Supplementary Materials

Contents

1 Code for the Illustrative Example **2**

1.1 Data preparation . . . 2

1.2 Multi-group MCAA (4 groups) . . . 4

1.3 Multi-group MCAA (two groups) . . . 6

1.3.1 White vs. non-White . . . 6

1.3.2 Hispanic vs. non-Hispanic . . . 6

1.4 Supporting analyses . . . 7

1.4.1 Latent correlation estimates . . . 7

1.4.2 Census example: impact of measurement noninvariance . . . 8

1.4.3 Sensitivity analysis using alignment optimization . . . 8

2 Tables **11**

3 Figures **21**

4 Differential item weights **27**

1 Code for the Illustrative Example

```
# Load libraries
library(here)
library(dplyr)
library(kableExtra)
library(lavaan)
# install.packages("remotes")
# remotes::install_github("marklhc/pinsearch")
library(pinsearch)
#remotes::install_github("mmm-usc/unbiasr", ref = "pkg")
library(unbiasr)
source('code/helper_functions_for_tables.R')
```

1.1 Data preparation

```
# Add Health Wave I, 1994-1995 public use data
# https://dataverse.unc.edu/dataset.xhtml?persistentId=doi:10.15139/S3/11900
# Put the downloaded files in a folder named 'dataverse_files' in the working directory

dt <- haven::read_sas(here("dataverse_files/w1inhome.sas7bdat"))
# See 'pubwgt1.pdf' for more info on the public use weights
weights <- haven::read_sas(here("dataverse_files/w1weight.sas7bdat"))

CESD_vars <- c(do.call(paste0, expand.grid("H1FS", 1:19)))
cesd_labs <-
  c("bothered", "appetite", "blues", "good", "mind", "depressed", "tired", "hopeful",
    "failure", "fearful", "happy", "talk", "lonely", "unfriendly", "enjoyed", "sad",
    "dislike", "getstarted", "life")
CESD_n <- c(do.call(paste0, expand.grid("cesd", c(1:19))))

# create an ethnicity variable with levels "Black",
# "White", "Hispanic", and "Asian" using Boolean operators
dt$ethnicity <- "Other"
dt[dt$H1GI6B==1 & dt$H1GI6A==0 & dt$H1GI6D==0 & dt$H1GI4==0, "ethnicity"] <-"Black"
dt[dt$H1GI6B==0 & dt$H1GI6A==1 & dt$H1GI6D==0 & dt$H1GI4==0, "ethnicity"] <-"White"
dt[dt$H1GI6B==0 & dt$H1GI6A==0 & dt$H1GI6D==1 & dt$H1GI4==0, "ethnicity"] <-"Asian"
dt[dt$H1GI6B==0 & dt$H1GI6A==0 & dt$H1GI6D==0 & dt$H1GI4==1, "ethnicity"] <-"Hispanic"

dt <- dt %>% select(AID, BIO_SEX, ethnicity, all_of(CESD_vars))
colnames(dt) <- c("AID", "sex", "ethnicity", CESD_n)

# Select the subset with complete CESD scores (filter rows where no item was
# 'refused' or 'don't know') and filter out the Other groups
dt_sub <- dt %>%
  filter_at(vars(starts_with('cesd')), all_vars(. != 6 & . != 8)) %>% # removes 47 obs.
  filter(ethnicity != "Other")
colnames(dt_sub) <- c("AID", "sex", "ethnicity", cesd_labs) # 5848

# Merge with weights
dat <- left_join(dt_sub, weights, "AID") # 5848 x 24
```

```
# Recode positively worded items to achieve a 'lack of' interpretation for consistency
dat$happy <- recode(dat$happy, `0` = 3, `1` = 2, `2` = 1, `3` = 0)
dat$good <- recode(dat$good, `0` = 3, `1` = 2, `2` = 1, `3` = 0)
dat$hopeful <- recode(dat$hopeful, `0` = 3, `1` = 2, `2` = 1, `3` = 0)
dat$enjoyed <- recode(dat$enjoyed, `0` = 3, `1` = 2, `2` = 1, `3` = 0)

labs <- paste0("i", 1:21)
labs[7] <- "i7r"; labs[20] <- "i20r"
labs <- labs[-c(11, 17)]
colnames(dat) <- c("AID", "sex", "ethnicity", labs, "CLUSTER2", "GSWGT1")

# Define binary ethnicity variables for the examples where groups are collapsed
dat$White <- "non-White"
dat$White[dat$ethnicity=="White"] <- "White"
dat$Hispanic <- "non-Hispanic"
dat$Hispanic[dat$ethnicity=="Hispanic"] <- "Hispanic"
```

```
dat <- readRDS("addhealth_w1_dat.rds")
```

```
# Define miscellaneous variables
item_factor <- paste0(
  labs, "-",  c("F1", "F1", "F2", "F3", "F1", "F2", "F1", "F3", "F2", "F2", "F3", "F1",
       "F2", "F4", "F3", "F2", "F4", "F1", "F2" ))
groups_wbha <- c("White", "Black", "Hispanic", "Asian")
fitv <- c("chisq", "npar", "df", "pvalue", "rmsea.robust", "cfi.robust", "tli.robust",
 "srmr")
cols <- c("#FF5733", "springgreen2", "magenta", "#33C3Ff")
colsW <- c(cols[1], "black")
colsH <- c("black", cols[3])
names(colsW) <- c("White", "non-White")
```

