modelo e criação do *dataframe*, inicia-se a customização do teste que será apresentado e posteriormente aplicado. A partir do item 4.1 são apresentados alguns cenários possíveis de serem aplicados e algumas discussões e interpretações.

4.1 TAC COM OS DADOS SOBRE FUTEBOL COM TAMANHO FIXO

Neste exemplo é utilizado o banco de dados de futebol, descrito na sessão 3.1.1. Como o banco de dados contém apenas 66 itens, será utilizado um tamanho fixo de itens para a estimação da habilidade, pois não se pode garantir que a estimativa da habilidade do respondente irá convergir a um critério de parada do tipo variável.

O teste tem sua estimativa de habilidade calculada através do método EAP e os itens são selecionados através do critério de *Kullback-Leibler*. Este método possui um poder de seleção maior no início dos testes pois consegue aproximar a estimativa

da habilidade do respondente ao verdadeiro valor mais rapidamente. Os itens têm sua exposição controlada pelo método de *Sympson-Hetter*, e o teste é encerrado quando o respondente responder a dez itens.

No QUADRO 7 apresenta-se a sequência de comandos para gerar um TAC com tamanho fixo no pacote *mirtCAT* (CHALMERS, 2015a) e colher as informações desejadas para a apresentação final dos resultados por meio da aplicação gerada pelo pacote *shiny* (CHANG, et al. 2015). São utilizados recursos computacionais, como por exemplo a função *for* que possibilita a geração de um ciclo para apresentar um número finito de testes e acumular seus resultados em um *dataframe* para a análise final do teste.

QUADRO 7 – TAC COM OS DADOS SOBRE FUTEBOL COM TAMANHO FIXO

(continua)

```

1. > require(mirtCAT)
2. > options(stringsAsFactors=FALSE)
3. > dados<-read.table("C:/Users/Tigo/Documents/bancodedados.csv",
4. + sep=";",dec=".",quote="\\"",h=T)
5. > #Número de respondentes do teste
6. > n <- 6
7. > #Gerando o objeto do teste
8. > a <- dados[,2]
9. > b <- dados[,3]
10. > g <- dados[,4]
11. > d <- -a*b
12. > pars <- data.frame(a1=a, d=d, g=g)
13. > mod <- generate.mirt_object(parameters=pars, itemtype='3PL')
14. > #Dataframe do teste
15. > opcao1 <- dados[,6]
16. > opcao2 <- dados[,7]
17. > opcao3 <- dados[,8]
18. > opcao4 <- dados[,9]
19. > opcao5 <- dados[,10]
20. > opcoes <- apply(as.data.frame(cbind(opcao1,opcao2,opcao3,opcao4,
21. + opcao5)),2,as.character)
22. > questões <- dados[,1]
23. > respostas <- as.character(dados[,5])
24. > base <- data.frame(Question=questoes, Answer=respostas,
25. + Option=opcoes, Type='radio')
26. > #Customização do teste
27. > title <- "Teste Adaptativo Computadorizado - Avaliação sobre
28. + Futebol"
29. > authors <- "Lucas Bozza e Tiago Teixeira"
30. > firstpage <- list(h3("TAC - Futebol"), h5("Responda as questões
31. + propostas para obter sua estimativa de conhecimento sobre
32. + futebol."))
33. > instructions <- c("Instructions:", "Clique no botão abaixo a cada
34. + passo do processo", "Next")
35. > lastpage <- function(person){
36. + nome <- as.vector(person$demographics$Nome)
37. + theta <- as.vector(person$thetas)

```

QUADRO 7 – TAC COM TAMANHO FIXO

(continuação)

```

38. + if (theta >-1){
39. + if(theta<0){
40. + return(list(h3(sprintf("Obrigado %s. Sua estimativa de habilidade
41. + é %.2f.", nome, theta)),
42. + (h3(sprintf("A interpretação para esta estimativa é de que
43. + você:"))),
44. + h4(sprintf(" *Conhece sobre símbolos e mascotes de times
45. + nacionais.")),
46. + h4(sprintf(" *Conhece sobre jogadores que fizeram parte da Seleção
47. + Brasileira.")),
48. + h4(sprintf(" *Conhece sobre títulos da Seleção Brasileira.")),
49. + h4(sprintf(" *Conhece sobre as seleções que conquistaram a Copa do
50. + Mundo.")),
51. + h4(sprintf(" *Conhece sobre técnicos brasileiros que atuaram em
52. + outras seleções nacionais.")),
53. + h4(sprintf(" *Conhece brevemente sobre as principais conquistas de
54. + times nacionais."))))}
55. + else if (theta <1){
56. + return(list(h3(sprintf("Obrigado %s. Sua estimativa de habilidade
57. + é %.2f.", nome, theta)),
58. + (h3(sprintf("A interpretação para esta estimativa é de que
59. + você:"))),
60. + h4(sprintf(" *Conhece sobre símbolos e mascotes de times
61. + nacionais.")),
62. + h4(sprintf(" *Conhece sobre jogadores que fizeram parte da Seleção
63. + Brasileira.")),
64. + h4(sprintf(" *Conhece sobre títulos da Seleção Brasileira.")),
65. + h4(sprintf(" *Conhece sobre as seleções que conquistaram a Copa do
66. + Mundo.")),
67. + h4(sprintf(" *Conhece sobre técnicos brasileiros que atuaram em
68. + outras seleções nacionais.")),
69. + h4(sprintf(" *Conhece sobre as principais conquistas de times
70. + nacionais.")),
71. + h4(sprintf(" *Conhece sobre os principais jogos da seleção
72. + brasileira.")),
73. + h4(sprintf(" *Conhece brevemente sobre times internacionais."))))}
74. + else if (theta < 2){
75. + return(list(h3(sprintf("Obrigado %s. Sua estimativa de habilidade
76. + é %.2f.", nome, theta)),
77. + (h3(sprintf("A interpretação para esta estimativa é de que
78. + você:"))),
79. + h4(sprintf(" *Conhece sobre símbolos e mascotes de times
80. + nacionais.")),
81. + h4(sprintf(" *Conhece sobre jogadores que fizeram parte da Seleção
82. + Brasileira.")),
83. + h4(sprintf(" *Conhece sobre as seleções que conquistaram a Copa do
84. + Mundo.")),
85. + h4(sprintf(" *Conhece sobre técnicos brasileiros que atuaram em
86. + outras seleções nacionais.")),
87. + h4(sprintf(" *Conhece sobre os principais jogos da seleção
88. + brasileira.")),
89. + h4(sprintf(" *Conhece brevemente sobre times internacionais.")),
90. + h4(sprintf(" *Conhece mais a fundo sobre conquistas de times e
91. + seleções nacionais."))))}
92. + else if (theta >2){
93. + return(list(h3(sprintf("Obrigado %s. Sua estimativa de habilidade
94. + é %.2f.", nome, theta)),
95. + (h3(sprintf("A interpretação para esta estimativa é de que
96. + você:"))),
97. + h4(sprintf(" *Conhece sobre símbolos e mascotes de times

