Supplementary online materials for:
A flexible full-information approach to the modeling of response styles

Carl F. Falk
Michigan State University

Li Cai
University of California, Los Angeles

This research was partially supported by a Social Sciences and Humanities Research Council of Canada Postdoctoral Fellowship awarded to Carl F. Falk. Li Cai's research is partially supported by a grant from the Institute of Education Sciences (R305D140046).

Address all correspondence to: Carl F. Falk, Measurement and Quantitative Methods, Michigan State University, 458 Erickson Hall, East Lansing, MI, USA 48824.
Email: falkcarl@msu.edu. Phone: 517-355-6684.

1 Additional results for PROMIS®smoking data

Following the text in these supplementary materials, Table 1 contains results of additional models fit to the six PROMIS®smoking data dimensions that are not reported in the full manuscript. Figure 1 plots EAP scores from Model 1 and 10 of the main manuscript at three levels of midpoint response style (MRS).

2 Additional simulation details and results

Tables 2 through 5 list item parameters for data generating models in Simulation 1 and 2 in the main manuscript. Tables 6 to 10 list factor correlation matrices for these data generating models. With the exception of the factor correlation between hedonic benefits and nicotine dependence in Table 10, all parameters were obtained from models estimated on actual PROMIS®data as described in the main manuscript.

3 Example code for model identification and fitting

This section contains supplemental R code (R Core Team, 2012) for the manuscript. The first section loads custom functions and necessary libraries, including *mirt* (Chalmers, 2012), *mvtnorm* (Genz et al., 2013), *numDeriv* (Gilbert & Varadhan, 2012), *MCMCpack* (Martin, Quinn, & Park, 2011), and *Matrix* (Bates & Maechler, 2014) packages. The following section provides an example check of identification of a model with a substantive factor and two response style factors (ERS and MRS; all correlated) with item slopes that are unconstrained across items. The final example demonstrates generating data from the MNRM and fitting the data using a custom item type with the mirt package.