1.2 Multi-group MCAA (4 groups)

```r
# Define the configural model string
m1 <- '
 F1 =~ i1 + i2 + i5 + i7r + i13 + i20r
 F2 =~ i3 + i6 + i9 + i10 + i14 + i18 + i21
 F3 =~ i4 + i8 + i12 + i16
 F4 =~ i15 + i19
'
```

```r
# Fit the configural model
configural_fit <- cfa(
  model = m1, data = dat, group = "ethnicity", estimator = "MLR", missing = "fiml",
  se = "robust.mlr", sampling.weights = "GSWGT1", group.label = groups_wbha)
```

```r
saveRDS(configural_fit, "rds/configural_fit_4g.rds")
```

```r
# Use pinsearch to automate the partial invariance model search process
pin_out <- pinSearch(
  config_mod = m1, data = dat, group = "ethnicity", estimator = "MLR", missing = "fiml",
  se = "robust.mlr", sampling.weights = "GSWGT1", group.label = groups_wbha,
  type = "residuals")
partial_fit <- pin_out[[1]]
```

```r
saveRDS(pin_out, "rds/pin_out.rds")
```

```r
# Equivalently, arguments used in repeated function calls can be stored in lists and
# passed to do.call() as below to minimize code repetition
args_cfa <-
  list(data = dat, group = "ethnicity", estimator = "MLR", missing = "fiml",
       se = "robust.mlr", sampling.weights = "GSWGT1", group.label = groups_wbha)
```

```r
configural_fit <- do.call(cfa, c(m1, args_cfa))
```

```r
args_pin <- c(args_cfa, type = "residuals")
```

```r
pin_out <- do.call(pinSearch, c(m1, args_pin))
partial_fit <- pin_out[[1]]
```

```r
# Extract model fit indices
config_fit_stat <- fitMeasures(configural_fit, fitv)
partial_fit_stat <- fitMeasures(partial_fit, fitv)

# Examine fMACS effect size indices reflecting item-level bias across groups
es <- pinsearch::pin_effsize(partial_fit)
sort(round(es[,item_factor[item_factor %in% colnames(es)]], 3), decreasing = TRUE)
```

```
##   i9-F2 i20r-F1  i13-F1   i4-F3   i8-F3   i5-F1   i2-F1  i21-F2   i6-F2  i10-F2
##   0.076   0.069   0.061   0.059   0.057   0.055   0.053   0.051   0.042   0.040
##  i18-F2  i16-F3
##   0.034   0.030
```

```
# Compute mixing proportions
weightsums <- tapply(dat$GSWGT1, dat$ethnicity, sum)
mixp_unord <- weightsums / sum(weightsums)
mixp <- matrix(c(mixp_unord[4], mixp_unord[2], mixp_unord[3], mixp_unord[1]), ncol = 4)
colnames(mixp) <- groups_wbha
```

```
# Latent factor weights: place weights proportional to item counts
lw <- c(6, 7, 4, 2) / sum(c(6, 7, 4, 2))

# Call PartInv() using the cfa() output object
out_4g <- PartInv(
  cfa_fit = partial_fit, pmix = mixp, weights_latent = lw, cut_z = 15.2,
  plot_contour = TRUE, show_mi_result = TRUE, custom_colors = cols)
```

```
# Overall ES
to_asin <- function(x) 2 * asin(sqrt(x))
get_multi_h <- function(pvec) {
  asin_pvec <- to_asin(pvec)
  2 * sqrt(mean((asin_pvec - mean(asin_pvec))^2))
}
multi_h_4g <- apply(out_4g$summary[, c(1, 5:7)], MARGIN = 1, FUN = get_multi_h)
multi_h_4g
```

```
##   A (true positive)  B (false positive)   C (true negative)  D (false negative)
##          0.03413632          0.07600809          0.03814189          0.06715411
## Proportion selected       Success ratio         Sensitivity         Specificity
##          0.06611096          0.10983606          0.14674332          0.08825685
```

```
# Alternatively, call PartInv() using lists of parameter estimates
est <- unnest_list(lavInspect(partial_fit, "est"))
out_4g <- PartInv(
  alpha = est$alpha, psi = est$psi, lambda = est$lambda, nu = est$nu,
  theta = est$theta, pmix = mixp, weights_latent = lw, cut_z = 15.2,
  plot_contour = TRUE, show_mi_result = TRUE, custom_colors = cols,
  labels = groups_wbha)
```

```
# Plot CAI and Adverse Impact Ratio across a range of thresholds
plot_CAI_across_range(
  x = out_4g, custom_colors = cols, cutoffs_from = 10, cutoffs_to = 30,
  add_vertical_threshold_at = 15.2)
```

1.3 Multi-group MCAA (two groups)

1.3.1 White vs. non-White

```
# Update stored values, call do.call() to obtain configural and partial invariance models
args_cfa_W <- args_cfa
args_pin_W <- args_pin
args_pin_W$group <- args_cfa_W$group <- "White"
args_pin_W$group.label <- args_cfa_W$group.label <- c("non-White", "White")

configural_fit_W <- do.call(cfa, c(m1, args_cfa_W))
```

```
pin_out_W <- do.call(pinSearch, c(m1, args_pin_W))
partial_fit_W <- pin_out_W[[1]]
```

```
saveRDS(pin_out_W, "rds/pin_out_W.rds")
```

```
# Examine fMACS effect size indices reflecting item-level bias across groups
es_W <- pinsearch::pin_effsize(partial_fit_W)
sort(round(es_W[, item_factor[item_factor %in% colnames(es_W)]], 3), decreasing = TRUE)
```

```
##  i9-F2 i13-F1 i21-F2 i10-F2 i14-F2 i7r-F1  i2-F1 i15-F4
##  0.246  0.182  0.168  0.114  0.114  0.097  0.092  0.087
```

```
config_fit_stat_W <- fitMeasures(configural_fit_W, fitv)
partial_fit_stat_W <- fitMeasures(partial_fit_W, fitv)

weightsums_W <- tapply(dat$GSWGT1, dat$White, sum)
mixp_W <- weightsums_W / sum(weightsums_W)
```

```
out_2g_W <- PartInv(
  cfa_fit = partial_fit_W, pmix = mixp_W, weights_latent = lw,
  cut_z = 15.2, plot_contour = TRUE, show_mi_result = TRUE,
  custom_colors = colsW, reference = "White")
```

```
plot_CAI_across_range(
  x = out_2g_W, cutoffs_from = 10, cutoffs_to = 30,
  add_vertical_threshold_at = 15.2, custom_colors = colsW)
```

1.3.2 Hispanic vs. non-Hispanic

```
# Update stored values, call do.call() to obtain configural and partial invariance models
args_cfa_H <- args_cfa
args_pin_H <- args_pin
args_pin_H$group <- args_cfa_H$group <- "Hispanic"
args_pin_H$group.label <- args_cfa_H$group.label <- c("non-Hispanic", "Hispanic")

configural_fit_H <- do.call(cfa, c(m1, args_cfa_H))
```

```
pin_out_H <- do.call(pinSearch, c(m1, args_pin_H))
partial_fit_H <- pin_out_H[[1]]
```

```
saveRDS(pin_out_H, "rds/pin_out_H.rds")
```

```
# Examine fMACS effect size indices reflecting item-level bias across groups
es_H <- pinsearch::pin_effsize(pin_out_H[[1]])
sort(round(es_H[, item_factor[item_factor %in% colnames(es_H)]], 3), decreasing = TRUE)
```

```
##  i4-F3  i9-F2 i21-F2  i1-F1  i8-F3 i10-F2
##  0.177  0.148  0.140  0.129  0.126  0.116
```

```
config_fit_stat_H <- fitMeasures(configural_fit_H, fitv)
partial_fit_stat_H <- fitMeasures(partial_fit_H, fitv)

weightsums_H <- tapply(dat$GSWGT1, dat$Hispanic, sum)
mixp_unord <- weightsums_H / sum(weightsums_H)
mixp_H <- matrix(c(mixp_unord[2], mixp_unord[1]), ncol = 2)
colnames(mixp_H) <- c("non-Hispanic", "Hispanic")
```

```
out_2g_H <- PartInv(
  cfa_fit = partial_fit_H, pmix = mixp_H, weights_latent = lw, cut_z = 15.2,
  plot_contour = TRUE, show_mi_result = TRUE, custom_colors = colsH,
  reference = "non-Hispanic")
```

```
plot_CAI_across_range(
  x = out_2g_H, cutoffs_from = 10, cutoffs_to = 30, by = 0.1,
  add_vertical_threshold_at = 15.2, custom_colors = colsH)
```