```

QUADRO 7 – TAC COM TAMANHO FIXO

(conclusão)

```

98. + nacionais.')),
99. + h4(sprintf("Conhece sobre jogadores que fizeram parte da Seleção
100. + Brasileira.")),
101. + h4(sprintf("Conhece sobre as seleções que conquistaram a Copa do
102. + Mundo.")),
103. + h4(sprintf("Conhece sobre técnicos brasileiros que atuaram em
104. + outras seleções nacionais.")),
105. + h4(sprintf("Conhece sobre os principais jogos da seleção
106. + brasileira.")),
107. + h4(sprintf("Conhece brevemente sobre times internacionais.")),
108. + h4(sprintf("Conhece mais a fundo sobre conquistas de times e
109. + seleções nacionais.")),
110. + h4(sprintf("Conhece mais a fundo sobre seleções
111. + internacionais."))))))
112. + else{
113. + return(list(h3(sprintf("Obrigado %s. Sua estimativa de habilidade
114. + é %.2f.", nome, theta))),
115. + (h3(sprintf("A interpretação para esta estimativa é de que
116. + você:"))),
117. + h4(sprintf("Conhece sobre símbolos e mascotes de times
118. + nacionais.")),
119. + h4(sprintf("Conhece sobre jogadores que fizeram parte da Seleção
120. + Brasileira."))))))
121. > demographics <- list(textInput(inputId="Nome",label="Insira seu
122. + nome e sobrenome:",value = "" ),
123. + selectInput(inputId = "Sexo", label = "Sexo", choices =
124. + c("Masculino", "Feminino"), selected = "" ),
125. + numericInput("Idade", "Idade", value=18 , min = 5, max = 111))
126. > temp <- paste0(getwd(), 'temp_file.rds')
127. > shinyGUI <- list(title = title, authors = authors, instructions =
128. + instructions, demographics = demographics, demographics_inputIDs
129. + = c("Nome", "Sexo", "Idade"), firstpage = firstpage, lastpage =
130. + lastpage, temp_file=temp)
131. > exposure <- rep(0.35,66)
132. > Sexo <- vector(mode = "character", length = n)
133. > Nomes <- vector(mode = "character", length = n)
134. > Theta <- vector(mode = "numeric", length = n)
135. > SE_theta <- vector(mode = "numeric", length = n)
136. > Idade <- vector(mode = "numeric", length = n)
137. > Items <- vector(mode = "character", length = n)
138. > for (i in 1:n){
139. + resp <- mirtCAT(df=base, mo=mod, method='EAP', criteria='KL',
140. + start_item='KL', design=list (max_items=10, min_SEM=0,
141. + exposure=exposure),shinyGUI=shinyGUI)
142. + Sexo[i] <- resp$demographics$Sexo
143. + Nomes[i] <- resp$demographics$Nome
144. + Theta[i] <- resp$thetas
145. + SE_theta[i] <- resp$SE_thetas
146. + Idade[i] <- resp$demographics$Idade
147. + Items[i]<- paste(resp$items_answered, collapse=",")
148. + da <- data.frame(Nomes, Sexo, Idade, Theta, SE_theta, Items)}

```

Com tais especificações, o teste é respondido por seis respondentes, e ao final de cada teste é apresentado o resultado individual e sua interpretação.

Na FIGURA 4, pode se observar a *firstpage* gerada pelo teste.

FIGURA 4 – FIRSTPAGE DE UMA TAC COM TAMANHO FIXO



Na FIGURA 5 pode-se observar a tela onde o respondente fornece suas informações pessoais. O respondente precisa digitar seu nome e sobrenome no primeiro campo, selecionar seu sexo no segundo campo e selecionar a sua idade no último campo disponível.

FIGURA 5 – INFORMAÇÕES PESSOAIS DE UM TESTE



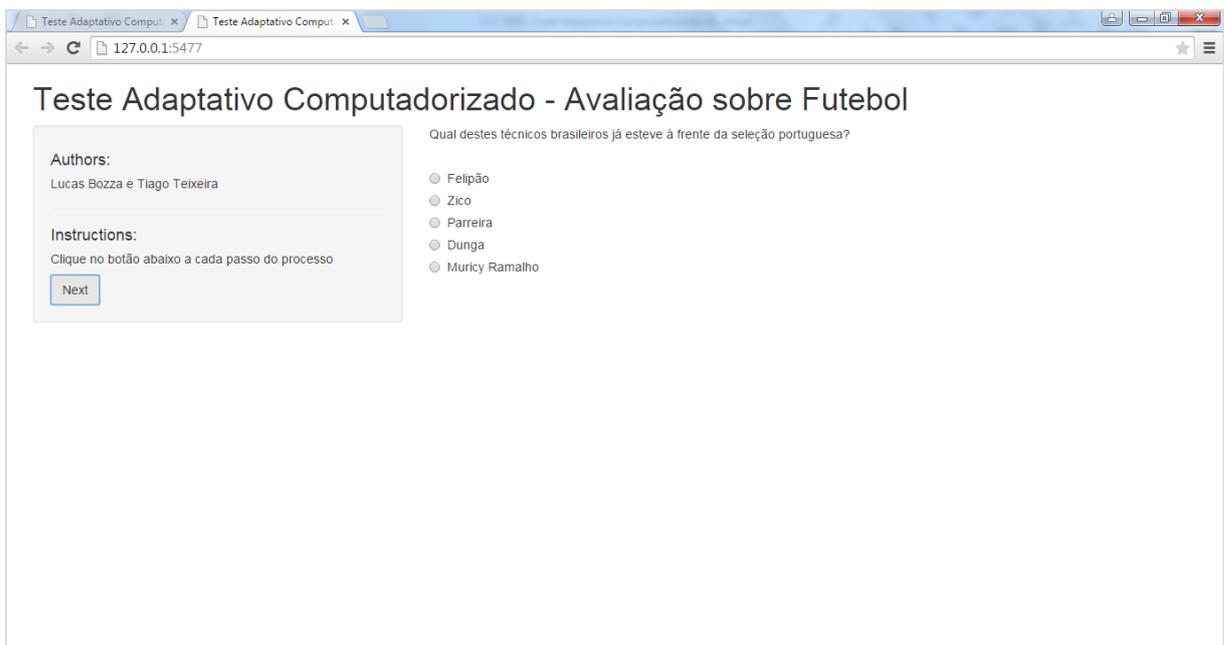
Na FIGURA 6, pode-se observar a última página apresentada ao respondente antes do teste iniciar.