3.1 Libraries and custom functions

The custom functions include those necessary for computing the response functions for the MNRM, a utility function for allowing some automation of using the response function for the MNRM with the mirt package, generating data from the MNRM, updating and getting item parameters, and computing model-implied probabilities for either all possible response patterns or all bivariate relationships.

```
######################################################################
##  Copyright 2014-2015 Carl F. Falk
##
##  This program is free software: you can redistribute it and/or
##    modify it under the terms of the GNU General Public License as
##    published by the Free Software Foundation, either version 2 of
##    the License, or (at your option) any later version.
##
##    This program is distributed in the hope that it will be useful,
##    but WITHOUT ANY WARRANTY; without even the implied warranty of
##    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
##    GNU General Public License for more details.
##    <http://www.gnu.org/licenses/>
##
##  All code was written by Carl F. Falk, with the exception of
##  Tnom.trend from flexMIRT support page PlottingExamples.zip
##  available at: <https://flexmirt.vpgcentral.com/Support/>

######################################################################
## Load external libraries
######################################################################

library(mirt)
library(mvtnorm)
library(numDeriv)
library(MCMCpack)
library(Matrix)

######################################################################
## Define Functions
######################################################################

## MNRM traceline function using Thissen & Cai parameterization.
## par: Vector of parameters
## Theta: Matrix of latent trait values
## ncat: Number of categories
## ndim: Number of latent trait dimensions
## Ta: List of Ta matrices for each dimension
## Tc: Tc matrix for intercepts
trace.MNRM.TC<-function(par,Theta,ncat,ndim,Ta,Tc,y=NULL){
    a<-par[1:ndim]
    gamma<-par[(ndim+1):(ndim+ncat-1)]
    alpha<-par[(ndim+ncat):(length(par))]

    N<-nrow(Theta)
    l<-rep(1,N)

    ## intercepts
    c<-Tc%*%gamma

    ## scoring functions
    S<-matrix(0,nrow=ndim,ncol=ncat)
    indx<-1
    for (i in 1:ndim){
        S[i,]<-Ta[[i]]%*%alpha[indx:(indx+ncat-2)]
        indx<-indx+ncat-1
    }

    Z<-Theta%*%(a*S)+l%*%t(c)
    Z<-exp(Z)
    P<-Z/rowSums(Z)

    if(!is.null(y)){
        ty<-vector()
        for(i in 0:(ncat-1)){
            ty<-cbind(ty,y==i)
        }
        P<-rowSums(P*ty)
    }

    return(P)
}

## Function to modify trace.MNRM.ThissenCai to be compatible with mirt
## package. Such work can be done manually by item. This is an attempt
## to automate some scoring function definitions used in our paper.
## Returns default starting values, estimated & fixed parameters.
## Currently works for odd category items with nrm, gpc, ers, mrs,
## and emrs scoring functions. Could be modified for use with even
## category items and other scoring function combinations.
## dims: Vector indicating how to define scoring function
##   for each latent dimension, in order. For example,
##   ’nrm’, ’gpc’, ’ers’, ’mrs’, ’emrs’, or ’n’ for 0 slope.
## ncat: Number of categories
trace.MNRM.mirt<-function(dims,ncat){

    ndim<-length(dims)

    ## slopes
    a<-ifelse(dims=="n",0,.5)
```

```
a.free<-ifelse(dims=="n",FALSE,TRUE)
a.names<-paste("a",1:ndim,sep="")

## intercepts
Tc<-Tnom.trend(ncat)
gamma<-rep(0,ncat-1)
gamma.free<-rep(TRUE,ncat-1)
gamma.names<-paste("gamma",1:(ncat-1),sep="")

## scoring functions
Ta<-list()
alpha<-NULL
alpha.free<-NULL
alpha.names<-NULL
for(i in 1:ndim){
    alpha.names<-c(alpha.names,
                   paste("alpha",i,"_",1:(ncat-1),sep=""))
    Ta[[i]]<-Tnom.trend(ncat)
    alpha<-c(alpha,c(1,rep(0,ncat-2)))
    if(dims[i]!="nrm"){
        Ta[[i]][,2:(ncat-1)]<-0
        alpha.free<-c(alpha.free,rep(FALSE,ncat-1))
    } else {
        alpha.free<-c(alpha.free,c(FALSE,rep(TRUE,ncat-2)))
    }

    if (dims[i]=="ers"){
        Ta[[i]][,1]<-c(1,rep(0,ncat-2),1)
    } else if (dims[i]=="mrs"){
        Ta[[i]][,1]<-c(rep(0,(ncat-1)/2),1,rep(0,(ncat-1)/2))
    } else if (dims[i]=="emrs"){
        Ta[[i]][,1]<-c(seq((ncat-1)/2,1),0,seq(1,(ncat-1)/2))
    }
}

start.par<-c(a,gamma,alpha)
est.par<-c(a.free,gamma.free,alpha.free)
names.par<-c(a.names,gamma.names,alpha.names)

names(start.par)<-names.par
names(est.par)<-names.par
formals(trace.MNRM.TC)<-
    c(alist(par=,Theta=,ncat=),
      pairlist(ndim=ndim,Ta=Ta,Tc=Tc,y=NULL))
    return(list(func=trace.MNRM.TC,
                start.par=start.par,est.par=est.par))
}

## Creates T matrix for Ta or Tc.
## From flexMIRT support page PlottingExamples.zip available at:
## https://flexmirt.vpgcentral.com/Support/
## nc: Number of categories
Tnom.trend <- function(nc) {
    T <- matrix(0,nc,nc-1)
    for (i in 1:nc) {
        T[i,1] <- i-1
    }
    for (k in 2:(nc-1)) {
        for (i in 2:(nc-1)) {
            T[k,i] <- sin(pi*(i-1)*(k-1)/(nc-1))
        }
    }
    return(T)
}

## simulates MNRM data
sim.mnrm<-function(N,covmu,items){
    D<-covmu$ndim
    sigma<-xpnd(covmu$cov)
    mu<-covmu$mu
    nitem<-length(items)
    X<-rmvnorm(N,mu,sigma=sigma)
    Y<-matrix(NA,nrow=N,ncol=nitem)
    for(i in 1:nitem){
        item<-items[[i]]
        P<-trace.MNRM.TC(item$par,X,item$ncat,
                         item$ndim,item$Ta,item$Tc)
        for(j in 1:N){
            Y[j,i]<-sample(0:(ncat-1),1,prob=P[j,])
        }
    }
    return(Y)
}

## Get parameters from NRM item
get.params<-function(item, free=TRUE){
    if (item$type=="NRM"){
        if(free){
            a<-item$a[item$afree]
            alp<-unlist(item$alpha)[unlist(item$alphafree)]
            gam<-item$gamma[item$gammafree]
        } else {
            a<-item$a
            alp<-unlist(item$alpha)
            gam<-item$gamma
        }
        p<-c(a,gam,alp)
    }
    return(p)
}

## Update parameters for NRM item
set.params<-function(item,p,free=TRUE){
    indx<-1
    if (item$type=="NRM"){
        D<-length(item$a)
        if(free){
            item$a[item$afree]<-p[indx:(indx+sum(item$afree)-1)]
            indx<-indx+sum(item$afree)
            item$gamma<-p[indx:(indx+sum(item$gammafree)-1)]
            indx<-indx+sum(item$gammafree)
            for(j in 1:D){
                item$alpha[[j]][item$alphafree[[j]]]<-
                    p[indx:(indx+sum(item$alphafree[[j]])-1)]
                indx<-indx+sum(item$alphafree[[j]])
            }
        } else {
            item$a<-p[indx:(indx+D-1)]
            indx<-indx+D
            item$gamma<-p[indx:(indx+item$ncat-2)]
            indx<-indx+item$ncat-1
            for(j in 1:D){
                item$alpha[[j]]<-p[indx:(indx+item$ncat-2)]
                indx<-indx+item$ncat-1
            }
        }
        item$par<-c(item$a,item$gamma,unlist(item$alpha))
    }
    return(item)
}

## Get parameters from list of items
get.params.items<-function(items, free=TRUE){
    ni<-length(items)
    p<-vector("numeric")
    for(i in 1L:ni){
        p<-c(p,get.params(items[[i]],free))
    }
    return(p)
}

## Update parameters for list of items
set.params.items<-function(items,p,free=TRUE){
    ni<-length(items)
    indx<-1
    for(i in 1L:ni){
        np<-length(get.params(items[[i]],free))
        items[[i]]<-set.params(items[[i]],p[indx:(indx+np-1)],free)
        indx<-indx+np
    }
    return(items)
}

## Density of response patterns as product of tracelines
## Y: Response Matrix
## X: Latent Trait Matrix
## items: List of items
den.pat<-function(Y,X,items){
    ni<-ncol(Y)
    N<-nrow(Y)
    result<-rep(1,N)
    for(i in 1:ni){
        item<-items[[i]]
        na<-!is.na(Y[,i])
        if(sum(na)>0){
            result[na]<-result[na]*
                trace.MNRM.TC(item$par,X[na,],item$ncat,
                              item$ndim,item$Ta,item$Tc,Y[na,i])
        }
    }
    result
}

## Create matrix of possible response vectors from list of items.
## Assumes categorical items.
## items: List of items
## order: Controls whether all possible (full) or all bivariate
##   response patterns are produced
possresp<-function(items,order="full"){
    ni<-length(items)
    cats<-lapply(items,function(x){0:(x$ncat-1)})
    if(order=="full"){
        resp<-expand.grid(cats)
    } else if (order=="second"){ ## bivariate resp patterns
        resp<-matrix(NA,nrow=0,ncol=ni)
        for(i in 1:ni){
            for(j in i:ni){
                if(i!=j){
                    resp<-rbind(
                        resp,
                        expand.grid(c(replicate(i-1,NA),
                                      list(cats[[i]]),
                                      replicate(j-i-1,NA),
                                      list(cats[[j]]),
                                      replicate(ni-j,NA))))
                }
            }
        }
    }
    return(resp)
}

## Compute all model implied probabilities for response matrix.
## resp: Response matrix, each row a response vector.
## items: List of items
## Function assumes nodes (theta) and weights (w)
##   are passed as arguments
mi.probs<-function(resp,items,theta,w){
    nq<-length(w)
    ni<-length(items)
    nresp<-nrow(resp)
    presp<-vector("numeric",nresp)
    for(i in 1:nresp){
        testpat<-resp[rep(i,each=nq),]
        presp[i]<-sum(den.pat(testpat,theta,items)*w)
    }
    return(presp)
}

## Takes single vector of model parameters, p, as first argument and
## then computes model implied probabilities assuming normally
## distributed traits. Useful for passing to functions that compute
## numerical derivatives.
## p: Vector of parameters. Must be same length as total number of
##   free parameters in "group"
## group: List containing items, group parameters, and response
##   patterns
## theta: min/max of quadrature nodes for each dimension
## nq: number of quadrature points per dimension
irt.probs<-function(p,group,theta=c(-3,3),nq=11){

    items<-group$items
    covmu<-group$covmu
    resp<-group$resp
    ni<-length(items)
    npitem<-length(get.params.items(items))
    npcov<-sum(covmu$free)
    npmu<-sum(covmu$mufree)

    ## set parameter values
    indx<-1
    if(npitem>0){
        items<-set.params.items(items,p[indx:(indx+npitem-1)])
        indx<-indx+npitem
    }
    if(npcov>0){
        covmu$cov[covmu$free]<-p[indx:(indx+npcov-1)]
        indx<-indx+npcov
    }
    if(npmu>0){
        covmu$mu[covmu$mufree]<-p[indx:(indx+npmu-1)]
        indx<-indx+npmu
    }

    ## set up nodes
    theta<-seq(theta[1],theta[2],length.out=nq)
    theta<-as.matrix(expand.grid(rep(list(theta),covmu$ndim)))
    w<-dmvnorm(theta,covmu$mu,xpnd(covmu$cov))
    w<-w/sum(w)

    miprobs<-mi.probs(as.matrix(resp),items,theta,w)
    return(miprobs)
}
```