1.4 Supporting analyses

1.4.1 Latent correlation estimates

```
# Extract list of parameter estimates
est <- unnest_list(lavInspect(partial_fit, "est"))
# Compute correlation coefficients for each group (%o%: outer product)
lapply(est$psi, function(x) x / (sqrt(diag(x)) %o% sqrt(diag(x))))
```

```
## [[1]]
##       F1    F2    F3    F4
## F1 1.000
## F2 0.857 1.000
## F3 0.554 0.609 1.000
## F4 0.526 0.546 0.393 1.000
##
## [[2]]
##       F1    F2    F3    F4
## F1 1.000
## F2 0.863 1.000
## F3 0.364 0.483 1.000
## F4 0.642 0.707 0.272 1.000
##
## [[3]]
##       F1    F2    F3    F4
## F1 1.000
## F2 0.847 1.000
## F3 0.546 0.547 1.000
## F4 0.613 0.607 0.344 1.000
##
## [[4]]
##       F1    F2    F3    F4
## F1 1.000
## F2 0.960 1.000
## F3 0.464 0.666 1.000
## F4 0.511 0.586 0.425 1.000
```

1.4.2 Census example: impact of measurement noninvariance

```r
# https://www2.census.gov/programs-surveys/demo/tables/hispanic-origin/1995/1994-cps/sumtab-1.txt

# First, compute approximate number of Hispanic individuals aged 10-19 using
# the percentages reported in Table 1 (rows 5-6, column `Hispanic-origin population`)
hisp10_19 <- 27521 * (9.1 + 8.5) / 100 * 1000

# Compare the proportion selected (PS) in the reference group (White) under partial
# invariance to the expected PS for the Hispanic group under partial invariance, after
# matching the underlying distributions

# PS for the Hispanic group assuming matched latent distributions
PS_H <- out_4g$summary[5, "E_R(Hispanic)"]

# PS for the reference group (benchmark under no measurement bias)
PS_ref <- out_4g$summary[5, "White"]

round(PS_H - PS_ref, 2) # ~3% difference
```

```
## [1] 0.03
```

```r
ceiling(hisp10_19 * (PS_H - PS_ref)) # ~145,889 more individuals screened in
```

```
## [1] 145889
```

1.4.3 Sensitivity analysis using alignment optimization

```r
library(sirt)

# Prepare the data and obtain the configural fit as before
# Extract loadings and intercepts
ests <- inspect(configural_fit, "est")
# Indices of different dimensions
ind_list <- list(1:6, 7:13, 14:17, 18:19)
# Group sizes
grp_size <- inspect(configural_fit, what = "nobs")
# First dimension
lam_fac1 <- lapply(ests, \(x) x$lambda[ind_list[[1]], 1])
lam_fac1 <- do.call(rbind, lam_fac1)
nu_fac1 <- lapply(ests, \(x) x$nu[ind_list[[1]], 1])
nu_fac1 <- do.call(rbind, nu_fac1)
align_fac1 <- invariance.alignment(
  lam_fac1,
  nu_fac1,
  wgt = matrix(sqrt(grp_size), nrow = length(grp_size), ncol = ncol(lam_fac1))
)
fac1_syntax <- paste0(
  "F1 ~~ c(", paste0(align_fac1$pars[, "psi0"]^2, collapse = ", "), ") * F1\n",
  "F1 ~ c(", paste0(align_fac1$pars[, "alpha0"], collapse = ", "), ") * 1\n"
)
# Second dimension
lam_fac2 <- lapply(ests, \(x) x$lambda[ind_list[[2]], 2])
lam_fac2 <- do.call(rbind, lam_fac2)
nu_fac2 <- lapply(ests, \(x) x$nu[ind_list[[2]], 1])
nu_fac2 <- do.call(rbind, nu_fac2)
align_fac2 <- invariance.alignment(
  lam_fac2,
  nu_fac2,
  wgt = matrix(sqrt(grp_size), nrow = length(grp_size), ncol = ncol(lam_fac2))
)
fac2_syntax <- paste0(
  "F2 ~~ c(", paste0(align_fac2$pars[, "psi0"]^2, collapse = ", "), ") * F2\n",
  "F2 ~ c(", paste0(align_fac2$pars[, "alpha0"], collapse = ", "), ") * 1\n"
)
# Third dimension
lam_fac3 <- lapply(ests, \(x) x$lambda[ind_list[[3]], 3])
lam_fac3 <- do.call(rbind, lam_fac3)
nu_fac3 <- lapply(ests, \(x) x$nu[ind_list[[3]], 1])
nu_fac3 <- do.call(rbind, nu_fac3)
align_fac3 <- invariance.alignment(
  lam_fac3,
  nu_fac3,
  wgt = matrix(sqrt(grp_size), nrow = length(grp_size), ncol = ncol(lam_fac3))
)
fac3_syntax <- paste0(
  "F3 ~~ c(", paste0(align_fac3$pars[, "psi0"]^2, collapse = ", "), ") * F3\n",
  "F3 ~ c(", paste0(align_fac3$pars[, "alpha0"], collapse = ", "), ") * 1\n"
)
# Fourth dimension
lam_fac4 <- lapply(ests, \(x) x$lambda[ind_list[[4]], 4])
lam_fac4 <- do.call(rbind, lam_fac4)
nu_fac4 <- lapply(ests, \(x) x$nu[ind_list[[4]], 1])
nu_fac4 <- do.call(rbind, nu_fac4)
```

```r
align_fac4 <- invariance.alignment(
  lam_fac4,
  nu_fac4,
  wgt = matrix(sqrt(grp_size), nrow = length(grp_size), ncol = ncol(lam_fac4))
)
fac4_syntax <- paste0(
  "F4 ~~ c(", paste0(align_fac4$pars[, "psi0"]^2, collapse = ", "), ") * F4\n",
  "F4 ~ c(", paste0(align_fac4$pars[, "alpha0"], collapse = ", "), ") * 1\n"
)
# Rerun CFA with aligned latent means and variances
aligned_fit <- do.call(cfa, c(args_cfa, auto.fix.first = FALSE,
  model = paste0(m1, fac1_syntax, fac2_syntax, fac3_syntax, fac4_syntax)))
# summary(aligned_fit)
```