FIGURA 6 – ÚLTIMA PÁGINA ANTES DO INÍCIO DE UM TESTE



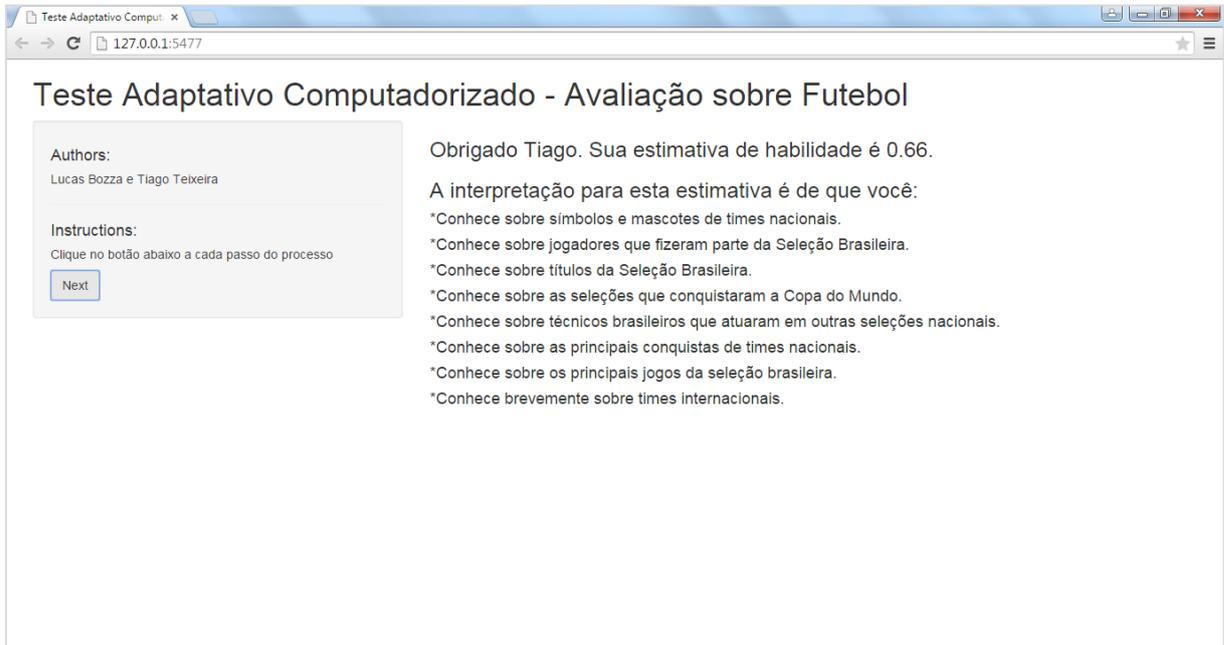
Na FIGURA 7, pode-se observar como serão apresentados os itens ao respondente. Desta maneira, o respondente precisa selecionar uma das respostas apresentadas e pressionar o botão “Next” para receber a próxima pergunta, se o critério de parada não for satisfeito.

FIGURA 7 – FORMA DE APRESENTAÇÃO DAS PERGUNTAS EM UM TESTE



Na FIGURA 8, pode-se observar a *lastpage* do teste, com o resultado individual do respondente e a interpretação de sua estimativa. Nesta tela o respondente pode observar todo o conhecimento que estima-se que ele possua no determinado assunto.

FIGURA 8 – LASTPAGE DE UM TAC COM TAMANHO FIXO



4.2 TAC COM CLASSIFICAÇÃO

Outra opção de cenário que pode vir a ser utilizada, é gerar um teste em que o interesse do julgador seja classificar os respondentes de acordo com um ou mais critérios especificados. Podem-se gerar vários tipos de classificação, como por exemplo alunos com estimativa acima de 1, alunos com estimativas entre 0 e 1, entre outros. Cada critério de classificação depende do interesse do julgador no teste aplicado.

Para gerar um TAC com um critério de classificação definido no pacote *mirtCAT* (CHALMERS, 2015a), e obter as informações desejadas para a apresentação final dos resultados, são utilizados os comandos do QUADRO 7 com as seguintes alterações:

1. Nas linhas 30 a 32, referente à *firstpage*, altera-se o código para:

```
> firstpage <- list(h3("TAC - Futebol"), h4("Avaliação Final"))
```

2. Nas linhas 35 a 120, referentes à *lastpage*, altera-se o código para:

```
> lastpage <- function(person){
+ nome <- as.vector (person$demographics$Nome)
+ theta <- as.vector(person$thetas)
+ if(theta>0){return(list(h3(sprintf("Obrigado %s. Sua
+ estimativa de habilidade é %.2f. Você passou na matéria.",
nome,
+ theta))))}
+ else { return(list(h3(sprintf("Obrigado %s. Sua estimativa de
+ habilidade é %.2f. Você não passou na matéria.", nome,
+ theta))))}}
```

3. Nas linhas 137 a 148, referentes à execução do teste, altera-se o código para:

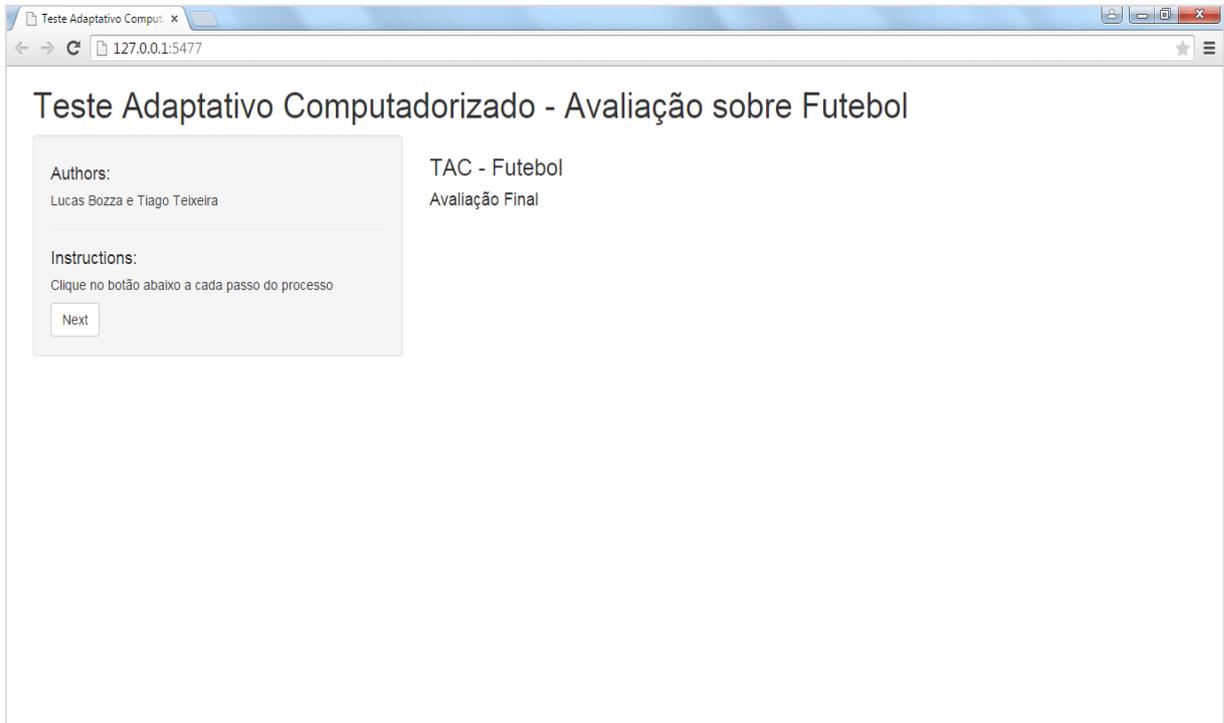
```
> Classificacao <- vector(mode = "character", length = n)
> for (i in 1:n){
+ resp <- mirtCAT(df=base, mo=mod, method='EAP', criteria='KL',
+ start_item='KL', design=list (max_items=10, min_SEM=0,
+ exposure=exposure),shinyGUI=shinyGUI)
+ Sexo[i] <- resp$demographics$Sexo
+ Nomes[i] <- resp$demographics$Nome
+ Theta[i] <- resp$thetas
+ SE_theta[i] <- resp$SE_thetas
+ Idade[i] <- resp$demographics$Idade
+ Classificacao[i]<- paste(resp$classification)
+ da <- data.frame(Nomes, Sexo, Idade, Theta, SE_theta,
+ Classificacao)}
```

