Software Review

Journal of Educational and Behavioral Statistics
2017, Vol. 42, No. 5, pp. 617–638
DOI: 10.3102/1076998617716462

Reporting Subscores Using R: A Software Review

Shenghai Dai
Dubravka Svetina
Xiaolin Wang
Indiana University Bloomington

There is an increasing interest in reporting test subscores for diagnostic purposes. In this article, we review nine popular R packages (subscore, mirt, TAM, sirt, CDM, NPCD, lavaan, sem, and OpenMX) that are capable of implementing subscore-reporting methods within one or more frameworks including classical test theory, multidimensional item response theory, cognitive diagnostic models, and factor analysis. A real data example is used to illustrate how to examine whether subscores should be reported and how to obtain subscores. We also briefly compare the features of selected packages for reporting subscores.

Keywords: *subscore reporting; added value; psychometric quality; R packages*

Motivated by the call for diagnostic assessment information, there has been an increasing interest in reporting formative test subscores to identify individual students' academic strengths and weaknesses in specific learning domains (Sinharay, Puhan, & Haberman, 2011). Accompanied by the recent reforms in education, the professional community has posited requirements for subscore reporting. Specifically, the No Child Left Behind Act of 2001 (2002, p. 6) requires the assessment scores to be interpretive, descriptive, and diagnostic for individuals and also to "include information regarding achievement on the academic assessments measured against the state's student academic achievement standards, which will help parents, teachers, and principals to understand and address the specific academic needs of students." Similarly, the National Research Council report, *Knowing What Students Know* (Pellegrino, Chudowsky, & Glaser, 2001), stated that the assessment should provide particular formative and diagnostic information about individual examinee's learning and achievement.

Given this call for subscores, they should be reported within a context in which they have added value, are reliable, and are distinct from other subscores (Sinharay, 2010; Tate, 2004). While calling for diagnostic information, current and previous editions of the *Standards for Educational and Psychological Testing* (American Educational Research Association, American Psychological Association, & National Council on Measurement in Education, 2014) emphasize that should an assessment result in multiple scores, the (sub)scores should have adequate psychometric quality in order to be reported. Similarly, Sinharay (2010) and other scholars emphasized that the decision of reporting subscores should be made based on evidence that such subscores are psychometrically sound and have added value.

In this article, we provide a review of nine statistical packages which could be used for subscore reporting in the R software (R Core Team, 2016)—subscore, mirt, TAM, sirt, CDM, NPCD, lavaan, sem, and OpenMX. We begin by briefly introducing popular subscoring methods within four psychometric modeling frameworks, which are classical test theory (CTT), item response theory (IRT), cognitive diagnostic models (CDMs), and factor analysis (FA). We then turn our attention to illustrate how to obtain subscore estimates using the R packages. Data from the 2011 administration of Trends in International Mathematics and Science Study (TIMSS), eighth-grade mathematics assessments are used to illustrate subscore reporting using the aforementioned R packages. Lastly, we conclude the review by comparing package features and estimated subscores obtained within different frameworks.

Subscore-Reporting Methods

Over the past two decades, methods within the frameworks of CTT, IRT, CDMs, and FA have been used to report subscores (Sinharay, Haberman, & Puhan, 2007). Within the CTT framework, popular subscore-reporting methods are based on Haberman's (2008) work and include methods where subscore is estimated as (1) a function of the observed subscore, (2) a function of the observed total score, and (3) a function of both the observed subscore and the observed total score.[1] Besides CTT, application of multidimensional item response theory (MIRT) models, CDMs, and FA in subscore reporting were also mentioned and investigated in literature (Haberman & Sinharay, 2010; Sinharay et al., 2007, 2011; Yao & Boughton, 2007).

In addition to estimating subscores, procedures in aforementioned frameworks are available to examine psychometric qualities of the subscores. In CTT, the proportional reduction in mean squared error (PRMSE; Haberman, 2008) is usually calculated to determine whether subscores are of added values and worth being reported. In the remaining three frameworks, understanding whether subscores have added value is typically accomplished by evaluating the overall data–model fit statistics and/or comparing multi- and single-factor models by examining relative fit indices.

In the literature, researchers have examined the use of the above methods in reporting subscores (e.g., Haberman & Sinharay, 2010; Sinharay et al., 2007). While some scholars suggested that MIRT is superior in terms of accuracy (e.g., Dwyer, Boughton, Yao, Steffen, & Lewis, 2006; Haberman & Sinharay, 2010; Yao & Boughton, 2007), we argue that the selection of a subscore-reporting

TABLE 1.
Comparison of the Four Subscoring Frameworks

Framework	Complex Structure	Exploratory and/or Confirmatory	Scale of Subscale Variables	Scale of Item Variables
CTT	No	Confirmatory	Continuous	Discrete
IRT	Yes	Both	Continuous	Discrete
CDM	Yes	Confirmatory	Dichotomous	Discrete
FA	Yes	Both	Continuous	Continuous

CTT = classical test theory; IRT = item response theory; CDM = cognitive diagnostic model; FA= factor analysis.

method should be determined by the estimation accuracy, as well as other factors, in practice. These factors include but are not limited to the type of the analysis (exploratory or confirmatory), the structure of subscales (simple[2] or complex structure), and the nature (observed or latent) and scale (discrete or continuous) of both subscale and item variables. For example, CTT, MIRT, and FA typically assume continuous subscore variables, while the CDMs often express the subscore at a categorical/binary level. The conceptualization of the subscore differs among the frameworks as well. In CDMs, for example, subscores are conceptualized as the cognitive knowledge structures in the hypothesized Q-matrix—a loading structure that indicates which attributes or skills are required for correct responses to specific items (Roberts & Gierl, 2010). The attribute estimates are discrete scores that represent examinees mastery to each of the attributes (usually "1" represents mastery of a certain attribute while "0" represents nonmastery). Table 1 presents a summary of the methods and their appropriateness of use based on these factors. In the table, we can see that the CDM will be the appropriate method if a dichotomous latent variable is defined for each subscale in a test. Readers interested in this topic should see Sinharay et al. for a broad overview of general subscore-reporting methods.

A Real Data Example

In this section, we review selected R packages that are available to report subscores within the four frameworks mentioned above using a real data example. Specifically, we focus our attention on demonstrating how to use these packages to (1) decide whether the subscores should be reported and (2) obtain subscore estimates when they are of adequate psychometric quality.