3.2 Identification example

This code sets up a single-group model with 5 items with 5 response options each. All items are modeled by three correlated dimensions (Substantive, ERS, and MRS) with scoring functions specified in the comments below. In order to test whether the model is identified, numerical derivatives are used to compute the $(m \times p)$ Jacobian, $J = \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\eta}}$, where $\boldsymbol{\pi}$ is a vector of the m model implied probabilities for each response pattern and $\boldsymbol{\eta}$ are the p free model parameters. Code is given for differentiating model-implied probabilities for the full 5^5 possible response patterns, or derivatives of only bivariate relationships among variables (e.g., first and second order marginals). A Jacobian of full column rank, as obtained here, suggests that the model is identified.

```
######################################################################
## Try a response style model. This one is a tri-dimensional model
## with 5 items that load on all factors, including a substantive
## dimension [0 1 2 3 4], ERS [1 0 0 0 1], and MRS [0 0 1 0 0].
## Correlation among all latent traits are free and nonzero

## Set up items
ncat<-5
nitem<-5
itemsERSMRS<-list()
for(i in 1:nitem){
    itemsERSMRS[[i]]<-list()
    itemsERSMRS[[i]]$type<-"NRM"
    itemsERSMRS[[i]]$ndim<-3
    itemsERSMRS[[i]]$ncat<-ncat
    itemsERSMRS[[i]]$a<-1.49 ## GPC slope
    itemsERSMRS[[i]]$afree<-TRUE
    itemsERSMRS[[i]]$alpha<-list()
    itemsERSMRS[[i]]$alpha[[1]]<-matrix(
        c(1,rep(0,ncat-2)),ncol=1,nrow=ncat-1)
    itemsERSMRS[[i]]$alphafree<-list()
    itemsERSMRS[[i]]$alphafree[[1]]<-rep(FALSE,ncat-1)
    itemsERSMRS[[i]]$gamma<-matrix(
        c(-.05,3.13,.02,.04),ncol=1,nrow=ncat-1)
    itemsERSMRS[[i]]$gammafree<-rep(TRUE,ncat-1)
    itemsERSMRS[[i]]$Ta<-list()
    itemsERSMRS[[i]]$Ta[[1]]<-Tnom.trend(ncat)
    itemsERSMRS[[i]]$Ta[[1]][,2:(ncat-1)]<-0
    itemsERSMRS[[i]]$Tc<-Tnom.trend(ncat)
    itemsERSMRS[[i]]$a[2]<-1.68 ## ERS slope
    itemsERSMRS[[i]]$afree[2]<-TRUE
    itemsERSMRS[[i]]$alpha[[2]]<-matrix(
        c(1,rep(0,ncat-2)),ncol=1,nrow=ncat-1)
    itemsERSMRS[[i]]$alphafree[[2]]<-rep(FALSE,ncat-1)
    itemsERSMRS[[i]]$Ta[[2]]<-Tnom.trend(ncat)
    itemsERSMRS[[i]]$Ta[[2]][,2:(ncat-1)]<-0
    itemsERSMRS[[i]]$Ta[[2]][,1]<-c(1,0,0,0,1) ## scoring function
    itemsERSMRS[[i]]$a[3]<-.76 ## MRS slope
    itemsERSMRS[[i]]$afree[3]<-TRUE
    itemsERSMRS[[i]]$alpha[[3]]<-matrix(
        c(1,rep(0,ncat-2)),ncol=1,nrow=ncat-1)
    itemsERSMRS[[i]]$alphafree[[3]]<-rep(FALSE,ncat-1)
    itemsERSMRS[[i]]$Ta[[3]]<-Tnom.trend(ncat)
    itemsERSMRS[[i]]$Ta[[3]][,2:(ncat-1)]<-0
    itemsERSMRS[[i]]$Ta[[3]][,1]<-c(0,0,1,0,0) ## scoring function
    itemsERSMRS[[i]]$par<-get.params(itemsERSMRS[[i]],free=FALSE)
}

## Set up mean and covariance of latent variables
covmuERSMRS<-list()
covmuERSMRS$ndim<-3
covmuERSMRS$cov<-c(1,.2,-.2,1,-.3,1) ## latent trait covariances
covmuERSMRS$free<-ifelse(covmuERSMRS$cov==1,FALSE,TRUE)
covmuERSMRS$mu<-rep(0,3)
covmuERSMRS$mufree<-rep(FALSE,3)

## set up items and mean/cov structure in a single list
ERSMRS<-list(items=itemsERSMRS,covmu=covmuERSMRS)

## free parameters
p.items<-get.params.items(ERSMRS$items)
p.cov<-ERSMRS$covmu$cov[ERSMRS$covmu$free]
p.mu<-ERSMRS$covmu$mu[ERSMRS$covmu$mufree]
testp<-c(p.items,p.cov,p.mu)
np<-length(testp)

## Compute Jacobian of model at current parameter values
## Using "full" set of possible response patterns
## 3-dimensional model will take a while to compute
ERSMRS$resp<-possresp(ERSMRS$items,order="full")
J<-jacobian(irt.probs, testp, group=ERSMRS,method="simple",nq=7)
```

```
rankMatrix(J,tol=1e-2) ## should be same as np for identified models
## low tol means more likely to label model as not identified
np
svd(J)$d ## manually look at singular values

## Compute Jacobian of model at current parameter values
## Using first and second-order marginals
ERSMRS$resp<-possresp(ERSMRS$items,order="second")
J<-jacobian(irt.probs, testp, group=ERSMRS,method="simple",nq=7)
rankMatrix(J,tol=1e-2) ## should be same as np for identified models
## low tol means more likely to label model as not identified
np
svd(J)$d ## manually look at singular values

## Accuracy may be increased with more quadrature points (np) and
## with better method for derivative computation (e.g., Richardson).
## This would reduce need to set low tolerance level for rankMatrix.
```