Com tais especificações, o teste tem sua estimativa de habilidade calculada pelo método EAP e os itens são selecionados através do critério de *Kullbak-Leibler*. Como demonstração de um cenário classificatório, é utilizado um critério de corte da seguinte maneira: respondentes que tiverem sua estimativa abaixo de 0 estarão reprovados na matéria e alunos acima de 0 estarão aprovados. É utilizado um intervalo de confiança com 95% para gerar esta classificação. Quando a estimativa da habilidade do respondente, ao nível de 95% de confiança, estiver acima (*above cutoff*) ou abaixo (*below cutoff*) da linha de corte especificada o teste é encerrado. Se o respondente responder a todos os itens do banco de dados e sua estimativa, ao nível de 95% de confiança, não tiver acima ou abaixo da linha de corte, o teste é encerrado e o respondente é classificado como sem decisão (*no decision*).

Com tais códigos R, este teste classificatório tem um *layout* de teste com algumas diferenças em relação ao cenário anterior, conforme figuras a seguir.

Na FIGURA 9, pode-se observar a *firstpage* gerada pelo teste, com apenas as alterações dos textos apresentados.

FIGURA 9 – *FIRSTPAGE* DE UM TAC COM CLASSIFICAÇÃO



Nas FIGURAS 10 e 11, pode-se observar a *lastpage* gerada pelo teste e apresentada ao respondente com sua estimativa de habilidade e sua interpretação, que neste caso é de que se o respondente está aprovado ou reprovado.

FIGURA 10 – *LASTPAGE* DE UM TAC COM CLASSIFICAÇÃO COM O RESPONDENTE APROVADO

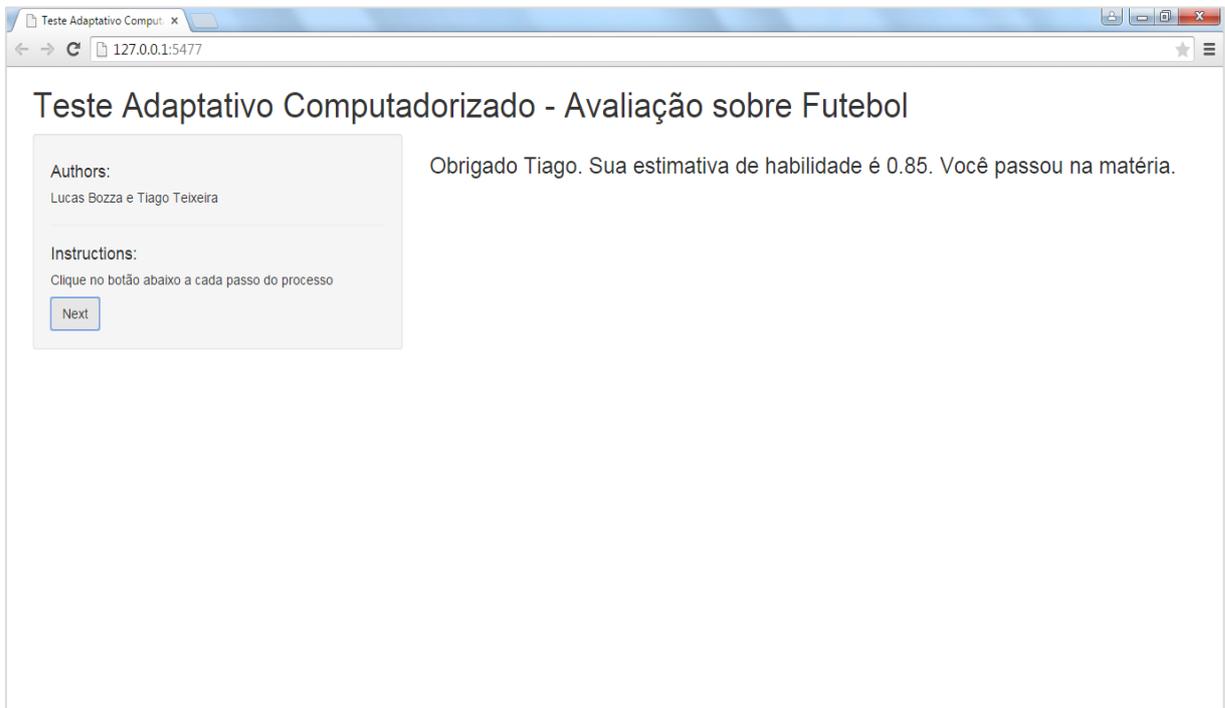


FIGURA 11 – *LASTPAGE* DE UM TAC COM CLASSIFICAÇÃO COM O RESPONDENTE REPROVADO



4.3 TESTES NÃO ADAPTATIVOS

Com o pacote *mirtCAT*, também é possível aplicar testes não adaptativos computadorizados aos respondentes de duas maneiras distintas. A primeira é com o respondente respondendo a todos os itens de forma sequencial e encerrando seu teste após responder a última questão do banco de itens, como um teste tradicional, e a outra maneira possível de se aplicar um teste não adaptativo é apresentar seus itens de forma também sequencial, porém seu teste irá encerrar depois que algum critério de parada do teste for satisfeito. Não é possível fazer testes não adaptativos computadorizados selecionando seus itens de forma aleatória no pacote *mirtCAT* (CHALMERS, 2015a).

Para gerar um teste não adaptativo sem um critério de parada, ou seja, um teste tradicional, com o respondente precisando responder à todas as perguntas, são utilizados os códigos do QUADRO 7 com as seguintes alterações:

1. Nas linhas 30 a 32, referentes à *firstpage*, altera-se o código para:

```
> firstpage <- list(h3("Teste Completo sobre Futebol"),
+ h5("Responda as questões propostas para obter sua estimativa
de
+ conhecimento sobre futebol mundial."))
```

2. A linha 131, referente à *exposure*, deve ser excluída, pois não se aplica a esse cenário.
3. Nas linhas 139 a 141, referentes à função *mirtCAT*, altera-se o código para:

```
+ resp <-mirtCAT(df=base, mo=mod, method = 'EAP',
criteria='seq',
+ design = list(min_SEM=0),shinyGUI=shinyGUI)
```

Com tais especificações o teste é apresentado de forma tradicional. A estimativa de habilidade do respondente é calculada pelo método EAP, os itens são apresentados de forma sequencial, do primeiro ao último, na ordem em que foram colocados no *dataframe* “base”, que é o que compõe o argumento “*df*”. Ao final de cada teste, o respondente recebe a estimativa de sua habilidade junto com sua interpretação.