The Data

The eighth-grade mathematics data from the 2011 TIMSS assessment were used in this study. TIMSS assesses achievement in countries around the world every 4 years and collects a rich array of information about the educational contexts for learning mathematics and science (Mullis, Martin, Gonzalez, & Chrostowski, 2004). It is one of the most popular large-scale assessments, and its public-use data are available for free. As described in the TIMSS 2011 Mathematics Framework, four content subscales were defined to specify the mathematics subject matter at the eighth grade. These four subscales were (1) number, (2) algebra, (3) geometry, and (4) data and chance. Due to the booklet design in TIMSS, students participating in the assessment were only administered a fraction (i.e., a booklet) of the total item pool. In this study, data from Booklet 11 that consists of two blocks (M11 and M12) were utilized. It contained responses from 765 students to 32 items with 6 to 9 items on each subscale. Omitted responses were treated as incorrect in this study.

Working With R

CTT-based methods. The R package subscore (Dai, Wang, & Svetina, 2016) is capable of obtaining subscore estimates using Haberman's regression-based methods. The main function that conducts the CTT-based subscore reporting procedures is `CTTsub(test.data, method="Haberman")`, where `method="Haberman"` indicates that Haberman's methods are used and `test.data` is a list that contains scored data on all subtests and on the entire test. A nice feature of the subscore package is that it has a function `data.prep(scored.data, subtest.infor)` that can generate such a data list. In our case, subscores based on Haberman's methods can be obtained by executing the following commands:

```
#Load the package.
>library(subscore)

#Specify the subtest information: four subscales of the TIMSS data with each
#one consists of 9, 9, 6 and 8 items, respectively. In the output all
#subscales are labeled as 'Subscore.1' to 'Subscore.4.'
>test.infor <- c(4,9,9,6,8)

#Read original scored item response data file, 'TIMSSdata.csv', and
#create a data frame object, 'data'
#The data file 'TIMSSdata.csv' is a comma delimited file that includes 765
#rows(i.e., examinees) and 32 columns (i.e., test items).
>data <- read.csv("TIMSSdata.csv", header=T)

#Generate the data list for the main function.
>test.data <- data.prep (data, test.infor)

#Run the main function to obtain Haberman's subscore reporting methods.
>Subscores <- CTTsub(test.data, method="Haberman")
>names(Subscores)
[1] "summary"          "PRMSE"            "subscore.original" "subscore.s"
  "subscore.x"       "subscore.sx"
```

```
#Obtain Results/Outputs – Original subscores.
>Subscores$subscore.Original
```

```
#Obtain Results/Outputs - estimated subscores.
#Only subscore.sx are reported here according to values of added.value.s and
#added value.sx. subscore.s and subscore.x are also available if they are of
#added value.
>Subscores <-Subscores$subscore.sx
```

	Subscore.sx.1	Subscore.sx.2	Subscore.sx.3	Subscore.sx.4
1	7.73	5.35	3.72	7.20
2	5.11	3.94	1.98	4.57
…	…	…	…	…

```
#Obtain Results/Outputs - summary statistics.
#Statistics reported mean and standard deviation of original subscore (Orig.mean &
#Orig.sd),subscore_s (s.mean & s.sd), subscore_sx (x.mean & x.sd) and
#subscore_sx (sx.mean & sx.sd)
>Subscores$summary
```

	Orig.mean	Orig.sd	s.mean	s.sd	x.mean	x.sd	sx.mean	sx.sd
Subscore.1	5.57	2.24	5.57	1.52	5.57	1.70	5.57	1.70
Subscore.2	3.22	2.09	3.22	1.41	3.22	1.45	3.22	1.51
Subscore.3	2.17	1.53	2.17	0.87	2.17	1.02	2.17	1.03
Subscore.4	5.34	2.05	5.34	1.39	5.34	1.50	5.34	1.52

```
#Obtain Results/Outputs – PRMSE.
#Including reliability (Cronbach Alpha) of original subscales and the total
#test, PRMSE values for each method that can be used to decide whether
#subscores should be reported, and subscore reporting recommendations
#added.value.s and added.value.sx)where '1' indicates that subscores are of added
#value and '0' vice versa.
>Subscores$PRMSE
```

	Orig. reliability	PRMSE.s	PRMSE.x	PRMSE.sx	added. value.s	added. value.sx
Subscore.1	0.68	0.68	0.84	0.85	0	0
Subscore.2	0.67	0.67	0.72	0.77	0	1
Subscore.3	0.57	0.57	0.77	0.79	0	0
Subscore.4	0.68	0.68	0.78	0.81	0	1
Total.test	0.87	/	/	/	/	/

As indicated in the output provided above, a list of six objects (i.e., "summary," "PRMSE," "subscore.original," "subscore.s," "subscore.x," and "subscore.sx") are generated and stored in the outcome object `Subscores` when the execution is finished, allowing users to accomplish subscore reporting tasks. Using the subscore package, all subscores, including original subscores, subscore$_s$, subscore$_x$, and subscore$_{sx,}$ are estimated at the same time when executing the main function. In our case, subscore$_{sx}$ can be obtained by running `Subscores$subscore.sx.` Summary statistics subscores such as mean and standard deviation can be obtained by executing `Subscores$summary.`

Before estimated subscores can be reported, one should examine whether they are of added value. Per Haberman (2008), this can be accomplished by examining the values of the PRMSEs. Specifically, the rules suggest (1) report subscore$_s$ when PRMSEs (PRMSE corresponding to subscore$_s$) is greater than PRMSE$_x$

(PRMSE corresponding to subscore$_x$) and (2) report subscore$_{sx}$ when PRMSE$_{sx}$ (PRMSE corresponding to subscore$_{sx}$) is substantively greater than the maximum of the other two PRMSE values (i.e., PRMSE$_x$ and PRMSE$_{sx}$). To decide what is "substantively greater," Haberman and Sinharay (2013) further suggested that PRMSE$_{sx}$ ought to reduce the distance of PRMSE$_s$ or PRMSE$_x$, whichever is larger, from 1.0 by at least 10% for augmented subscores to have added values. That is to say, to report augmented subscores, the following condition should be met:

$$\text{PRMSE}_{sx} - (\text{PRMSE}_s \text{ or } \text{PRMSE}_x) > 0.1 \times (1 - (\text{PRMSE}_s \text{ or } \text{PRMSE}_x)).$$

The above criteria are imbedded in the subscore package, and subscore reporting recommendations are provided along with reported PRMSE values (i.e., added.value.s and added.value.sx in the output). The above results indicated that none of the PRMSE$_s$ values were larger than their corresponding PRMSE$_x$, as indicated by all values under column `added.value.s` being "0." Therefore, the subscores were of no added value. Then, by following Haberman and Sinharay's (2013) rule, calculations revealed that augmented subscores of the algebra subscale and the data and chance subscale were of added value (i.e., `added.value.sx` values were "1"), while those of number and geometry were not (i.e., `added.value.sx` values were "0").