3.3 Model fitting example

This section is intended to illustrate the feasibility and our progress of coding model fitting for the MNRM for possible inclusion in a stand-alone or existing R package. In addition, future releases of *flexMIRT*®(Cai, 2013) are planned that will include the current paper's MNRM parameterization and may also be used for fitting any of the models specified in this paper. Alternatively, an anonymous reviewer pointed out that *Latent Gold* (Vermunt & Magidson, 2005) may be able to currently fit the some response style models discussed in our paper, albeit using a different estimation algorithm.

Data is generated following the model in the previous section. The *mirt* package (Chalmers, 2012) is then used to do model fitting, including the GPC using *mirt*'s built-in code and parameterization, a GPC response function based on custom code, and the true data generating model also based on a custom item definition. The final comments include the true model parameters, which are reasonably close to the estimated values.

Note that in this example *mirt* is using quadrature and by default will use numerical derivatives when supplied with a user-specified item definition unless a gradient and hessian function are supplied. Estimation speed is expected to deteriorate with more items and dimensions. It can be improved by using MHRM and writing custom functions to compute analytical derivatives for item parameters.

```
######################################################################
## Tests of fitting MNRM response style model to simulated data
## using the 'mirt' package.
## Below uses quadrature and numerical derivatives, which will
## not be as efficient as MHRM and analytical derivatives with
## a large number of dimensions and items. Code used for
## simulations and the empirical example used a custom
## implementation of MHRM and analytical derivatives (i.e., not
## the mirt pacakge) that is currently being prepared for release.

## Generate data using above model
N<-50000
set.seed(5234)
dat<-sim.mnrm(N,covmuERSMRS,itemsERSMRS)

## Fit GPC model to data using built-in GPC model
mirt.gpc<-mirt(dat,1,itemtype="gpcm")
mirt.gpc
coef(mirt.gpc)

## Fit GPC model w/ custom GPC traceline
## Slower due to use of numeric derivatives
gpc.trace<-trace.MNRM.mirt("gpc",5)
GPC_test<-createItem("GPC_test",
                     gpc.trace$start.par,
                     gpc.trace$est.par,
                     gpc.trace$func)
custom.gpc<-mirt(dat,1,itemtype="GPC_test",
                 customItems=list(GPC_test=GPC_test))
custom.gpc
coef(custom.gpc)

## Fit true model (GPC, ERS, and MRS) to data
mymodel<-"F1=1-5
F2=1-5
F3=1-5
COV=F1*F2*F3"
mymodel<-mirt.model(mymodel)
ersmrs.trace<-trace.MNRM.mirt(c("gpc","ers","mrs"),5)
ERSMRS_test<-createItem("ERSMRS_test",ersmrs.trace$start.par,
                        ersmrs.trace$est.par,ersmrs.trace$func)
mirt.ersmrs<-mirt(dat,mymodel,itemtype="ERSMRS_test",
                  customItems=list(ERSMRS_test=ERSMRS_test))
mirt.ersmrs
coef(mirt.ersmrs)
## True item params are a1=1.49, a2=1.68, a3=.76, gamma1=-.05,
## gamma2=3.13, gamma3=.02, gamma4=.04
## True group params are COV_21 = .2, COV_31 = -.2, cov_32 = -.3
```

References

Bates, D., & Maechler, M. (2014). *Matrix: Sparse and dense matrix classes and methods.* Retrieved from `http://CRAN.R-project.org/package=Matrix` (R package version 1.1-4)

Cai, L. (2013). flexMIRT® version 2: Flexible multilevel item factor analysis and test scoring [Computer software]. Chapel Hill, NC: Vector Psychometric Group, LLC.