Para gerar um teste não adaptativo com um critério de parada especificado, são utilizados os códigos do QUADRO 7 com as seguintes alterações:

1. Nas linhas 30 a 32, referente à *firstpage*, altera-se o código para:

```
> firstpage <- list(h3("Teste Completo sobre Futebol"),
+ h5("Responda as questões propostas para obter sua estimativa
de
+ conhecimento sobre futebol mundial."))
```

2. A linha 131, referente à *exposure*, deve ser excluída, pois não se aplica a esse cenário.
3. Nas linhas 139 a 141, referentes à função *mirtCAT* (CHALMERS, 2015a), altera-se o código para:

```
+ resp <-mirtCAT(df=base, mo=mod, method = 'EAP',
criteria='seq',
+ design = list(max_items = 20),shinyGUI=shinyGUI)
```

Com tais especificações o teste também não é adaptativo. A estimativa de habilidade é calculada pelo método EAP, os itens são apresentados de forma sequencial, porém este teste é encerrado quando os 20 primeiros itens forem respondidos ou o erro padrão da habilidade estimada for menor que 0,3, o que ocorrer primeiro. Este critério de parada relacionado com o erro padrão da estimativa da habilidade do respondente é padrão da função *mirtCAT* e funciona sempre que não seja definido nenhum valor para o argumento *min_SEM*. Ao final de cada teste, também é apresentada a estimativa da habilidade individual e sua interpretação.

Nas FIGURAS 12 e 13, pode-se observar, respectivamente, a *firstpage* e a *lastpage* para os testes não adaptativos computadorizados utilizando o banco de dados futebol. As outras telas apresentadas ao respondente seguem o mesmo modelo utilizada nos cenários anteriores.

FIGURA 12 – FIRSTPAGE DE TESTES NÃO ADAPTATIVOS



FIGURA 13 – LASTPAGE DE TESTES NÃO ADAPTATIVOS



4.4 TAC COM BANCO DE ITENS SIMULADO

Foi simulado pelos autores um banco de itens de tamanho 1000 para exemplificar o processo de estimação da habilidade do respondente em diferentes pares de critério de seleção dos itens e métodos de estimação da habilidade do respondente no pacote *mirtCAT* (CHALMERS, 2015a).

Abaixo, apresenta-se a sequência de códigos em R (R DEVELOPMENT CORE TEAM, 2015) que são utilizados para gerar os parâmetros e o modelo de todos os testes adaptativos computadorizados simulados nesta exemplificação, além da quantidade de simulações geradas. As linhas de códigos 1, 2 e 3, representam respectivamente os parâmetros de discriminação (a), dificuldade (b) e acerto casual (c) dos itens. A linha 4, representa o parâmetro d associado aos parâmetros a e b parametrizado na escala da função normal. Nas linhas 5 e 6, pode-se observar a geração do modelo associado aos itens. A linha 7 representa a quantidade de simulações geradas para cada cenário desenvolvido. Neste exemplo foi utilizado, também, o modelo logístico unidimensional de três parâmetros (3PL).

```

1. > a <- rnorm(1000,1.35,0.11) #simulação de 1000 números com
   média igual a 1.35 e desvio padrão igual a 0.11.
2. > b <- rnorm(1000, 0,1) #simulação de 1000 números com média
   igual a 0 e desvio padrão igual a 1.
3. > g <- rep(0.2,1000) #1000 números iguais a 0.2
4. > d <- -a*b
5. > pars <- data.frame(a1=a, d=d, g=g)
6. > mod <- generate.mirt_object(pars, '3PL')
7. > n <- 100 #número de simulações igual a 100

```

A seguir, são simulados alguns dos possíveis cenários do pacote *mirtCAT* (CHALMERS, 2015a) e em cada caso são apresentados a média das habilidades estimadas e o número médio de itens utilizados no processo de estimação dos respondentes. Para obtenção das médias, cada um dos cenários foi simulado 100 vezes pois com mais simulações demanda muito tempo computacional.

Na TABELA 1, estão apresentadas as estimativas da habilidade média e o número médio de itens respondidos em simulações onde o critério de parada do teste

utilizado foi o erro padrão da habilidade estimada for menor que 0,2 ou quando todo o banco de itens for apresentado. O argumento *Theta* utilizado para gerar o padrão de respostas é igual a zero, ou seja, os respondentes possuem habilidade média igual a 0. Os métodos de cálculo da estimativa de habilidade são o EAP, MAP ou ML, e os itens são selecionados através do critério MI, KL ou MEI. Foram geradas 100 simulações para cada uma das nove combinações entre um método e um critério. Os códigos utilizados para realizar essas simulações são:

```

1. > Theta <- vector(mode = "numeric", length = n)
2. > I_Answered <- vector(mode = "numeric", length = n)
3. > for (i in 1:n){
4. + pat <- generate_pattern(mo = mod, Theta = 0)
5. + simulado <- mirtCAT ( mo=mod, method = 'MET_ESCOLHIDO',
6. + criteria='CRIT_ESCOLHIDO', local_pattern =
7. + pat, design = list(min_SEM = 0.2))
8. + Theta[i] <- simulado$thetas
9. + I_Answered[i] <- length(simulado$items_answered)
10. + df<-data.frame(mean(Theta),mean(I_Answered)) }

```

Observa-se que devem ser alterados os valores atribuídos aos argumentos *criteria* e *method*, respectivamente CRIT_ESCOLHIDO e MET_ESCOLHIDO, por valores válidos.

TABELA 1 – HABILIDADES ESTIMADAS E ITENS RESPONDIDOS EM TESTES SIMULADOS COM DIFERENTES ARGUMENTOS *CRITERIA* E *METHOD*.

Valores atribuídos respectivamente aos argumentos <i>criteria</i> e <i>method</i> :	Habilidade média:	Itens respondidos – Média e desvio padrão
EAP e MI	0,0044	67,00 ; 0,1881
EAP e KL	0,0268	67,03 ; 0,1723
EAP e MEI	0,0126	69,09 ; 0,1885
MAP e MI	0,0172	64,68 ; 0,1830
MAP e KL	-0,0053	64,29 ; 0,1863
MAP e MEI	0,0205	66,26 ; 0,2074
ML e MI	0,0056	68,57 ; 0,1949
ML e KL	0,0112	68,30 ; 0,2013
ML e MEI	0,0127	69,38 ; 0,1969

Na TABELA 2, observam-se diferentes simulações com os padrões de resposta para os valores no argumento *Theta* iguais a -2, -1, 0, 1 e 2, em testes onde supõe-se a *priori* que a habilidade do respondente seja igual a zero. Com isso, os primeiros itens são apresentados com nível de dificuldade próximo ou igual a zero. As diferenças nos códigos para essas simulações em relação às anteriores ocorrem na linha 4, onde o *Theta* recebe diferentes valores, na linha 5, onde o argumento *method* é sempre EAP, e na linha 6, onde o argumento *criteria* é sempre MI. Estes argumentos *method* e *critéria* foram fixados apenas para exemplificar os cenários pois são os usualmente mais utilizados.