In summary, the subscore package allows users to obtain estimated subscores (simple structure only) using Haberman's three regression-based methods (i.e., subscore$_s$, subscore$_x$, and subscore$_{sx}$) within the framework of CTT (see Table 2). In addition to providing PRMSE values that are used to examine the psychometric quality of subscores, it automatically gives recommendations on whether the subscores should be reported by applying Haberman and Sinharay's (2013) criteria.

IRT-based methods. Within the IRT framework, subscores come in the form of latent ability (or theta) estimates that are obtained from a specific MIRT model. Packages such as mirt (Chalmers, 2017), TAM (Kiefer, Robistzsch, & Wu, 2017), and sirt (Robitzsch, 2017) are available for users to obtain subscores within the MIRT framework. mirt can fit a variety of general compensatory MIRT models including the Rasch model, the 2- to 4-parameter logistic (PL) models, and the graded response model with both dichotomous and polytomous item response. TAM can be used to fit compensatory multidimensional Rasch models as well as some other IRT models (e.g., 2- to 3-PL, multifacet models) with both dichotomous and polytomous response data. The sirt package complements the other two R packages in IRT modeling. It can fit noncompensatory, compensatory, and partially compensatory MIRT models but only with dichotomous item responses. Another difference among the three packages lies in the estimation methods. mirt supports multiple estimation methods including the default expectation–maximization (EM) algorithm, quasi-Monte Carlo EM

TABLE 2.
R Packages for Examining Subscore Psychometric Quality and Reporting Subscores

Subscore Framework	R Packages	Main Function	Complex Subscale Structure	Polytomous Item Response	Estimated Subscores	Reporting Recommendation	Supported Methods/ Models	Estimation Methods
CTT	Subscore	CTTsub	No	Yes	True subscores	Yes	Haberman's methods	Regression
IRT	mirt	mirt	Yes	Yes	Theta estimates	No	General compensatory multidimensional IRT (MIRT) models	Expectation Maximization (EM), quasi-Monte Carlo EM, Metropolis–Hastings Robbins–Monro, andBock and Lieberman
	TAM	tam.mml, tam.jml	Yes	Yes	Theta estimates	No	Compensatory multidimensional Rasch, 2-3PL, and so on	Marginal maximum likelihood (MML), joint maximum likelihood
	sirt	smirt	Yes	No	Theta estimates	No	Noncompensatory, compensatory and partially compensatory MIRT models, and so on	MML
CDM	CDM	gdina	Yes	Yes	Attribute profiles	No	DINA, DINO, RRUM, MC-DINA, ACDM, LCDM, GDM, GDINA, pGDINA, MG-GDINA	Weighted least squares (WLS), unweighted least squares (ULS), and maximum likelihood (ML)

(continued)

TABLE 2. *(continued)*

Subscore Framework	R Packages	Main Function	Complex Subscale Structure	Polytomous Item Response	Estimated Subscores	Reporting Recommendation	Supported Methods/ Models	Estimation Methods
	NPCD	JMLE, ParMLE	Yes	No	Attribute profiles	No	DINA, DINO, NIDA, GNIDA, RRUM	Joint maximum likelihood estimation (nonparametric), ML
FA	lavaan	cfa	Yes	Yes	Factor scores	No	Confirmatory FA (CFA)	ML, full information ML (FIML), general least squares (GLS), WLS, ULS, diagonally weighted least squares, and extensions of ML (e.g., WLMV)
	sem	sem	Yes	Yes	Factor scores	No	CFA	FIML, ML, GLS
	OpenMX	mxRun	Yes	Yes	Factor scores	No	CFA	FIML, ML, WLS

DINA = the deterministic input, noisy-and-gate model; DINO = the deterministic input, noisy-or-gate model; RRUM = the reduced reparametrized unified model; MC-DINA = the multiple choice DINA; ACDM = the additive CDM; LCDM = the log-linear CDM; GDM = the generalized diagnostic model; GDINA = the generalized DINA; pGDINA = polytomous attributes GDINA; MG-GDINA = multiple group GDINA; NIDA = deterministic input, noisy-or-gate (NIDA) model; GNIDA = the generalized NIDA.

estimation, the Metropolis–Hastings Robbins–Monro estimation, and the Bock and Lieberman approach. TAM supports both marginal maximum likelihood (MML) and joint maximum likelihood (JML) estimators in which JML is for Rasch models only, while sirt uses only MML employing the EM algorithm as its estimation method.