Examine f_{MACS} effect size indices reflecting which items contribute the most bias.

```r
es_a <- pinsearch::pin_effsize(aligned_fit)
sort(round(es_a[, item_factor[item_factor %in% colnames(es_a)]], 3),
     decreasing = TRUE)
```

```
##  i13-F1   i9-F2   i1-F1  i19-F4   i4-F3  i7r-F1   i8-F3  i21-F2  i12-F3  i14-F2
##   0.122   0.102   0.096   0.093   0.092   0.088   0.086   0.077   0.068   0.068
##   i3-F2  i15-F4 i20r-F1  i10-F2   i6-F2  i16-F3   i5-F1  i18-F2   i2-F1
##   0.063   0.062   0.062   0.053   0.052   0.052   0.049   0.040   0.019
```

2 Tables

Table 1: Model fit indices for the configural and partial invariance models (two groups).

Model	χ^2	npar	df	p	RMSEA	CFI	TLI	SRMR
White vs. Non-White								
Configural	2554.74	126	292	0.00	0.05	0.93	0.92	0.04
Partial	2588.88	106	312	0.00	0.05	0.93	0.93	0.04
Non-Hispanic vs. Hispanic								
Configural	2402.46	126	292	0.00	0.05	0.94	0.93	0.03
Partial	2461.14	91	327	0.00	0.04	0.94	0.94	0.04

Note: CFI = Robust Comparative Fit Index. TLI = Robust Tucker-Lewis Index. RMSEA = Robust Root Mean Square Error of Approximation. SRMR = Square Root Mean Residual. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Table 2: Non-invariant measurement parameters (two groups).

	White vs. non-White				non-Hispanic vs. Hispanic			
Item	λ	ν	θ	d_{MACS}	λ	ν	θ	d_{MACS}
1. Somatic Complaints								
i1 (bothered)	-	-	*	-	-	*	*	.129
i2 (appetite)	*	*	*	.092	-	-	-	-
i5 (mind)	-	-	-	-	-	-	*	-
i7r (tired)	-	*	*	.097	-	-	-	-
i13 (talk)	-	*	*	.182	-	-	*	-
i20r (getstarted)	-	-	-	-	-	-	-	
2. Negative Affect								
i3 (blues)	-	-	*	-	-	-	-	-
i6 (depressed)	-	-	*	-	-	-	-	-
i9 (failure)	*	*	*	.246	-	*	*	.148
i10 (fearful)	*	*	*	.114	-	*	*	.116
i14 (lonely)	*	*	*	.114	-	-	-	-
i18 (sad)	-	-	*	-	-	-	-	-
i21 (worthliving)	*	*	*	.168	-	*	*	.140
3. (Lack of) Positive Affect								
i4 (good)	-	-	*	-	-	*	*	.177
i8 (hopeful)	-	-	*	-	-	*	*	.126
i12 (happy)	-	-	*	-	-	-	*	-
i16 (enjoy)	-	-	*	-	-	-	-	-
4. Interpersonal Relations								
i15 (unfriendly)	-	*	-	.087	-	-	-	-
i19 (dislike)	-	-	*	-	-	-	*	-

Note. "-" indicates invariance. λ, ν, and θ refer to factor loadings, intercepts, and uniqueness parameters. d_{MACS} reflects the item-level effect size of measurement noninvariance across two groups. * indicates that the parameter was noninvariant.

Table 3: Factor loading and measurement intercept estimates under partial invariance (2 groups).

	White vs. Non-White				Non-Hispanic vs. Hispanic			
	Loadings		Intercepts		Loadings		Intercepts	
Item	NW	W	NW	W	NH	H	NH	H
1. Somatic Complaints								
i1	0.46	0.46	0.55	0.55	0.41	0.41	0.48	0.39
i2	0.34	0.40	0.50	0.52	0.34	0.34	0.45	0.45
i5	0.51	0.51	0.89	0.89	0.46	0.46	0.80	0.80
i7r	0.41	0.41	0.83	0.76	0.37	0.37	0.71	0.71
i13	0.30	0.30	0.70	0.56	0.28	0.28	0.55	0.55
i20r	0.36	0.36	0.68	0.68	0.33	0.33	0.62	0.62
2. Negative Affect								
i3	0.52	0.52	0.45	0.45	0.48	0.48	0.37	0.37
i6	0.63	0.63	0.60	0.60	0.58	0.58	0.49	0.49
i9	0.40	0.29	0.30	0.23	0.30	0.30	0.19	0.27
i10	0.34	0.27	0.38	0.36	0.27	0.27	0.31	0.38
i14	0.53	0.47	0.56	0.50	0.46	0.46	0.44	0.44
i18	0.51	0.51	0.63	0.63	0.48	0.48	0.55	0.55
i21	0.32	0.26	0.23	0.17	0.26	0.26	0.14	0.21
3. (Lack of) Positive Affect								
i4	0.50	0.50	1.15	1.15	0.49	0.49	1.04	1.21
i8	0.52	0.52	1.25	1.25	0.51	0.51	1.14	1.26
i12	0.59	0.59	0.98	0.98	0.59	0.59	0.86	0.86
i16	0.64	0.64	0.87	0.87	0.64	0.64	0.74	0.74
4. Interpersonal Relations								
i15	0.43	0.43	0.45	0.51	0.40	0.40	0.40	0.40
i19	0.59	0.59	0.53	0.53	0.55	0.55	0.41	0.41

Note: NW, W, NH, and H refer to non-White, White, non-Hispanic, and Hispanic respectively. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Table 4: Latent variance-covariance and mean estimates across groups.

	Latent variance-covariances				Latent means
	1	2	3	4	
non-White					
1. Somatic Complaints	1.00				0.00
2. Negative Affect	0.88	1.00			0.00
3. (Lack of) Positive Affect	0.42	0.52	1.00		0.00
4. Interpersonal Relations	0.63	0.67	0.29	1.00	0.00
White					
1. Somatic Complaints	0.74				-0.21
2. Negative Affect	0.68	0.85			-0.20
3. (Lack of) Positive Affect	0.47	0.56	0.99		-0.24
4. Interpersonal Relations	0.40	0.45	0.35	0.80	-0.27

Table 5: Latent variance-covariance and mean estimates across groups.