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

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., & andTorsten Hothorn, F. S. (2013). *mvtnorm: Multivariate Normal and t Distributions.* Retrieved from `http://CRAN.R-project.org/package=mvtnorm` (R Package Version 0.9-9996)

Gilbert, P., & Varadhan, R. (2012). *numderiv: Accurate numerical derivatives.* Retrieved from `http://CRAN.R-project.org/package=numDeriv` (R package version 2012.9-1)

Martin, A. D., Quinn, K. M., & Park, J. H. (2011). MCMCpack: Markov chain monte carlo in R. *Journal of Statistical Software*, *42*(9), 22. Retrieved from `http://www.jstatsoft.org/v42/i09/`

R Core Team. (2012). *R: A language and environment for statistical computing.* Vienna, Austria. Retrieved from `http://www.R-project.org` (ISBN 3-900051-07-0)

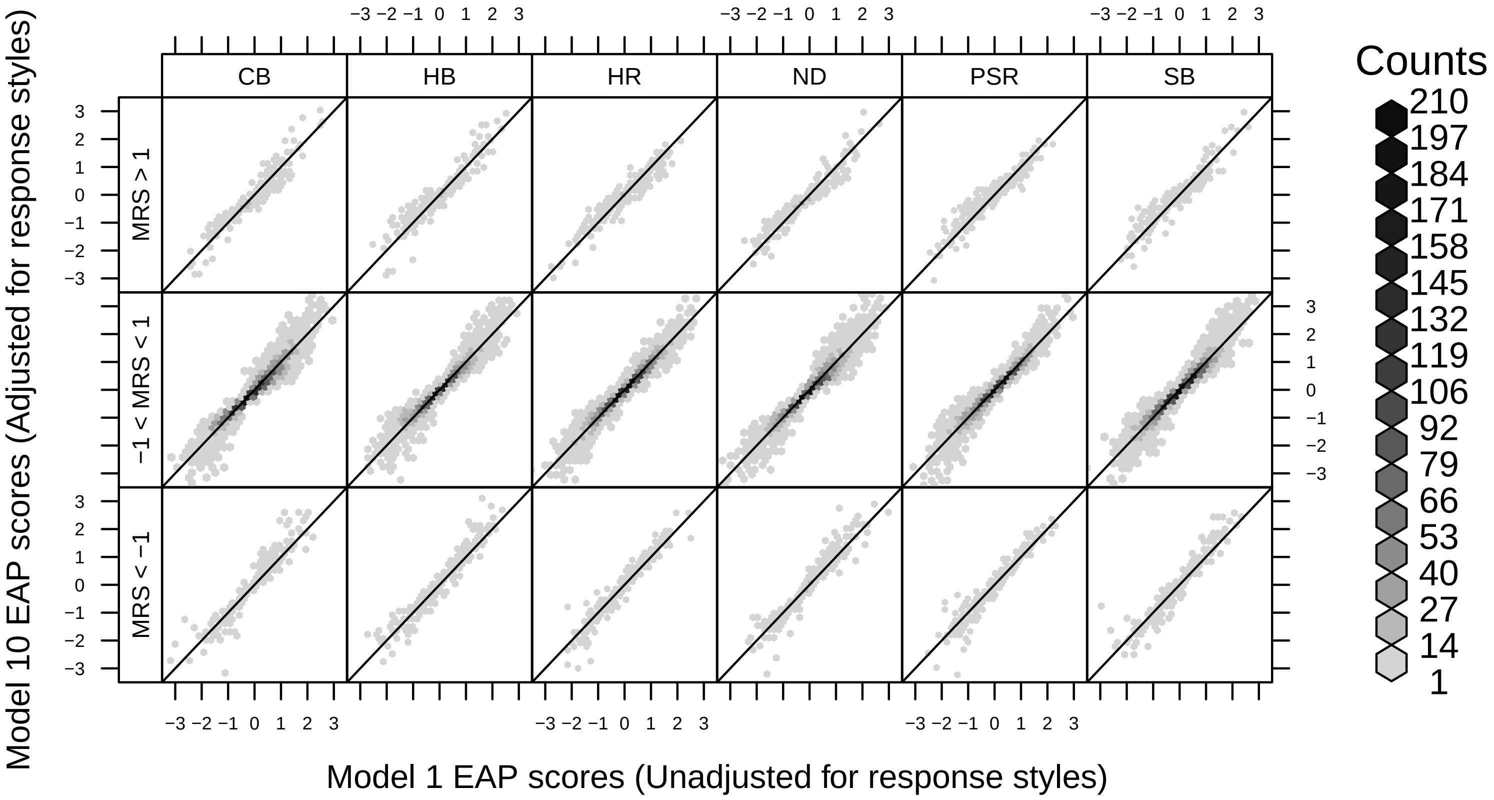
Vermunt, J. K., & Magidson, J. (2005). *Latent GOLD 4.0 User's Guide*. Belmont, Massachusetts: Statistical Innovations Inc.

Table 1: Additional models fit to PROMIS® smoking data

Dimensions	**a**	*r*	# param.	AIC	BIC
6D, ERS	free, free	0	872	[744685, 744848]	[750432, 750595]
6D, ERS	free, equal	0	735	[746185, 746347]	[751030, 751192]
6D, MRS	free, free	0	872	[762037, 762187]	[767785, 767935]
6D, MRS	free, equal	0	735	[762071, 762220]	[766916, 767065]
6D, EMRS	free, free	0	872	[750364, 750524]	[756111, 756272]
6D, EMRS	free, equal	0	735	[751330, 751487]	[756175, 756332]
6D, ERS, MRS	free, free free	0	1012	[740479, 740674]	[747149, 747345]
6D, ERS, MRS	free, equal, free	0	875	[742077, 742268]	[747845, 748035]
6D, ERS, MRS	free, free, equal	0	875	[740690, 740881]	[746457, 746649]
6D, ERS, MRS	free, equal, equal	0	738	[742262, 742451]	[747126, 747315]
6D, NRM	free, equal	0	738	[745540, 745704]	[750404, 750568]
6D ST	free, equal	0	339	[782487, 782605]	[784721, 784839]
6D ST, NRM	free, equal	0	345	[758573, 758734]	[760847, 761008]

Note. "**a**" indicates whether slopes were free across items or constrained equal across items for each dimension listed. "*r*" indicates whether correlations between response style and each of the substantive factors were estimated or fixed to 0. AIC and BIC are 95% CI estimates based on a Monte Carlo integration estimate of the marginal log-likelihood for each model.