To report subscores or theta estimates within the MIRT framework, the main function `mirt(data, model, ...)` in the mirt package allows users to fit a selected MIRT model by specifying estimation methods as well as other features (e.g., number of quadrature points per dimension). Similarly, MIRT modeling can be achieved by functions `tam.mml(resp, Q,...)` and `tam.jml(resp, Q,...)`, depending on the selection of estimation methods (MML or JML) in TAM and the main function `smirt(dat, Qmatrix,...)` in sirt. All three packages require a data frame of examinees' responses and a specified dimension loading structure as the basic input and at the same time allow specifications that are more advanced. Decisions on which package to be used can be made by taking into account the selection of the MIRT model, the estimation method, and characteristics of data (e.g., polytomous item responses). For the TIMSS example, the package mirt was chosen for the reasons that (1) TIMSS data usually contain some polytomous items responses and (2) there are four dimensions/subscales specified for the test. As suggested by Chalmers (2017), the EM algorithm is effective when there are three or fewer dimensions modeled. However, quasi–Monte Carlo EM or Metropolis–Hastings Robbins–Monro estimation methods should be considered when a model contains more than three dimensions. The primary tutorial for this package is provided in Chalmers (2012). Specifically, the MIRT modeling procedure for the TIMSS example can be executed via the following commands:

```
#Load the package.
>library(mirt)

#Specify the dimensional structure of the test.
#In this step, a model syntax object dim.structure is defined to specify
#the four-dimensional MIRT model, as eluded above.
>dim.structure <- 'number = 1-9
                   algebra = 10-18
                   geometry = 19-24
                   data.chance = 25-32'

#Build specified model for the main function.
#The mirt.model() function then scans the specified dimension structure and
#transfers it to a model object model that can be executed by the main function.
>model <- mirt.model(dim.structure)

#When both the data set and the specified model object are ready,
#the MIRT analysis can be conducted using the following commands:
```

```
#Fit the four-factor 3PL mirt model using the main function.
#(1)"data" term represents examinees' responses in matrix format of data frame;
#(2)"model" is the model object defined in the preview step.
#In this example, a 3PL MIRT model is fit given the presence of multiple choice
#item responses. The QMCEM approach (method="QMCEM") was selected given that the
#number of dimensions exceeded three.
>mirt.result <- mirt (data, model, itemtype= "3PL", method="QMCEM")
#Obtain theta estimates (subscores) via MAP estimation.
>theta.estimates<-fscores(mirt.result, full.scores=T, method='MAP',
                          scores.only=TRUE, QMC=TRUE)
            F1         F2         F3         F4
     1    0.193015   0.957919   1.202631   0.472394
     2    -0.22609   1.379777   -0.04208   -0.97232
     …       …          …          …          …

#Obtain model fit indices: Log-likelihood, AIC, and BIC.
>mirt.result
Log-likelihood = -13402.17
AIC = 26996.34; AICc = 27024.22
BIC = 27441.77; SABIC = 27136.93

#Obtain global model-data fit indices, including M2, a popular fit index
#used in MIRT, and other fit indices (e.g., TLI, CFI) that are
#based on fitting the null model.
#quasi-Monte Carlo (QMC) integration should be used for high-dimensional models.
>M2(mirt.result, QMC=TRUE)

>M2(mirt.result, QMC = T)
            M2  df p RMSEA RMSEA_5 RMSEA_95 SRMSR   TLI   CFI
stats 2298.433 432 0 0.075   0.072    0.078 0.159 0.837 0.858

#Obtain item fit indices.
>itemfit(mirt.result, QMC=TRUE)
            item     S_X2    df.S_X2       p.S_X2
     1        Q1    29.88         20         0.07
     …         …        …          …            …
    32       Q32    35.13         21         0.03

#Obtain person fit indices.
>personfit(mirt.result, QMC=TRUE, method="EAP",
           Theta=NULL, stats.only=TRUE)
              Zh
     1     -0.13
     …         …
   765      0.85
```

As stated previously, subscores within the MIRT framework are estimated in the form of theta estimates on each dimension or subscale. After all commands are executed, the function `fscores (mirt.result,...)` in the mirt package allows users to estimate subscores using five different methods: expected a posteriori (EAP), maximum a posteriori (MAP), weighted likelihood estimation (WLE), maximum likelihood (ML), and expected a posteriori for sum scores (EAPsum). In the TIMSS example, as indicated in the output provided above, MIRT subscores for all four subscales using the MAP method are obtained through this function (see highlighted commands above). The same function (i.e., `fscores`) can also be used to compute subscores for a model that is estimated using the sirt package. When TAM is used, subscores are computed using WLE, MLE, or EAP via the main function (e.g., tam.mml or tam.jml).

After all commands are executed, users ought to examine the psychometric quality of the subscores before they can be reported. The mirt package (and the other two packages) does not automatically give recommendations on whether subscores should be reported. However, decisions can be made by evaluating the model–data fit statistics that are provided by executing corresponding functions in the package. As indicated in the commands and outputs listed above, the mirt package provides functions, assessing the model fit in a variety of ways including the global model fit, item fit, and person fit. The `M2(mirt.result)` function gives the global model fit indices including M_2 statistics (Maydeu-Olivares & Joe, 2006) as well as other popular fit statistics (e.g., root mean square error of approximation [RMSEA] < .05 indicates good fit, Tucker–Lewis index [TLI] < .90 indicates acceptable fit, and comparative fit index [CFI] < .90 indicates acceptable fit). The `itemfit(mirt.result)` function assesses the goodness-of-fit indices at the item level by using the S-χ^2 test statistics (Kang & Chen, 2007; Orlando & Thissen, 2000, 2003). The `personfit(mirt.result)` function computes Z_h statistics (Drasgow, Levine, & Williams, 1985; values bigger than 0 indicates a better fit than expected while less than 0 vice versa) that can be used to assess the person fit. Decisions on whether subscore reporting is meaningful within the MIRT framework should be made by taking into account multiple model fit statistics, especially when some indices disagree with others. A MIRT model is of adequate psychometric quality when all or most global model fit statistics (e.g., M_2, RMSEA, TLI, CFI, etc.) are at least at acceptable levels and most of item and person estimates also had acceptable fit. In the TIMSS example, results returned by the `M2()` function suggested that the four-dimensional model did not fit the data very well, with M_2 being 2,298.43 (df = 432, p < .001) and RMSEA, TLI, and CFI being .08, .84, and .86, respectively. The S-χ^2 statistics given by the `itemfit()` function showed that 14 of the 32 items might not fit the model very well with p values less than .05. The person fit indices provided by the `personfit()` function suggested that 191 of 765 examinees were flagged as not fitting very well as suggested by the negative Z_h values. In conclusion, the MIRT model lacked adequate psychometric quality, and MIRT subscores should not be reported.

Another way to decide whether subscores should be reported using the mirt package is to fit a unidimensional IRT model in addition to the four-dimensional model and compare the model fit indices (e.g., log likelihood, Akaike information criterion, and Bayesian information criterion) between the two models.

Results showed that the unidimensional model fitted the data a bit better than the four-dimensional model as suggested by the higher TLI (.94) and CFI (.94) indices and the lower AIC (26187.99) and BIC (26633.41) values, which indicated that the MIRT subscores might lack enough psychometric qualities to be reported.

A summary of the three packages that are capable of reporting subscores within the IRT framework is provided in Table 2. They share similar basic input requirements (i.e., item response data and dimension loading structure) to fit a MIRT model and allow for more advanced specifications. As shown in Table 2, the selection of a specific package should be based on the model to be used, the appropriate estimation method, the item response format, and other specifications. MIRT subscores can be obtained in the form of θ or ability estimates via corresponding functions. None of the packages provide recommendations on whether subscores should be reported. However, users can make the decision by examining the model fit indices and comparing model fit statistics between the MIRT model and the unidimensional IRT model.