TABELA 2 – HABILIDADES ESTIMADAS E ITENS RESPONDIDOS EM TESTES SIMULADOS COM DIFERENTES ARGUMENTOS *THETA*.

Valor atribuído ao argumento <i>Theta</i>	Habilidade média estimada	Itens respondidos – Média e desvio padrão
-2	-1,9258	104,67 ; 0,1986
-1	-0,9633	72,36 ; 0,2096
0	0,0205	66,91 ; 0,1852
1	0,9662	70,91 ; 0,1955
2	1,9176	82,06 ; 0,2102

Observa-se que, para esses dados simulados, quanto maior a diferença entre o valor verdadeiro da habilidade do respondente e o valor estabelecido a *priori*, maior o tamanho do teste para um mesmo erro padrão. Por isso, muitos testes consideram a habilidade inicial como a média a *priori* da população ou tentam obter alguma informação sobre o respondente, utilizando um pré TAC, por exemplo.

Com tais resultados, pode-se gerar uma análise para identificar se existem diferenças significativas para esses pares de combinações de métodos de estimativa da habilidade do respondente e critérios de seleção dos itens, para poder avaliar como cada combinação se adapta a diferentes tipos de banco de itens, como, por exemplo, de tamanho pequeno, médio ou grande, ou, também, que contenham vários itens para apenas alguns níveis de habilidade, entre outras possibilidades que podem ocorrer em situações do cotidiano.

4.5 APRESENTAÇÃO DOS RESULTADOS

Após todos os testes, de um determinado cenário, serem finalizados, é gerada uma aplicação do pacote *shiny* (CHANG, et al., 2015) para a apresentação dos resultados finais do teste. Pode-se adicionar análises descritivas e gráficas à aplicação de acordo com o interesse do julgador no teste proposto. A aplicação é controlada de forma interativa pelo julgador, podendo, o mesmo, selecionar apenas os resultados individuais de cada respondente, também pode-se selecionar os resultados conforme o sexo ou idade dos respondentes.

No QUADRO 8, pode-se observar um exemplo de código para gerar uma aplicação no pacote *shiny*, sendo necessária a criação de dois arquivos R (R DEVELOPMENT, 2015), chamados de *ui.R* e *server.R*. O pacote *shiny* é utilizado em sua versão 0.12.0 (CHANG, et al., 2015).

QUADRO 8 – CÓDIGOS PARA GERAR A APLICAÇÃO EM *SHINY*

(continua)

```
>#ui.R
>media<-mean(da$Theta)
>sexo <- tapply(da$Theta,da$Sexo,mean)
>sexof <- sexo[1]
>sexom <- sexo[2]
>shinyUI(fluidPage(
+titlePanel("Resultados Finais"),
+fluidRow(column(4,selectInput("Nomes","Nome:",c("Todos",
+unique(as.character(da$Nomes))))),
+column(4,selectInput("Sexo","Sexo:",c("Todos",unique(as.character(da$Sexo)
+o))))),column(4,selectInput("Idade","Idade:",c("Todos",
+unique(as.character(da$Idade)))))),
+fluidRow(column(12,dataTableOutput(outputId="table"),h4(sprintf("A
+estimativa de habilidade média do teste foi %.6s",media)),h5(sprintf("A
+estimativa de habilidade média do sexo masculino foi
+%.6s",sexom)),h5(sprintf("A estimativa de habilidade média do sexo
+feminino foi %.6s",sexof)))),
+fluidRow(column(6,plotOutput("graf")),column(6,plotOutput("geral")))))

> #server.R
>shinyServer(function(input, output){
+individuo <- reactive({input$Nomes})
+output$table <- renderDataTable({
+data <- da
+if (input$Nomes != "Todos"){data <- data[data$Nome == input$Nomes,]}
+if (input$Sexo != "Todos"){data <- data[data$Sexo == input$Sexo,]}
+if (input$Idade != "Todos"){data <- data[data$Idade == input$Idade]}
+data})
+output$graf <- renderPlot({
```

QUADRO 8 – CÓDIGOS PARA GERAR A APLICAÇÃO EM SHINY

(continuação)

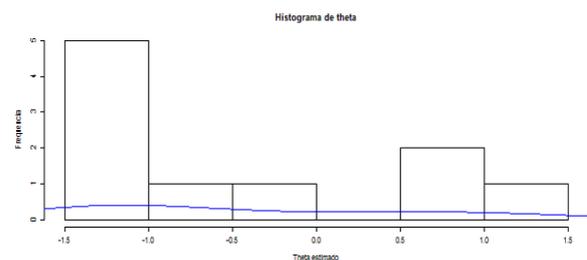
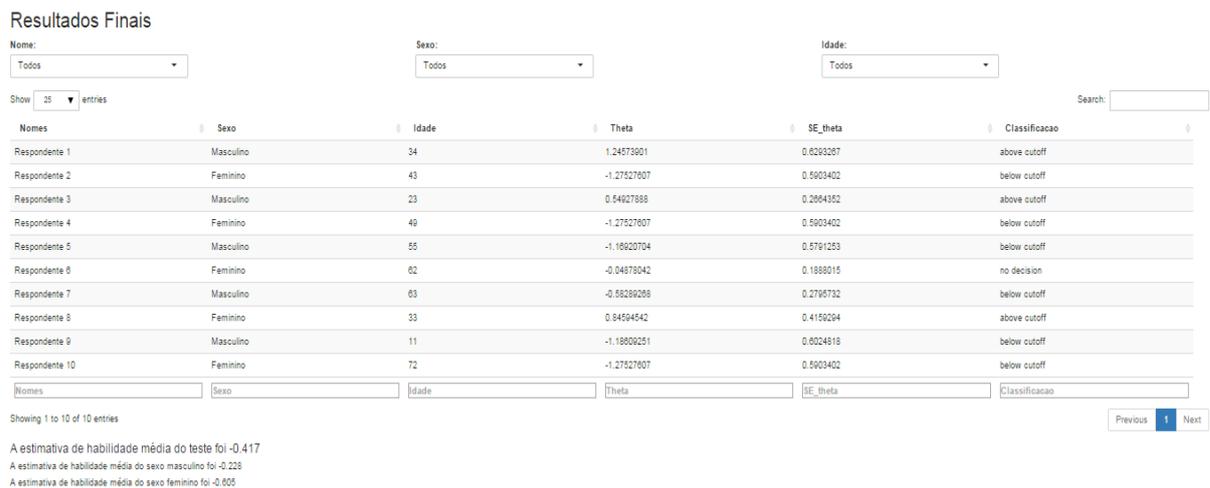
```

+if(input$Nomes == "Todos"){plot(0, 0, type = "n", axes = F, xlab = "",
+ylab="")} else {plot(Thetas_history[[input$Nomes]], type = "o", xlab =
+"Passos da estimativa", ylab = "Habilidade estimada")}}
+output$geral <- renderPlot({hist(Theta, freq = TRUE, xlab = "Theta
+estimado", ylab = "Frequencia", main = "Histograma de theta")
+lines(density(Theta), col=4)}))

```

Na FIGURA 14, observa-se a aplicação que é apresentada ao final de todos os testes, por exemplo do cenário de classificação, com os resultados de todos os respondentes, com algumas análises descritivas sobre o teste e um histograma da estimativa de habilidade do teste com uma linha de tendência.