	Latent variance-covariances				Latent means
	1	2	3	4	
non-Hispanic					
1. Somatic Complaints	1.00				0.00
2. Negative Affect	0.87	1.00			0.00
3. (Lack of) Positive Affect	0.52	0.59	1.00		0.00
4. Interpersonal Relations	0.56	0.59	0.37	1.00	0.00
Hispanic					
1. Somatic Complaints	1.20				0.25
2. Negative Affect	1.09	1.38			0.21
3. (Lack of) Positive Affect	0.61	0.66	1.10		0.31
4. Interpersonal Relations	0.64	0.68	0.35	0.91	0.14

Table 6: Observed and expected CAI (two groups).

	White vs Non-White		Non-Hispanic vs. Hispanic	
	White	non-White	non-Hispanic	Hispanic
True Positive (TP)				
Strict	0.207	0.307	0.227	0.331
Partial	0.205	0.311	0.227	0.338
Partial (Ef)	-	0.211	-	0.234
False Positive (FP)				
Strict	0.046	0.047	0.047	0.047
Partial	0.042	0.058	0.046	0.058
Partial (Ef)	-	0.058	-	0.059
True Negative (TN)				
Strict	0.703	0.595	0.680	0.572
Partial	0.708	0.583	0.681	0.560
Partial (Ef)	-	0.692	-	0.668
False Negative (FN)				
Strict	0.044	0.052	0.046	0.051
Partial	0.045	0.048	0.047	0.044
Partial (Ef)	-	0.040	-	0.040
Proportion Selected (PS)				
Strict	0.253	0.353	0.274	0.377
Partial	0.247	0.369	0.273	0.396
Partial (Ef)	-	0.268	-	0.293
Success Ratio (SR)				
Strict	0.819	0.867	0.830	0.876
Partial	0.832	0.842	0.833	0.852
Partial (Ef)	-	0.785	-	0.800
Sensitivity (SE)				
Strict	0.825	0.855	0.831	0.866
Partial	0.819	0.867	0.829	0.885
Partial (Ef)	-	0.840	-	0.855
Specificity (SP)				
Strict	0.939	0.927	0.936	0.924
Partial	0.945	0.909	0.937	0.905
Partial (Ef)	-	0.923	-	0.919

Note: Rows labeled "Strict" and "Partial" contain observed classification accuracy indices (CAI), and rows labeled "Partial (Ef)" contain the expected CAI for the focal group(s) if the underlying distributions matched the latent distribution of the reference group. Groups were collapsed to achieve a binary grouping for illustrative purposes.

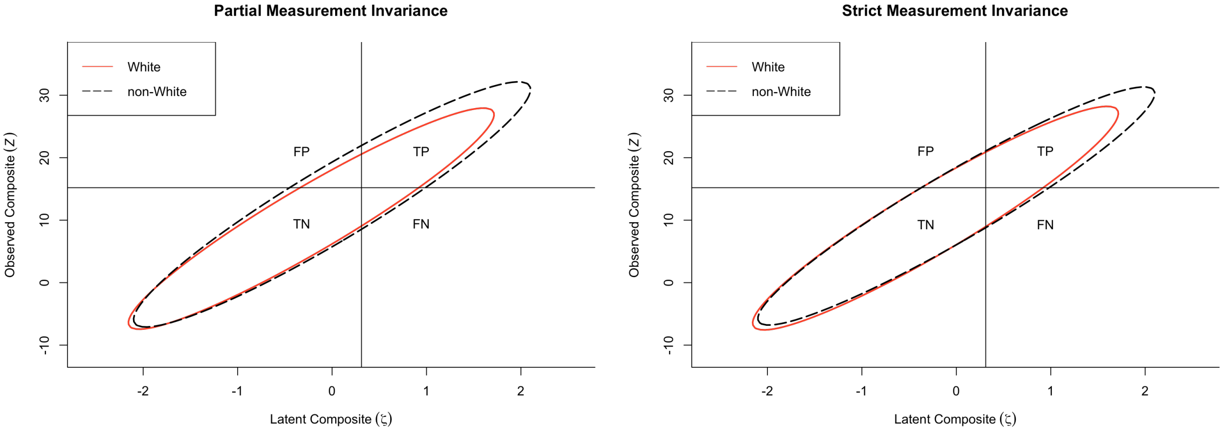
Table 7: Cohen's h effect sizes for discrepancies between observed and expected CAI.

	White vs Non-White	Non-Hispanic vs. Hispanic
	$h_{\text{non-White}}$	h_{Hispanic}
True Positive (TP)	-0.01	-0.02
False Positive (FP)	-0.07	-0.06
True Negative (TN)	0.04	0.03
False Negative (FN)	0.03	0.03
Proportion Selected (PS)	-0.05	-0.04
Success Ratio (SR)	0.12	0.08
Sensitivity (SE)	-0.06	-0.07
Specificity (SP)	0.09	0.07

Note: Expected classification accuracy indices (CAI) are computed under partial invariance assuming that the underlying distribution for the focal group(s) match that of the reference group. Groups were collapsed to achieve a binary grouping for illustrative purposes.

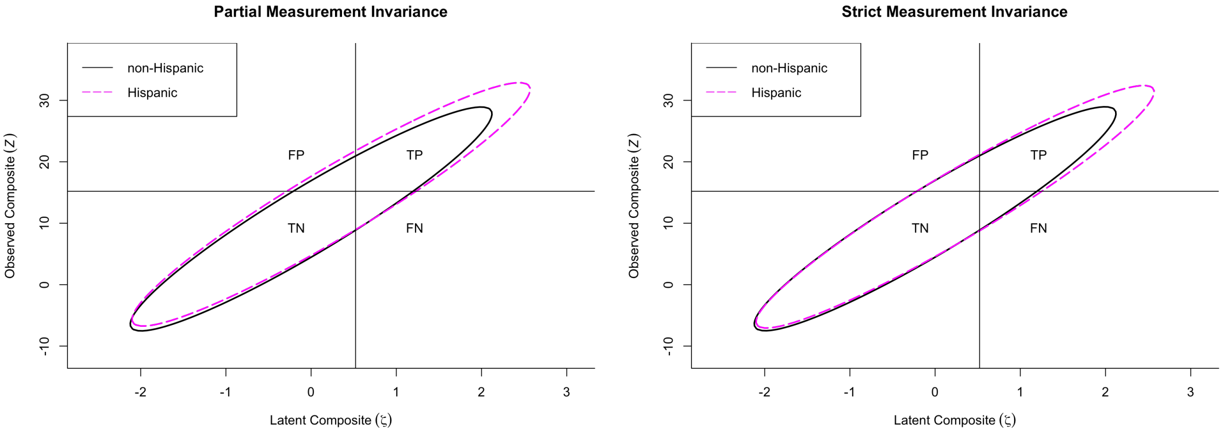
3 Figures

Figure 1: Joint bivariate distributions of observed and latent scores (White, non-White).