CDM-based methods. Subscores in the CDM framework come in the form of attribute profiles (Sinharay, 2010). Such profiles generally consist of 1s or 0s that indicate whether an examinee has mastered the required attributes or not. To fit a CDM and obtain attribute profile estimates in R, currently there are two packages available: CDM (Robitzsch, Kiefer, George, & Uenlue, 2017) and NPCD (Zheng & Chiu, 2016). The CDM package allows users to fit a variety of CDMs including the deterministic input, noisy-and-gate (DINA) model (Junker & Sijtsma, 2001), the deterministic input, noisy-or-gate (DINO) model (Templin & Henson, 2006), and the reduced reparameterized unified model (RRUM; Hartz, 2002; see Table 2 for a list of CDMs that are available in the CDM package). The NPCD package can fit the DINA model, the DINO model, the RRUM model, the deterministic input, noisy-or-gate (NIDA) model (Templin & Henson, 2006), and the Generalized NIDA (GNIDA) model. In addition to the models that can be fitted, there are two major differences between the two packages (see Table 2). First, CDM uses parametric estimation methods, such as the EM algorithm, while NPCD uses the joint maximum likelihood estimation (JMLE), a hybrid method combing the nonparametric classification method and the maximum likelihood estimation method (Chiu, Köhn, Zheng, & Henson, 2015). Second, CDM works with both dichotomous and polytomous item responses while NPCD only works with dichotomous data.

The main function `gdina(data, q.matrix, rule,...)` in the CDM package allows users to fit a selected CDM by specifying the item response data, the Q-matrix (see Table 3 for the Q-matrix of the TIMSS example), and the CDM to be used as the basic input. Other specifications that are more advanced such as the estimation method are also available in this function. In NPCD, the CDM

TABLE 3.
Q-Matrix of the TIMSS Data

	Subscales/Attributes			
Items	Number	Algebra	Geometry	Data and Chance
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0
7	1	0	0	0
8	1	0	0	0
9	1	0	0	0
10	0	1	0	0
11	0	1	0	0
12	0	1	0	0
13	0	1	0	0
14	0	1	0	0
15	0	1	0	0
16	0	1	0	0
17	0	1	0	0
18	0	1	0	0
19	0	0	1	0
20	0	0	1	0
21	0	0	1	0
22	0	0	1	0
23	0	0	1	0
24	0	0	1	0
25	0	0	0	1
26	0	0	0	1
27	0	0	0	1
28	0	0	0	1
29	0	0	0	1
30	0	0	0	1
31	0	0	0	1
32	0	0	0	1

modeling procedure is achieved by the main function `JMLE(Y,Q,model,...)` by specifying the data, the Q-matrix, and the model.

To illustrate the application of CDMs using the TIMSS data example, the CDM package was selected due to the presence of polytomous item responses on the assessment. Ravand and Robitzsch (2015) described in detail the steps of fitting a CDM and how do conduct CDM analysis using this package. By

```
#Load the package.
>library(CDM)

#Read Q-matrix.
>Q.matrix<-read.csv("Qmatrix.TIMSS.csv", header = F)

#Fit RRUM.
>RRUM.result<-gdina(data, q.matrix=Q.matrix, rule="RRUM")

#Obtain subscores (i.e., estimated attribute profiles) for each examinee
>Attribute.profiles <- RRUM.result$pattern

 mle.est  mle.post  map.est  map.post  post.attr1  post.attr2  post.attr3  post.attr4
   1111     0.98     1111     1.00       1.00        1.00        1.00        1.00
   0100     0.53     0100     0.73       0.00        1.00        0.27        0.01
     …        …        …        …          …           …           …           …

#Model fit - this function gives both relative and absolute global
#model fit statistics.
>RRUM.model.fit<-IRT.modelfit(RRUM.result)
#Absolute fit indices
>RRUM.model.fit$statlist
    maxX2 p_maxX2 MADcor SRMSR 100*MADRESIDCOV MADQ3 MADaQ3
1 212.505       0  0.074 0.099           1.508 0.053  0.048

#Relative fit indices
>RRUM.model.fit$IRT.IC
  loglike  Deviance Npars   Nobs       AIC       BIC      AIC3      AICc      CAIC
-13321.54  26643.08 75.00 765.00  26793.08  27141.07  26868.08  26809.62  27216.07

#Item fit
>itemfit.sx2(RRUM.result)
       item   itemindex   S-X2    df      p  S-X2_df  RMSEA  Nscgr  Npars  p.holm
    1  Q1             1  28.29    23   0.21     1.23   0.02     25      2    1.00
    2  Q2             2  24.51    23   0.38     1.07   0.01     25      2    1.00
    …  …              …      …     …      …        …      …      …      …       …
   32  Q32           32  35.38    23   0.05     1.54   0.03     25      2    1.00

#Person fit
>data <- RRUM.result$data
>probs <- RRUM.result$pjk
>skillclassprobs <- RRUM.result$attribute.patt[,1]
>RRUM.personfit<-personfit.appropriateness(data , probs , skillclassprobs)
>summary(RRUM.personfit)
                        appr.type M.rho SD.rho median.SE.rho prop.sign.T2
Spuriously High Scorers         1 0.128  0.209         0.183        0.139
Spuriously Low Scorers          0 0.134  0.208         0.162        0.157
```

choosing RRUM as the CDM for the TIMSS data, CDM analysis can be achieved by running the following commands in R:

When all the commands are executed, examinees' discrete attribute profiles can be obtained as subscores. Attribute profile estimates are usually obtained by applying a cutoff point (e.g., .50) to their estimated posterior probabilities on each attribute (de la Torre, 2009). Both packages CDM and NPCD provide attribute profile estimates simultaneously when executing the main functions.

Specifically, CDM computes attribute profiles using both MLE and MAP and returns results in the output object `pattern` (see the highlighted commands above), and NPCD computes attribute profiles using JMLE and returns results as the object `alpha.est`.