Figure 1: Substantive trait EAP scores at three grouped levels of MRS.



Note. CB = Coping benefits; HB = Hedonic benefits; HR = Health risks; ND = Nicotine dependence; PSR = Psychosocial risks; SB = Social benefits.

Table 2: Item parameters for GPC data generating model

Item	a_1	a_2	γ_1	γ_2	γ_3	γ_4
1	1.41	0.00	0.11	2.59	0.06	-0.03
2	1.55	0.00	-0.18	3.29	-0.07	-0.20
3	1.03	0.00	-0.43	1.15	-0.01	-0.17
4	1.62	0.00	0.42	3.14	0.17	0.10
5	0.87	0.00	-0.51	0.95	-0.04	-0.29
6	1.20	0.00	-0.31	2.18	0.11	-0.06
7	1.01	0.00	-0.51	1.49	-0.05	-0.12
8	1.06	0.00	0.79	2.11	0.10	0.06
9	0.85	0.00	0.30	1.12	0.09	-0.08
10	0.94	0.00	-0.46	1.38	-0.05	-0.15
11	1.70	0.00	0.34	2.78	0.08	0.05
12	1.27	0.00	0.04	2.23	0.10	0.12
13	0.98	0.00	-0.11	1.56	0.02	-0.07
14	1.48	0.00	1.59	3.14	0.14	0.11
15	1.72	0.00	0.53	2.92	0.02	-0.01
16	0.74	0.00	-0.57	0.92	0.15	-0.14
17	0.00	2.02	0.14	2.59	-0.00	0.02
18	0.00	2.31	1.43	2.85	0.11	0.14
19	0.00	2.29	1.29	3.39	0.09	0.14
20	0.00	2.73	0.82	3.65	0.25	0.21
21	0.00	0.84	0.51	1.43	0.03	-0.06
22	0.00	2.41	0.85	4.04	0.19	0.21
23	0.00	2.38	-0.29	4.07	0.17	0.02
24	0.00	1.53	-0.28	2.30	-0.04	-0.06
25	0.00	0.67	-0.22	1.57	-0.15	-0.07
26	0.00	1.10	-0.52	2.24	0.15	-0.02
27	0.00	1.66	0.36	2.41	0.08	0.06
28	0.00	1.48	0.33	1.93	0.24	0.13
29	0.00	1.65	0.47	2.42	0.11	-0.04
30	0.00	2.21	1.19	4.33	0.15	0.03
31	0.00	1.42	-0.02	2.33	-0.02	-0.09
32	0.00	0.68	-0.32	1.54	0.16	0.05
33	0.00	0.75	0.35	1.36	-0.06	-0.15
34	0.00	1.98	0.56	3.51	-0.03	-0.04
35	0.00	1.91	0.54	3.25	0.08	-0.05
36	0.00	1.44	1.40	1.42	0.24	0.08
37	0.00	0.84	0.53	1.33	0.14	0.01
38	0.00	0.97	0.57	2.17	-0.09	-0.14
39	0.00	0.53	1.33	1.47	-0.60	-0.23
40	0.00	0.75	-0.20	0.57	0.14	-0.11
41	0.00	0.63	0.43	2.48	-0.08	-0.31
42	0.00	0.31	0.41	-0.04	-0.40	-0.14
43	0.00	0.74	1.41	1.14	-0.19	-0.18

Note. a_1 = slope for hedonic benefits, a_2 = slope for nicotine dependence, $\gamma_1 \ldots \gamma_4$ = item intercepts.

Table 3: Item parameters for ERS and MRS model with orthogonal response style factors

Item	a_1	a_2	a_3	a_4	γ_1	γ_2	γ_3	γ_4
1	1.39	0.00	1.48	0.71	0.01	2.92	-0.01	0.03
2	1.47	0.00	1.73	0.67	-0.31	3.73	-0.19	-0.14
3	1.09	0.00	1.20	0.62	-0.52	1.37	-0.08	-0.13
4	1.57	0.00	1.51	0.56	0.32	3.46	0.11	0.14
5	0.91	0.00	1.14	0.54	-0.59	1.15	-0.11	-0.25
6	1.24	0.00	1.33	0.54	-0.41	2.57	0.03	-0.01
7	1.09	0.00	1.15	0.66	-0.61	1.78	-0.12	-0.07
8	1.07	0.00	1.35	0.64	0.75	2.37	0.06	0.11
9	0.81	0.00	1.06	0.68	0.25	1.17	0.05	-0.03
10	1.02	0.00	1.03	0.56	-0.55	1.64	-0.12	-0.11
11	1.67	0.00	1.61	0.59	0.24	3.08	0.01	0.09
12	1.27	0.00	1.37	0.71	-0.04	2.52	0.03	0.18
13	1.03	0.00	1.30	0.69	-0.19	1.87	-0.06	-0.01
14	1.47	0.00	1.23	0.75	1.57	3.25	0.08	0.14
15	1.64	0.00	1.57	0.66	0.43	3.12	-0.06	0.02
16	0.82	0.00	0.93	0.60	-0.66	1.12	0.10	-0.09
17	0.00	2.02	1.28	0.43	0.09	2.81	-0.02	0.03
18	0.00	2.16	1.60	0.66	1.38	2.89	0.11	0.16
19	0.00	2.15	1.46	0.58	1.23	3.47	0.08	0.15
20	0.00	2.51	1.54	0.53	0.73	3.69	0.25	0.22
21	0.00	0.80	1.13	0.47	0.49	1.55	0.01	-0.02
22	0.00	2.30	1.50	0.61	0.79	4.24	0.18	0.23
23	0.00	2.40	1.47	0.55	-0.35	4.43	0.17	0.04
24	0.00	1.64	1.36	0.37	-0.35	2.70	-0.08	-0.03
25	0.00	0.70	1.13	0.44	-0.26	1.93	-0.19	-0.01
26	0.00	1.14	0.99	0.42	-0.57	2.54	0.14	0.02
27	0.00	1.59	1.28	0.63	0.33	2.58	0.08	0.10
28	0.00	1.44	1.19	0.49	0.30	2.09	0.25	0.16
29	0.00	1.58	1.26	0.52	0.43	2.59	0.12	-0.02
30	0.00	2.07	1.65	0.58	1.13	4.60	0.17	0.06
31	0.00	1.45	1.53	0.61	-0.07	2.78	-0.04	-0.04
32	0.00	0.70	0.75	0.53	-0.35	1.71	0.14	0.10
33	0.00	0.73	0.90	0.34	0.34	1.50	-0.07	-0.13
34	0.00	1.89	1.55	0.56	0.51	3.82	-0.03	-0.01
35	0.00	1.85	1.63	0.65	0.50	3.60	0.09	-0.01
36	0.00	1.38	0.72	0.46	1.38	1.37	0.23	0.09
37	0.00	0.79	0.94	0.41	0.51	1.38	0.14	0.04
38	0.00	0.92	1.19	0.20	0.54	2.38	-0.11	-0.10
39	0.00	0.54	0.77	0.58	1.34	1.51	-0.60	-0.17
40	0.00	0.76	0.84	0.45	-0.22	0.65	0.13	-0.08
41	0.00	0.57	1.18	0.28	0.40	2.83	-0.10	-0.24
42	0.00	0.29	0.61	0.32	0.40	-0.08	-0.42	-0.12
43	0.00	0.70	0.76	0.31	1.38	1.04	-0.20	-0.18