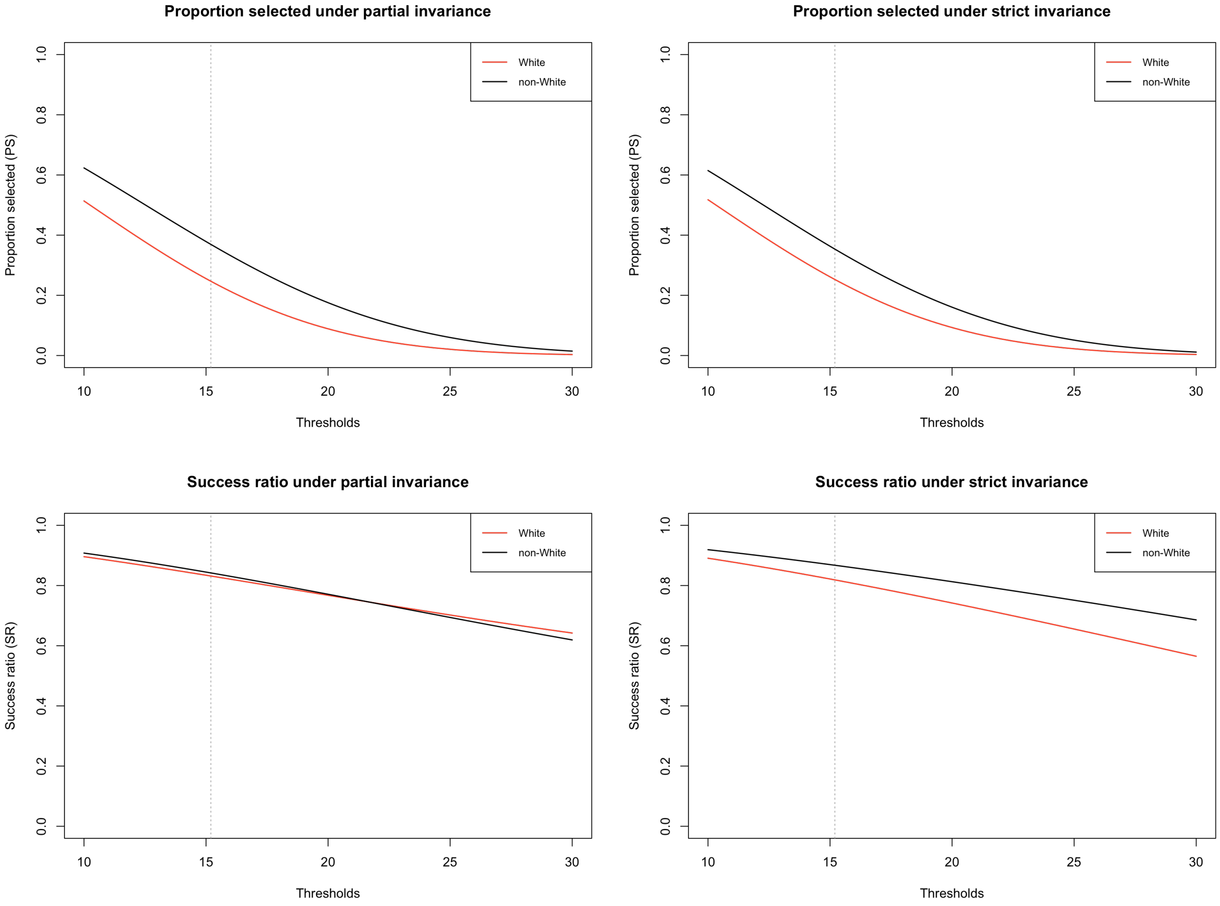
The figures illustrate the joint bivariate distributions of observed and latent scores for each group under partial (left) and strict (right) measurement invariance (two groups). Quadrants TP, FP, TN, and FN correspond to True Positive, False Positive, True Negative, and False Negative rates. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Figure 2: Joint bivariate distributions of observed and latent scores (non-Hispanic, Hispanic).



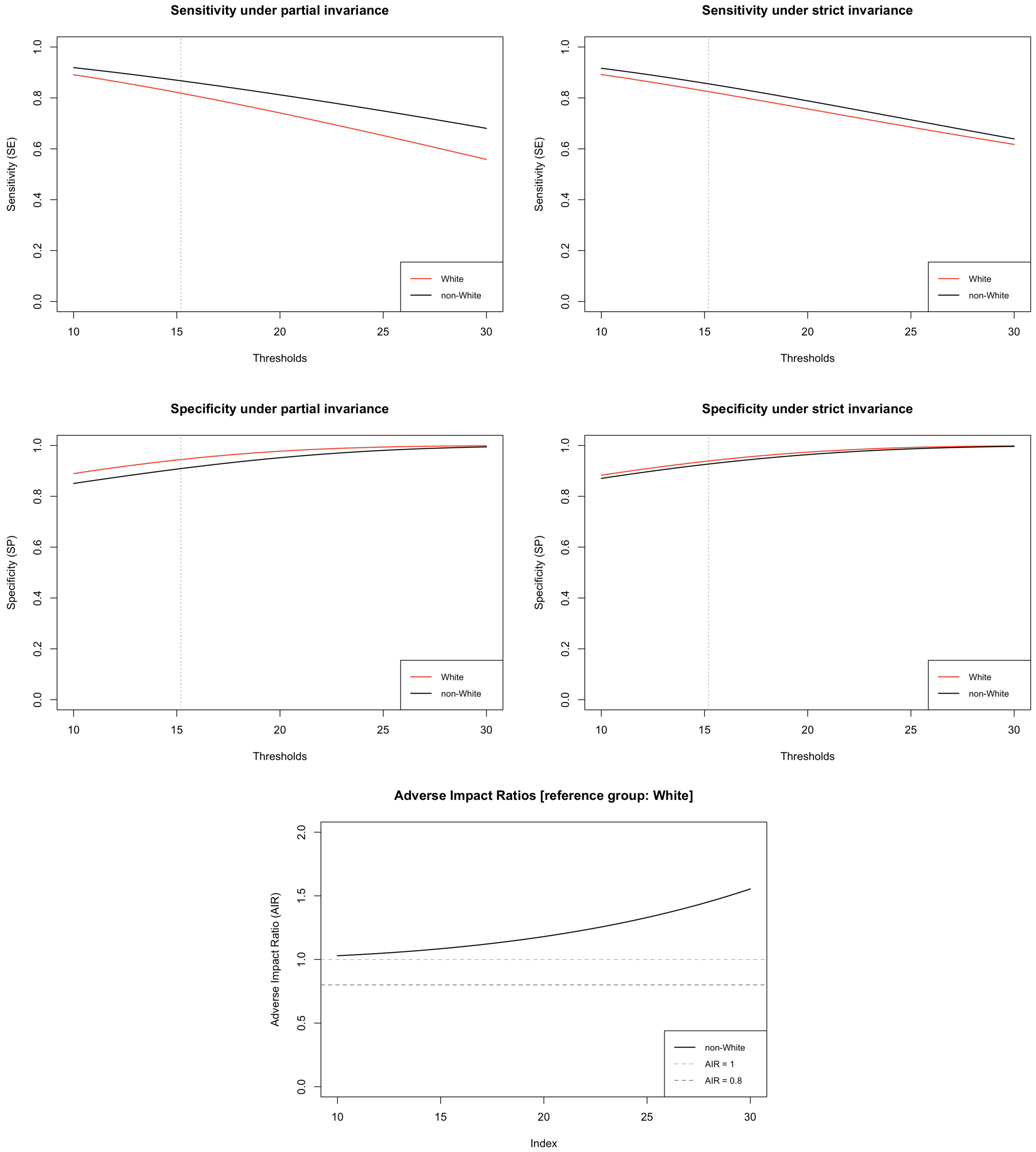
The figures illustrate the joint bivariate distributions of observed and latent scores for each group under partial (left) and strict (right) measurement invariance (two groups). Quadrants TP, FP, TN, and FN correspond to True Positive, False Positive, True Negative, and False Negative rates. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Figure 3: Proportions selected (PS) and success ratios (SR) at different cutoffs (Z_c) by group (White, non-White).