Similar to the other frameworks, an important step before reporting CDM subscores is to determine whether they are of adequate psychometric quality. In implementation of CDMs, the Q-matrix is usually prespecified and assumed theoretically defensible. However, it is still necessary to check the psychometric quality of the subscore estimates through examining the goodness of fit of the selected CDMs. Similar to the MIRT framework, within CDMs, neither CDM or NPCD provides recommendations on whether subscores should be reported based on CDMs. Users are expected to make the decisions by collecting evidence from model fit statistics. The CDM package provides a variety of global model fit, item fit, and person fit statistics by executing corresponding functions. Both relative and absolute model fit statistics can be obtained by the `IRT.modelfit()` function. Specifically, the relative fit indices include log likelihood, AIC, BIC, AIC3, AIC with a correction for finite sample sizes (AICc), and consistent AIC (CAIC). Absolute model fit statistics include max χ^2 (Chen, de la Torre, & Zhang, 2013) and its associated p value. It also contains several effect size measures of the absolute model fit, such as mean of absolute deviations in observed and expected correlations (MADcor). The function `itemfit.sx2()` gives both S-χ^2 test statistics and RMSEA to assess item fit. The person fit statistics are provided by the function `personfit.appropriateness()` by conducting likelihood ratio tests for appropriateness statistics (ρ; Liu, Douglas, & Henson, 2009). The appropriateness index is the probability of an examinee responding aberrantly beyond his or her true ability. It examines two types of aberrant behaviors: (1) Type 1 appropriateness (or spuriously high scorers) flags examinees who do not master required skills but report correct response on many items and (2) Type 0 appropriateness (or spuriously low scorers) flags examinees who have mastered all required skills but failed to respond correctly to expected number of items. Compared to the CDM package that produces a number of different fit indices, the NPCD package reports only model and item fit statistics. Specifically, it assesses the model fit using AIC and BIC through the function `ModelFit()`, and examines item fit using RMSEA and the Q_1 χ^2 statistic (Wang, Shu, Shang, & Xu, 2015) through the function `ItemFit()`.

For the TIMSS example, results of the max χ^2 test ($\chi^2 = 212.505$, $p <$ .001) indicated that the RRUM did not fit the data very well. In terms of the item fit (i.e., per S-χ^{22} test), 11 of the 32 items were flagged as poor fitting with p values less than .05. RMSEA values, however, suggested acceptable item fit, ranging from .00 to .05. Person fit, evaluated via statistic of appropriates, suggested that 13% and 16% of examinees were flagged as having spuriously high and low scores, respectively. Based on the above discussion, we could conclude that we should be cautious about reporting subscores for this example within the CDM framework.

In summary, both packages CDM and NPCD are capable of reporting subscores within the framework of CDMs. As indicated in Table 2, major differences between the two packages lie in the CDMs and estimation methods that are available, and the item response format (dichotomous vs. polytomous). We suggest readers use the CDM package for general subscore reporting purposes for its support for a larger number of models and both binary and polytomous item responses. The NPCD package is recommended when users want to apply the NIDA model or to implement the nonparametric estimation method. Subscores can be obtained in the form of attribute profiles within the framework of CDMs. Neither of the packages give recommendations automatically on whether subscores should be reported. Decisions should be made by taking into account model fit indices provided by the package.

FA-based methods. Both exploratory factor analysis (EFA) and confirmatory factor analysis (CFA) are available within the FA frameworks. In this review, only CFA is considered. Within the framework of CFA, subscores are estimated as factor scores for each factor or subscale. Popular packages lavaan (Rosseel, 2017), sem (Fox, Nie, & Byrnes, 2016), and OpenMX (Boker, Neale, & Maes, 2017) allow users to conduct CFA and obtain factor scores. sem package uses ML, full information ML (FIML), or general least squares (GLS) estimation to fit SEM models, while OpenMX allows for more computational complex modeling using ML, FIML, and weighted least squares (WLS) estimation. lavaan can fit a variety of models as well while supporting more estimator options compared to the other two, including ML, FIML, GLS, WLS, unweighted least squares, and multiple extensions of the ML estimation such as ML with robust mean and variance (WLMV). The lavaan package is more user-friendly in conducting SEM analysis, especially for those who are R novices but are familiar with other commercial SEM tools, such as Mplus (Muthén & Muthén, 2015). The `mplus2lavaan()` function allows users to convert the Mplus syntax directly into the lavaan syntax. In addition, the `mimic()` argument allows users to generate the output that looks like those from commercial packages (e.g., M*plus*).

CFA analysis is achieved through the function `cfa()` in the lavaan package. In packages sem and OpenMx, it is executed using functions `sem()` and `mxRun()`, respectively. All three packages allow users to fit a CFA model with a prespecified model syntax, but with different syntax language styles. The lavaan package uses its own syntax language known as the lavaan syntax (Rosseel, 2012). Both sem and OpenMX adopt the reticular action model (RAM) language (a popular modeling language in SEM to specify models using matrices or paths) to model specifications. The commands and outputs below

```
#Load the lavaan package
>library(lavaan)

#Model specification – this part specifies the four-factor model to be fit
#according to the test information and place it to the model object
#four.factor.model.
>four.factor.model<-"number=~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9
                     algebra=~ Q10+Q11+Q12+Q13+Q14+Q15+Q16+Q17+Q18
                     geometry=~ Q19+Q20+Q21+Q22+Q23+Q24
                     data.chance=~ Q25+Q26+Q27+Q28+Q29+Q30+Q31+Q32"

#Fit the four factor model and generate results in the format of Mplus output.
#In this process, estimation methods can be specified using the argument
#estimator. The ML method is used by default.
>fit.four.factor<-cfa(four.factor.model, data=data, mimic="mplus")

#Estimate factor scores.
>factor.scores<-predict(fit.four.factor)
          number    algebra   geometry  data.chance
  1       0.079445  0.158131  0.368674  0.25428
  2       -0.07407  0.194593  -0.02844  -0.13876
  …       …         …         …         …
#Obtain the Mplus style output
#The summary() command in the last line gives the output of the model
#estimation. Model fit statistics are included by specifying
#fit.measures=T.
>summary(fit.four.factor,fit.measures=T)

lavaan (0.5-23.1097) converged normally after 158 iterations

  Number of observations                           765
  Number of missing patterns                         1

  Estimator                                         ML
  Minimum Function Test Statistic             1360.415
  Degrees of freedom                               458
  P-value (Chi-square)                           0.000

Model test baseline model:
  Minimum Function Test Statistic             5196.301
  Degrees of freedom                               496
  P-value                                        0.000

User model versus baseline model:
  Comparative Fit Index (CFI)                    0.808
  Tucker-Lewis Index (TLI)                       0.792

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)             -13304.265
  Loglikelihood unrestricted model (H1)     -12624.057

  Number of free parameters                        102
  Akaike (AIC)                               26812.530
  Bayesian (BIC)                             27285.798
  Sample-size adjusted Bayesian (BIC)        26961.903

Root Mean Square Error of Approximation:
  RMSEA                                          0.051
  90 Percent Confidence Interval          0.048  0.054
  P-value RMSEA <= 0.05                          0.341

Standardized Root Mean Square Residual:
  SRMR                                           0.065

Parameter Estimates:
  Information                                 Observed
  Standard Errors                             Standard
```

demonstrate the CFA analysis for the TIMSS example using the lavaan package.