Note. a_1 = slope for hedonic benefits, a_2 = slope for nicotine dependence, a_3 = slope for ERS, a_4 = slope for MRS, $\gamma_1 \ldots \gamma_4$ = item intercepts.

Table 4: Item parameters for ERS and MRS model with correlated response style factors

Item	a_1	a_2	a_3	a_4	γ_1	γ_2	γ_3	γ_4
1	1.41	0.00	1.51	0.72	-0.09	2.96	-0.09	0.04
2	1.49	0.00	1.74	0.78	-0.43	3.78	-0.28	-0.13
3	1.14	0.00	1.20	0.60	-0.60	1.47	-0.14	-0.11
4	1.58	0.00	1.55	0.60	0.21	3.46	0.02	0.15
5	0.96	0.00	1.15	0.52	-0.67	1.26	-0.17	-0.24
6	1.28	0.00	1.34	0.58	-0.50	2.65	-0.04	0.00
7	1.13	0.00	1.15	0.67	-0.69	1.89	-0.18	-0.05
8	1.05	0.00	1.40	0.58	0.65	2.30	-0.02	0.12
9	0.81	0.00	1.08	0.65	0.19	1.16	-0.00	-0.02
10	1.06	0.00	1.07	0.55	-0.63	1.76	-0.18	-0.09
11	1.66	0.00	1.70	0.58	0.12	3.09	-0.08	0.10
12	1.29	0.00	1.42	0.69	-0.13	2.57	-0.04	0.20
13	1.05	0.00	1.34	0.67	-0.27	1.94	-0.12	0.01
14	1.42	0.00	1.25	0.72	1.45	3.11	0.00	0.15
15	1.61	0.00	1.63	0.65	0.31	3.09	-0.14	0.03
16	0.87	0.00	0.96	0.54	-0.73	1.25	0.05	-0.08
17	0.00	2.04	1.28	0.43	0.02	2.85	-0.05	0.04
18	0.00	2.12	1.60	0.67	1.28	2.82	0.07	0.18
19	0.00	2.13	1.44	0.58	1.15	3.45	0.04	0.16
20	0.00	2.50	1.55	0.57	0.64	3.68	0.22	0.23
21	0.00	0.79	1.15	0.43	0.44	1.55	-0.02	-0.01
22	0.00	2.29	1.51	0.60	0.71	4.24	0.15	0.24
23	0.00	2.44	1.46	0.63	-0.44	4.49	0.14	0.06
24	0.00	1.67	1.36	0.38	-0.42	2.76	-0.12	-0.02
25	0.00	0.72	1.11	0.43	-0.29	1.96	-0.22	0.00
26	0.00	1.17	0.96	0.40	-0.62	2.59	0.11	0.02
27	0.00	1.59	1.27	0.64	0.26	2.58	0.04	0.11
28	0.00	1.44	1.21	0.48	0.24	2.10	0.21	0.17
29	0.00	1.58	1.27	0.52	0.37	2.60	0.08	-0.01
30	0.00	2.05	1.68	0.64	1.04	4.57	0.11	0.07
31	0.00	1.47	1.53	0.64	-0.14	2.82	-0.09	-0.02
32	0.00	0.71	0.72	0.51	-0.38	1.74	0.12	0.10
33	0.00	0.72	0.90	0.32	0.30	1.48	-0.10	-0.12
34	0.00	1.88	1.57	0.58	0.43	3.81	-0.08	-0.00
35	0.00	1.85	1.64	0.70	0.42	3.61	0.04	0.00
36	0.00	1.33	0.73	0.43	1.32	1.30	0.21	0.10
37	0.00	0.77	0.95	0.39	0.47	1.35	0.10	0.05
38	0.00	0.90	1.23	0.20	0.49	2.36	-0.15	-0.10
39	0.00	0.50	0.77	0.54	1.28	1.42	-0.63	-0.16
40	0.00	0.77	0.81	0.45	-0.25	0.67	0.11	-0.07
41	0.00	0.56	1.19	0.30	0.36	2.84	-0.13	-0.23
42	0.00	0.29	0.64	0.26	0.38	-0.08	-0.43	-0.12
43	0.00	0.66	0.78	0.28	1.33	0.96	-0.22	-0.17

Note. a_1 = slope for hedonic benefits, a_2 = slope for nicotine dependence, a_3 = slope for ERS, a_4 = slope for MRS, $\gamma_1 \ldots \gamma_4$ = item intercepts.