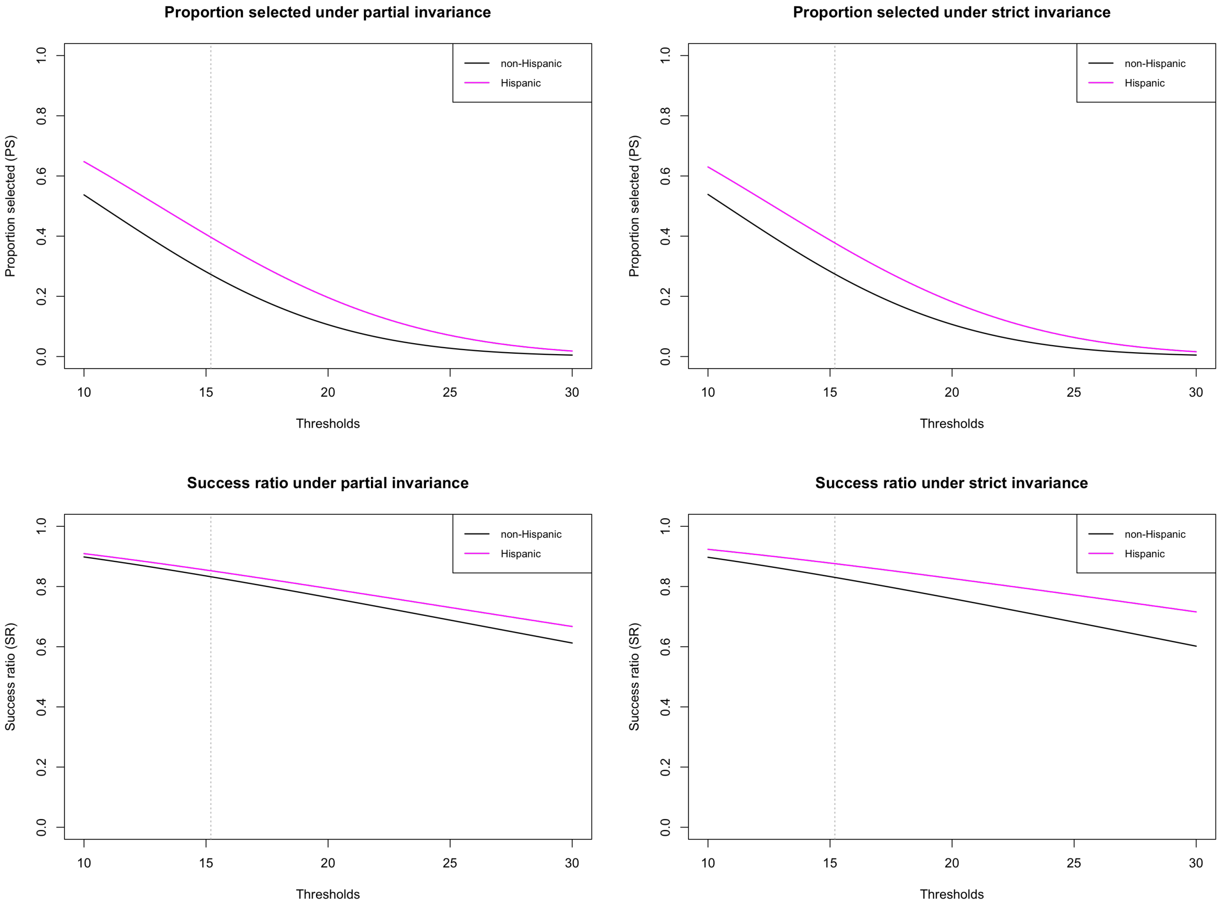
Figures on the left reflect partial MI and the figures on the right indicate strict MI. The vertical dotted line at 15.2 is equivalent to Radloff's (1977) suggested cutoff of 16 on 20 items. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Figure 4: Sensitivity (SE), Specificity (SP), and Adverse Impact ratios (AIR) at different cutoffs (Z_c) by group (White, non-White).