After all commands are executed, subscores or factor scores on each subscale can be obtained through corresponding functions: `predict()` in lavaan, `fscores()` in sem, and `mxFactorScores()` in OpenMx. Assuming the subscores in the TIMSS example were of sufficient psychometric quality and thus would be reported, the function `predict(fit.four.factor)` can be used to compute factor scores using estimated parameters, as indicated by the last two lines of the commands.

Similar to IRT- and CDM-based packages, none of the three CFA-based packages gives recommendations on whether subscores should be reported. Here again, users will need to examine the psychometric quality of the subscores by looking into available model fit statistics. As shown in the output above, the lavaan package provides a variety of model–data fit statistics such as the χ^2 test, TLI, CFI, RMSEA, AIC, and BIC. Results of the χ^2 test showed that the four-factor model did not fit the data well ($\chi^2 = 1{,}360.42$, $df = 458$, $p < .001$). Other fit indices suggested poor data–model fit as well including TLI = .79, CFI = .81, and RMSEA = .051. In addition to the four-factor model, we fitted a single-factor model and compared the incremental fit. Results showed that the single-factor model fit the data worse than the four-factor model with lower TLI (.71) and CFI (.73) indices and higher RMSEA (.06), AIC (27,166.33), and BIC (27,611.76) values. In conclusion, although the four-factor fit the TIMSS data better than did the one-factor model, both the χ^2 test and the model fit indices suggested it lacked adequate psychometric quality. Therefore, there is no evidence to support the subscore reporting for the TIMSS example. Similar model fit statistics are available when using the other two packages, too.

In summary, all three packages are capable of reporting subscores within the framework of FA. As indicated in Table 2, selection among the three packages for a CFA analysis lies in the availability of different estimation methods, and users' preference of model specification languages and familiarity of the output formats. For example, users may choose sem or OpenMx if they are more familiar to RAM. Subscores can be obtained in the form of factor scores using either of the packages. None of the packages provides recommendations automatically on whether subscores should be reported, and as suggested above, decisions should be made by taking into account available model fit indices.

Discussion

In this article, we reviewed available full-featured packages in R to report subscores within four frameworks including CTT, MIRT, CDM, and FA. We also briefly compared features of different packages when there are more than one package available within the same framework. Specifically, we reviewed performance of selected packages in examining psychometric quality of the subscores and reporting subscores when they can be reported. The 2011 TIMSS eighth-grade mathematic data were used for a real data example to illustrate how to use selected packages to report subscores.

We limited our scope to R software primarily for two reasons: (1) R is a free and computational efficient tool, and (2) it contains powerful packages that allow users, including those with limited programing skills, to conduct various analyses and obtain nicely formatted outputs. Decisions on which package to use should be made based on both theoretical and practical considerations including but not limited to (1) whether latent trait variables are hypothesized; (2) whether subscales are of a complex structure (i.e., in the presence of multiple- or cross-loadings); (2) measurement scales of subscale variables (i.e., discrete or continuous); (3) whether item responses are dichotomous, polytomous, or continuous; and (4) model selection and parameter estimation methods. Table 2 presented a brief review on all the packages discussed in this article. Alongside Table 2, readers who are interested in reporting subscores may use information in this table to decide which method and package to use in analysis. We also noted that results from different methods and packages might vary and even be contradictory when examining whether subscores were of added value. In the TIMSS example, PRMSE statistics from the CTT framework showed that only algebra and data and chance subscales were of added value while number and geometry subscales were not. Global model fit indices from MIRT, CDM, and CFA methods all indicated a relative poor model–data fit when a four-dimensional model was considered. Fit statistics on item level and individual level revealed that a small portion of items and examinees were ill fitting.

Finally, we noted that the review of R packages in reporting subscores might not be comprehensive and exhaustive due to the rapid development of R and active maintenance of these packages. For example, the latest version of the mirt package also allows users to fit some CDMs such as DINA and DINO. In this review, we chose to only review main features of these packages in reporting subscores and refer readers to their original documentations for the comprehensive descriptions of their package features (as cited).

Declaration of Conflicting Interests

The author(s) declared no conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Notes

1. In the literature, estimated subscores of the above three types are usually referred to as subscore$_s$ (or s_s), subscore$_x$ (or s_x), and subscore$_{sx}$ (or s_{sx}), respectively (e.g., Haberman, 2008; Sinharay, 2010).

2. Simple structure means each item on a test is associated with only one subscale, while complex structure allows the existence of items that are associated with more than one subscales.

References

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. Washington, DC: American Educational Research Association.

Boker, S., Neale, M., & Maes, H. (2017). *OpenMx: Extended structural equation modelling* (R package version 2.7.4) [Computer software]. Retrieved from https://cran.r-project.org/package=OpenMx

Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*, 1–29.

Chalmers, R. P. (2017). *mirt: Multidimensional item response theory* (R package version 1.22) [Computer software]. Retrieved from http://CRAN.R-project.org/package=mirt

Chen, J., de la Torre, J., & Zhang, Z. (2013). Relative and absolute fit evaluation in cognitive diagnosis modeling. *Journal of Educational Measurement*, *50*, 123–140.

Chiu, C.-Y., Köhn, H.-F., Zheng, Y., & Henson, R. (2015). Exploring joint maximum likelihood estimation for cognitive diagnosis models. In L. A. van der Ark, D. M. Bolt, W-C. Wang, J. A. Douglas, & S-M. Chow (Eds.), *Quantitative psychology research* (pp. 263–277). Madison, WI: Springer.

Dai, S., Wang, X., & Svetina, D. (2016). *subscore: Sub-score computing functions in classical test theory* (R package version 1.3) [Computer software]. Retrieved from http://CRAN.R-project.org/package=subscore

de La Torre, J. (2009). A cognitive diagnosis model for cognitively based multiple-choice options. *Applied Psychological Measurement*, *33*, 163–183.