FIGURA 14 – APRESENTAÇÃO DOS RESULTADOS DO TESTE



Nas FIGURAS 15 16 e 17, observa-se a aplicação que é apresentada ao final de todos os testes, quando é selecionado apenas os resultados de apenas um respondente. São observadas algumas informações descritivas sobre o teste, um histograma do teste com uma linha de tendência e um gráfico com a estimativa da habilidade do respondente durante o processo do TAC. Pode-se observar ao final da tabela se o responde teve sua estimativa próxima ao ponto de corte, nesse caso o

respondente é classificado como sem decisão (*no decision*), abaixo do ponto de corte (*below cutoff*) ou acima do ponto de corte (*above cutoff*).

FIGURA 15 – APRESENTAÇÃO DOS RESULTADOS DO TESTE COM UM RESPONDENTE CLASSIFICADO COMO *NO DECISION*

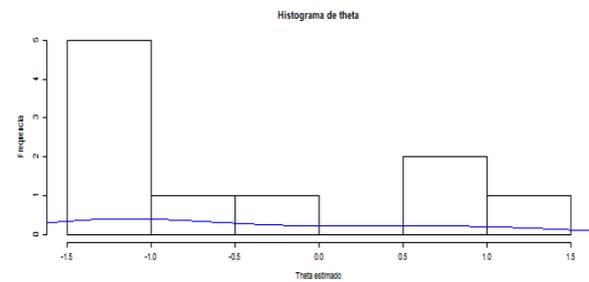
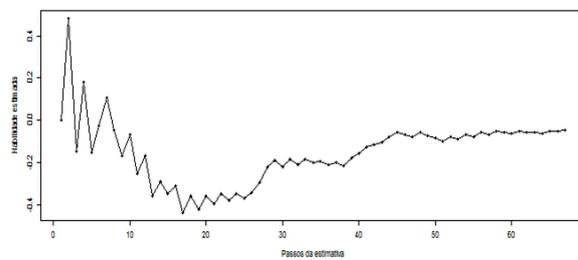
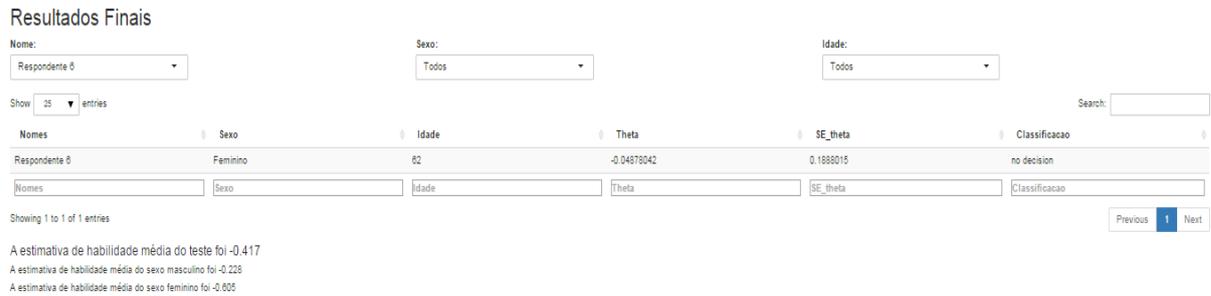


FIGURA 16 – APRESENTAÇÃO DOS RESULTADOS DO TESTE COM UM RESPONDENTE CLASSIFICADO COMO *BELOW CUTOFF*

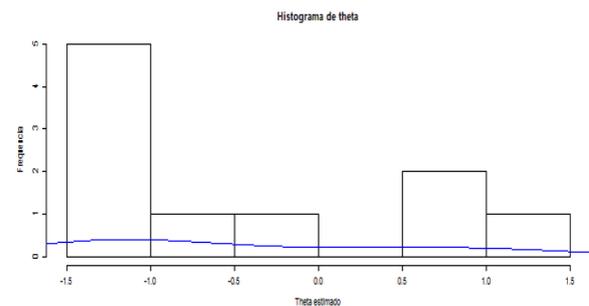
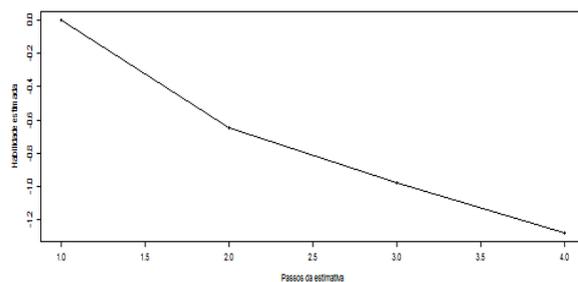
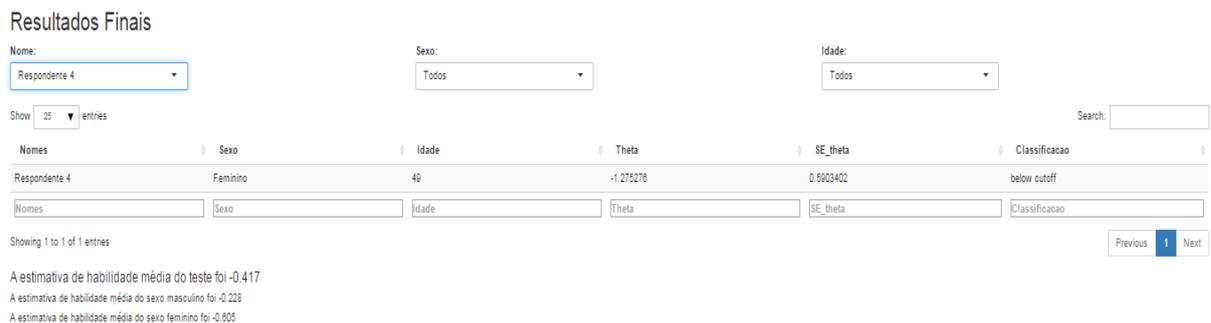


FIGURA 17 – APRESENTAÇÃO DOS RESULTADOS DO TESTE COM UM RESPONDENTE CLASSIFICADO COMO *ABOVE CUTOFF*

Resultados Finais

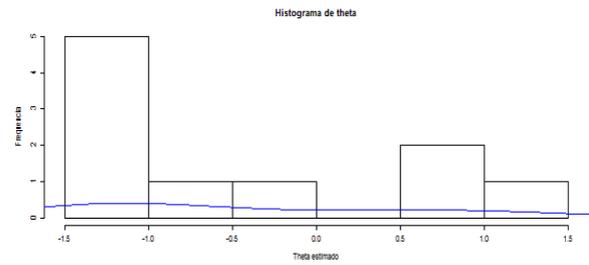
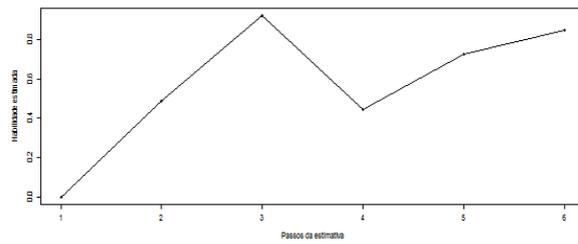
Nome: Sexo: Idade:

Show entries Search:

Nomes	Sexo	Idade	Theta	SE_theta	Classificacao
Respondente 8	Feminino	33	0.8459454	0.4159294	above cutoff

Showing 1 to 1 of 1 entries Previous 1 Next

A estimativa de habilidade média do teste foi -0.417
 A estimativa de habilidade média do sexo masculino foi -0.228
 A estimativa de habilidade média do sexo feminino foi -0.605



5 CONSIDERAÇÕES FINAIS

Ainda no fim de 2014, tínhamos o desejo de desenvolver um estudo sobre um assunto que não tínhamos tido contato durante a graduação para que pudéssemos ter um ganho de conhecimento em alguma área da estatística que ainda não havíamos explorado. Foi nesse momento que nos foi apresentado o TAC, que até então era algo que não tínhamos nenhum conhecimento, e isso nos motivou a encarar esse desafio durante esse trabalho de conclusão de curso.

No início do trabalho, tínhamos o desafio de buscar entender todo funcionamento de um TAC, a teoria por traz do processo, suas aplicações e possibilidades para poder colocar em prática todo nosso esforço e dedicação.

Durante a execução do trabalho esbarramos em algumas limitações do pacote *mirtCAT* (CHALMERS, 2015a), pois por ser um pacote recente e em contínuo desenvolvimento ainda não abrange todas as possibilidades para diferentes aplicações, o que acabava limitando nosso trabalho e alterando nossas ideias no momento. Tivemos algumas trocas de e-mails com o desenvolvedor do pacote *mirtCAT* para saber as possibilidades de alterações rápidas no pacote e sempre com respostas rápidas enviadas pelo mesmo, conseguimos desenvolver com êxito nosso propósito para o trabalho, de demonstrar diferentes aplicações de um TAC e apresentar uma aplicação em *shiny* (CHANG et al., 2015) com os resultados dos testes.

Como sugestão para futuros estudos, podem-se aplicar TACs no ambiente de *web* por meio dos pacotes *mirtCAT* (CHALMERS, 2015a) e *shiny* (CHANG et al., 2015), pois recentemente ficaram disponíveis algumas funções no pacote *mirtCAT* que possibilitam a conexão entre o servidor local e o servidor *web*, o que torna o pacote mais completo e atrativo. Pode-se gerar um estudo simulado para comparar diferentes combinações entre os métodos de cálculo de estimativa da habilidade dos respondentes e os critérios de seleção dos itens e buscar uma combinação que melhor se adapte aos itens. Por fim, sugerimos a aplicação de testes multidimensionais por meio do pacote *mirtCAT*.

REFERÊNCIAS

ANDRADE, D. F.; TAVARES, H. R.; VALLE, R. C. **Teoria da resposta ao item: conceitos e aplicações**. São Paulo: Associação Brasileira de Estatística, 2000.

CHALMERS, P. **MirtCAT: Computerized Adaptive Testing with Multidimensional Item Response Theory**. R package version 0.5. Disponível em: <<http://CRAN.R-project.org/package=MirtCAT>>. Acesso em: 09 mai. 2015a.

_____. **MirtCAT: Computerized Adaptive Testing with Multidimensional Item Response Theory**. GitHub repository. Disponível em: <<https://github.com/philchalmers/mirtCAT>>. Acesso em: 16 mai. 2015b.

CHANG, C.; CHENG, J.; ALLAIRE, J. J.; XIE, Y.; MCPHERSON, J. **Shiny: Web Application Framework for R**. R package version 0.12.0. Disponível em: <<http://CRAN.Rproject.org/package=Shiny>>. Acesso em: 18 mai. 2015.

COSTA, D. R. **Métodos Estatísticos em Testes Adaptativos Informatizados**. 2009, 120 f. Dissertação (Mestrado) – Departamento de Métodos Estatísticos, Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009.

FORBELLONE, A. A.; INÁCIO, L. V. **Construção e Validação de um Instrumento de Mensuração para Avaliar o Conhecimento sobre Futebol Utilizando a Teoria da Resposta ao Item**. 2013. Projeto apresentado à disciplina de Teorias de Avaliação, Curso de Bacharelado em Estatística, Universidade Federal do Paraná, Curitiba, 2013.

GONÇALVES, P. J. **Desafios e Perspectivas da Implementação Computacional de Testes Adaptativos Multidimensionais para Avaliações Educacionais**. 2013, 177 f. Tese (Doutorado em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2013.

LABARRÈRE, J. G.; DA-SILVA, C. Q.; COSTA, D. R. Testes Adaptativos Computadorizados. **Revista Brasileira de Biometria**, São Paulo, v. 29, n. 2, p. 229-261, abr.-jun. 2011.

MOREIRA JUNIOR, F. J. **Sistemática para a implantação de Testes Adaptativos Informatizados baseados na Teoria da Resposta ao Item**. 2011. 334 f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em

Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2011.

MOREIRA JUNIOR, F. J. et al. Processo de um teste adaptativo informatizado com base na teoria da resposta ao item para a estimação da usabilidade de sites de e-commerce. **Produção**, São Paulo, v. 23, n. 3, p. 525-536, jul./set. 2013.

OLIVEIRA FILHO, A. A.; FONSECA, M. A. **Análise de Itens e da Confiabilidade de um Teste de Avaliação de Conhecimentos - Um Estudo de Caso**. 2006. 62 f. Trabalho de Conclusão de Curso (Graduação em Estatística) – Curso de Bacharelado em Estatística, Universidade Federal do Paraná, Curitiba, 2006.

R DEVELOPMENT CORE TEAM. **R: A Language and Environment for Statistical Computing**. R Foundation for Statistical Computing. Vienna, Austria. ISBN: 3-900051-07-0. Disponível em: <<http://www.R-project.org/>>. Acesso em: 03 fev. 2015.

RSTUDIO. **RStudio: Integrated development environment for R (Versão 0.98.1102) [Computer software]**. Boston, MA. Disponível em: <<http://www.rstudio.com/>>. Acesso em: 01 mar. 2015.

RICARTE, T. A. M. **Teste Adaptativo Computadorizado nas Avaliações Educacionais e Psicológicas**. 2013. 67 f. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2013.

SANTOS, M. L. **Desempenho Escolar em Pernambuco: Análise dos Itens e das Habilidades usando Teoria Clássica e TRI**. 2008, 89 f. Dissertação (Mestrado em Estatística) – Departamento de Estatística, Universidade Federal de Pernambuco, Recife, 2008.

WAINER, H. TACs: Whither and whence. **Psicológica**, Espanha, v. 21, n. 1, p. 121-133, 2000.