Table 5: Item parameters for ERS and ARS data generating models

Item	a_1	a_2	a_3	a_4	γ_1	γ_2	γ_3	γ_4
1	1.46	0.00	1.50	0.73	-0.05	3.04	-0.05	0.01
2	1.50	0.00	1.70	0.93	-0.39	3.79	-0.23	-0.16
3	1.12	0.00	1.16	0.87	-0.59	1.45	-0.10	-0.15
4	1.61	0.00	1.54	0.80	0.25	3.52	0.07	0.13
5	1.03	0.00	1.14	0.39	-0.67	1.34	-0.15	-0.26
6	1.29	0.00	1.30	0.61	-0.47	2.66	-0.00	-0.02
7	1.15	0.00	1.14	0.74	-0.68	1.92	-0.15	-0.09
8	1.01	0.00	1.43	0.75	0.66	2.27	-0.00	0.09
9	0.81	0.00	1.09	0.54	0.21	1.19	0.01	-0.07
10	0.98	0.00	1.02	1.08	-0.61	1.66	-0.11	-0.15
11	1.49	0.00	1.64	1.55	0.13	2.85	0.01	0.06
12	1.16	0.00	1.41	1.60	-0.14	2.41	0.04	0.12
13	0.95	0.00	1.33	1.18	-0.27	1.83	-0.07	-0.06
14	1.28	0.00	1.30	1.14	1.41	2.93	0.03	0.12
15	1.47	0.00	1.60	1.52	0.31	2.89	-0.06	-0.01
16	0.79	0.00	0.92	1.22	-0.74	1.18	0.12	-0.17
17	0.00	2.10	1.24	0.73	0.07	2.89	-0.02	0.03
18	0.00	2.21	1.66	0.65	1.37	2.98	0.10	0.13
19	0.00	2.18	1.47	0.70	1.21	3.53	0.07	0.14
20	0.00	2.51	1.56	0.94	0.69	3.69	0.25	0.22
21	0.00	0.80	1.15	0.56	0.45	1.55	-0.02	-0.03
22	0.00	2.31	1.50	1.01	0.75	4.25	0.18	0.23
23	0.00	2.45	1.41	0.79	-0.38	4.49	0.17	0.05
24	0.00	1.71	1.30	0.49	-0.38	2.80	-0.09	-0.03
25	0.00	0.71	1.12	0.58	-0.29	1.96	-0.22	-0.02
26	0.00	1.14	0.93	1.00	-0.61	2.52	0.15	-0.01
27	0.00	1.60	1.26	0.94	0.29	2.59	0.08	0.09
28	0.00	1.40	1.19	1.02	0.25	2.02	0.24	0.16
29	0.00	1.55	1.26	0.96	0.39	2.54	0.11	-0.02
30	0.00	2.09	1.72	0.84	1.10	4.67	0.16	0.06
31	0.00	1.48	1.52	0.90	-0.11	2.82	-0.06	-0.05
32	0.00	0.70	0.72	1.07	-0.41	1.74	0.16	0.05
33	0.00	0.72	0.91	0.50	0.31	1.48	-0.09	-0.13
34	0.00	1.89	1.55	0.89	0.47	3.81	-0.04	-0.01
35	0.00	1.93	1.68	0.74	0.48	3.75	0.07	-0.01
36	0.00	1.30	0.78	0.81	1.31	1.27	0.22	0.08
37	0.00	0.77	0.94	0.83	0.47	1.32	0.10	0.04
38	0.00	0.89	1.22	0.61	0.49	2.33	-0.14	-0.10
39	0.00	0.49	0.78	0.77	1.28	1.36	-0.67	-0.19
40	0.00	0.76	0.80	0.76	-0.25	0.66	0.12	-0.09
41	0.00	0.57	1.17	0.77	0.35	2.82	-0.12	-0.25
42	0.00	0.28	0.65	0.36	0.38	-0.09	-0.44	-0.13
43	0.00	0.66	0.80	0.50	1.34	0.94	-0.23	-0.18

Note. a_1 = slope for hedonic benefits, a_2 = slope for nicotine dependence, a_3 = slope for ERS, a_4 = slope for ARS, $\gamma_1 \ldots \gamma_4$ = item intercepts.

Table 6: Factor correlation matrix for GPC data generating model

	HB	ND
HB	1.00	0.49
ND	0.49	1.00

Note. HB = Hedonic benefits, ND = Nicotine dependence.

Table 7: Factor correlation matrix for ERS and MRS model with orthogonal response style factors

	HB	ND	ERS	MRS
HB	1.00	0.47	0.00	0.00
ND	0.47	1.00	0.00	0.00
ERS	0.00	0.00	1.00	0.00
MRS	0.00	0.00	0.00	1.00

Note. HB = Hedonic benefits, ND = Nicotine dependence, ERS = Extreme response style, MRS = Midpoint response style.

Table 8: Factor correlation matrix for ERS and MRS model with correlated response style factors

	HB	ND	ERS	MRS
HB	1.00	0.46	0.31	-0.12
ND	0.46	1.00	0.17	-0.09
ERS	0.31	0.17	1.00	-0.27
MRS	-0.12	-0.09	-0.27	1.00

Note. HB = Hedonic benefits, ND = Nicotine dependence, ERS = Extreme response style, MRS = Midpoint response style.

Table 9: Factor correlation matrix for ERS and ARS model with correlated response style factors

	HB	ND	ERS	ARS
HB	1.00	0.46	0.23	0.01
ND	0.46	1.00	0.11	-0.09
ERS	0.23	0.11	1.00	0.30
ARS	0.01	-0.09	0.30	1.00

Note. HB = Hedonic benefits, ND = Nicotine dependence, ERS = Extreme response style, ARS = Acquiescent response style.

Table 10: Factor correlation matrix for ERS and ARS model with correlated response style factors and low HB-ND correlation

	HB	ND	ERS	ARS
HB	1.00	0.05	0.23	0.01
ND	0.05	1.00	0.11	-0.09
ERS	0.23	0.11	1.00	0.30
ARS	0.01	-0.09	0.30	1.00

Note. HB = Hedonic benefits, ND = Nicotine dependence, ERS = Extreme response style, ARS = Acquiescent response style.