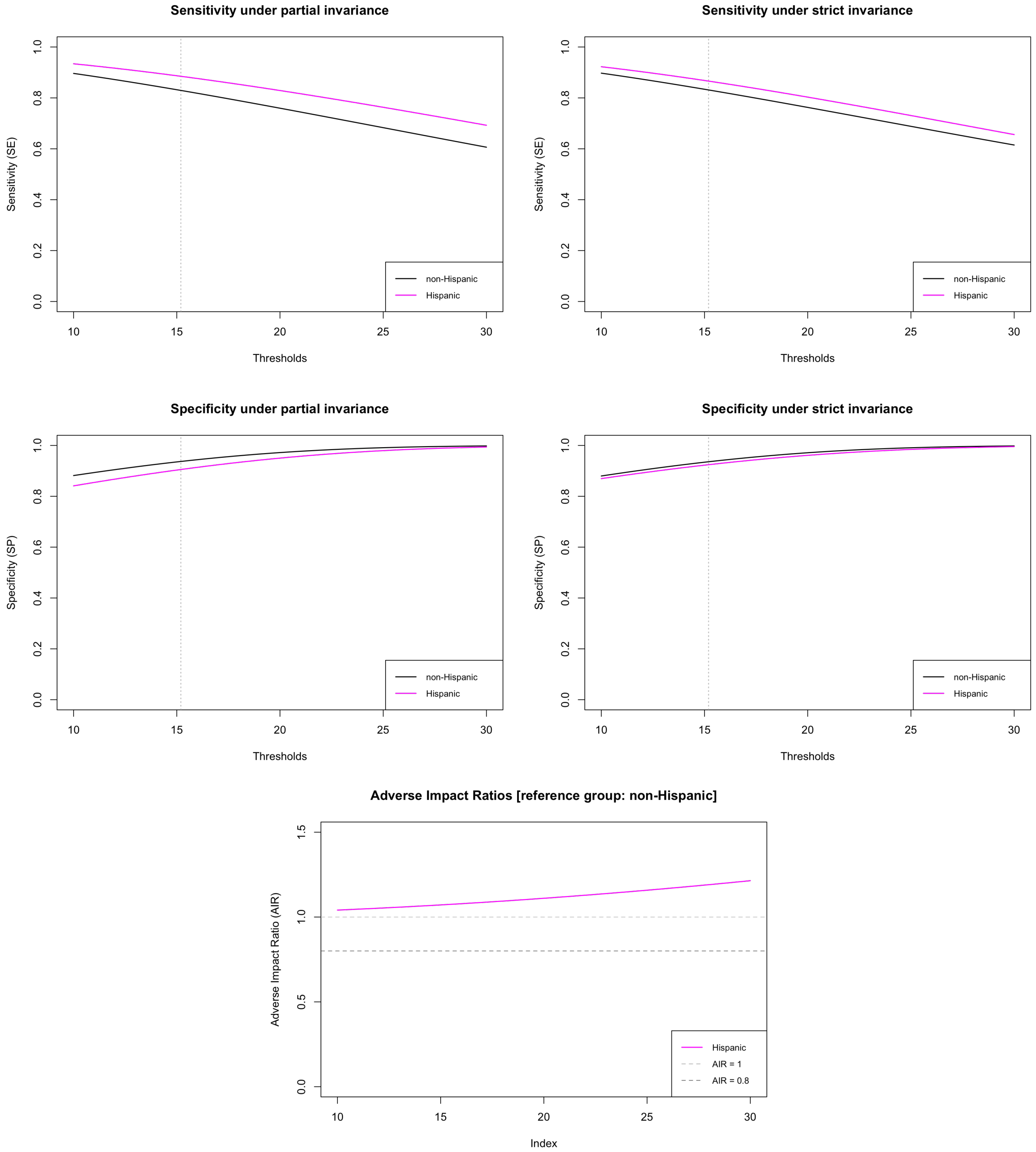
In the first two rows, figures on the left reflect partial MI and the figures on the right indicate strict MI. The vertical dotted line at 15.2 is equivalent to Radloff's (1977) suggested cutoff of 16 on 20 items. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Figure 5: Proportions selected (PS) and success ratios (SR) at different cutoffs (Z_c) by group (non-Hispanic, Hispanic).



Figures on the left reflect partial MI and the figures on the right indicate strict MI. The vertical dotted line at 15.2 is equivalent to Radloff's (1977) suggested cutoff of 16 on 20 items. Groups were collapsed to achieve a binary grouping for illustrative purposes.

Figure 6: Sensitivity (SE), Specificity (SP), and Adverse Impact ratios (AIR) at different cutoffs (Z_c) by group (non-Hispanic, Hispanic).



In the first two rows, figures on the left reflect partial MI and the figures on the right indicate strict MI. The vertical dotted line at 15.2 is equivalent to Radloff's (1977) suggested cutoff of 16 on 20 items. Groups were collapsed to achieve a binary grouping for illustrative purposes.

4 Differential item weights

Here, we demonstrate how item weighting with factor scores can be implemented. In practice, sum scores are commonly used for instruments such as the CES-D, and this section is provided for illustrative purposes only. In the following, we used `lavPredict()` with the fitted `cfa` object to obtain Bartlett score coefficient matrices for each group, averaged these to derive a single set of coefficients, mapped items to their respective factors using the model syntax, and then extracted the corresponding item–factor coefficients to serve as item weights. The new cutoff adjusting for the revised weighting scheme was $z_c = 7.16$. *pmix* was specified as before, and `PartInv()` was called with these configurations and the partial MI `cfa()` output from Step 1 to apply the multi-group MCAA framework.

```r
# Note: Before running the code below, run the relevant code chunks from the
# illustrative example to prepare the data, fit the configural and partial
# models, define the mixing weights (mixp) etc.

# Latent factor weights: place weights proportional to item counts
lw <- c(6, 7, 4, 2) / sum(c(6, 7, 4, 2))

# Item weights: use Bartlett factor score coefficients
fsm <- attr(lavPredict(partial_fit, method = "bartlett", fsm = TRUE), "fsm")
# Average matrices across groups
fsm_avg <- Reduce("+", fsm) / length(fsm)
# Map items to factors using the model syntax string
map_i_F <- with(subset(lavaanify(m1), op == "=~"), setNames(lhs, rhs))
# Extract the coefficient for each item for the appropriate factor
iw <- fsm_avg[cbind(map_i_F, names(map_i_F))]
# adj_w: scaling constant = average of item weights (used to adjust cutoff values)
adj_w <- sum(iw) / 19

# Call PartInv() using the cfa() output object
out_4g_w <- PartInv(
  cfa_fit = partial_fit, pmix = mixp, weights_latent = lw, weights_item = iw,
  cut_z = 15.2 * adj_w, plot_contour = TRUE, show_mi_result = TRUE, custom_colors = cols)

# Plot CAI and Adverse Impact Ratio across a range of thresholds
plot_CAI_across_range(
  x = out_4g_w, custom_colors = cols, cutoffs_from = 10 * adj_w,
  cutoffs_to = 30 * adj_w, add_vertical_threshold_at = 15.2 * adj_w)
```