Drasgow, F., Levine, M. V., & Williams, E. A. (1985). Appropriateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology*, *38*, 67–86.

Dwyer, A., Boughton, K. A., Yao, L., Steffen, M., & Lewis, D. (2006). *A comparison of subscale score augmentation methods using empirical data*. Paper presented at the annual meeting of the national council of measurement in education, San Francisco, CA.

Fox, J., Nie, Z., & Byrnes, J. (2016). *sem: Structural equation models* (R package version 3.1-8) [Computer software]. Retrieved from https://cran.r-project.org/package=sem

Haberman, S. J. (2008). When can subscores have value? *Journal of Educational and Behavioral Statistics*, *33*, 204–229.

Haberman, S. J., & Sinharay, S. (2010). Reporting of subscores using multidimensional item response theory. *Psychometrika*, *75*, 209–227.

Haberman, S. J., & Sinharay, S. (2013). Does subgroup membership information lead to better estimation of true subscores? *British Journal of Mathematical and Statistical Psychology*, *66*, 452–469.

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, *25*, 258–272.

Kang, T., & Chen, T. T. (2007). *An investigation of the performance of the generalized S-X^2 item-fit index for polytomous IRT models* (ACT Research Report Series, 2007-1). Iowa City, IA: ACT, Inc.

Kiefer, T., Robistzsch, A., & Wu, M. (2017). *TAM: Test analysis modules* (R package version 1.99999-31) [Computer software]. Retrieved from https://cran.r-project.org/package=TAM

Liu, Y., Douglas, J. A., & Henson, R. A. (2009). Testing person fit in cognitive diagnosis. *Applied Psychological Measurement*, *33*, 579–598.

Maydeu-Olivares, A., & Joe, H. (2006). Limited information goodness-of-fit testing in multidimensional contingency tables. *Psychometrika*, *71*, 713–732.

Mullis, I. V., Martin, M. O., Gonzalez, E. J., & Chrostowski, S. J. (2004). *TIMSS 2003 International Mathematics Report: Findings from IEA's Trends in International Mathematics and Science Study at the Fourth and Eighth Grades*. Chestnut Hill, MA: TIMSS & PIRLS International Study Center, Boston College.

Muthén, L. K., & Muthén, B. O. (2015). *Mplus user's guide: Statistical analysis with latent variables: User's guide* (7th ed.). Los Angeles, CA: Muthén & Muthén.

Orlando, M., & Thissen, D. (2000). Likelihood-based item-fit indices for dichotomous item response theory models. *Applied Psychological Measurement*, *24*, 50–64.

Orlando, M., & Thissen, D. (2003). Further investigation of the performance of S-X2: An item fit index for use with dichotomous item response theory models. *Applied Psychological Measurement*, *27*, 289–298.

Pellegrino, J. W., Chudowsky, N., & Glaser, R. (2001). *Knowing what students know. The science and design of educational assessment*. Washington, DC: National Academies Press.

R Core Team. (2016). *R: A language and environment for statistical computing* (Version 3.2.3) [Computer software]. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/

Ravand, H., & Robitzsch, A. (2015). Cognitive diagnostic modeling using R. *Practical Assessment, Research & Evaluation*, *20*, 1–12.

Roberts, M. R., & Gierl, M. J. (2010). Developing score reports for cognitive diagnostic assessments. *Educational Measurement: Issues and Practice*, *29*, 25–38.

Robistzsch, A. (2017). *sirt: Supplementary item response theory models* (R package version 1.15-41) [Computer software]. Retrieved from https://cran.r-project.org/package=sirt

Robistzsch, A., Kiefer, T., George, A. C., & Uenlue, A. (2017). *CDM: Cognitive diagnosis modeling* (R package version 5.4-0) [Computer software]. Retrieved from https://cran.r-project.org/package=CDM

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*, 1–36.

Rosseel, Y. (2017). *lavaan: Latent variable analysis* (R package version 0.5-23.1097) [Computer software]. Retrieved from https://cran.r-project.org/package=lavaan

Sinharay, S. (2010). When can subscores be expected to have added value? Results from operational and simulated data. *ETS Research Report Series, 2010*, i–28.

Sinharay, S., Haberman, S., & Puhan, G. (2007). Subscores based on classical test theory: To report or not to report. *Educational Measurement: Issues and Practice*, *26*, 21–28.

Sinharay, S., Puhan, G., & Haberman, S. J. (2011). An NCME instructional module on subscores. *Educational Measurement: Issues and Practice*, *30*, 29–40.

Tate, R. L. (2004). Implications of multidimensionality for total score and subscore performance. *Applied Measurement in Education*, *17*, 89–112.

Templin, J. L., & Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods*, *11*, 287.

Wang, C., Shu, Z., Shang, Z., & Xu, G. (2015). Assessing item-level fit for the DINA model. *Applied Psychological Measurement*, *39*, 525–538.

Yao, L., & Boughton, K. A. (2007). A multidimensional item response modeling approach for improving subscale proficiency estimation and classification. *Applied Psychological Measurement*, *31*, 83–105.

Zheng, Y., & Chiu, C.-Y. (2016). *NPCD: Nonparametric methods for cognitive diagnosis* (R package version 1.0-10) [Computer software]. Retrieved from https://cran.r-project.org/package=NPCD

Authors

SHENGHAI DAI is a PhD candidate of inquiry methodology at Indiana University, 201 N. Rose Avenue, Bloomington IN, 47405; email: dais@indiana.edu. His research interests are psychometrics and quantitative methodology, including item response theory, cognitive diagnostic models, large-scale assessment, subscore reporting, and missing data.

DUBRAVKA SVETINA is an assistant professor of inquiry methodology at Indiana University, 201 N. Rose Avenue, Bloomington IN, 47405; email: dsvetina@indiana.edu. Her research interests involve methodological investigations of the performance of current psychometric models and procedures, including (multidimensional) item response theory, dimensionality assessment, cognitive diagnostic models, measurement invariance, and subscore reporting in educational assessment.

XIAOLIN WANG is a PhD candidate of inquiry methodology at Indiana University, 201 N. Rose Avenue, Bloomington IN, 47405; email: xw41@indiana.edu. Her research interests involve the application of psychometrics to educational assessments and surveys, including (multidimensional) item response theory, subscore reporting, measurement invariance, survey response style, and cognitive diagnostic models.

Manuscript received June 27, 2016
First revision received September 3, 2016
Second revision received March 15, 2017
Accepted March 28, 2017