

**visualizemi: Visualization, Effect Size, and Replication of Measurement
Invariance for Registered Reports**

Erin M. Buchanan¹

¹ Harrisburg University of Science and Technology

Author Note

Thank you to K.D. Valentine and Chelsea Parlett-Pelleriti for feedback on some ugly graphs.

Correspondence concerning this article should be addressed to Erin M. Buchanan, 326 Market St., Harrisburg, PA, USA. E-mail: ebuchanan@harrisburgu.edu

Abstract

Latent variable modeling as a lens for psychometric theory is a popular tool for social scientists to examine measurement of constructs (Beaujean, 2014). Journals such as *Assessment* regularly publish articles supporting measures of latent constructs wherein a measurement model is established. Confirmatory factor analysis can be used to investigate the replicability and generalizability of the measurement model in new samples, while multigroup confirmatory factor analysis is used to examine the measurement model across groups within samples (Brown, 2015). With the rise of the replication crisis and “psychology’s renaissance” (Nelson et al., 2018), interest in divergence in measurement has increased, often focused on small parameter differences within the latent model. This manuscript presents `visualizemi`, an *R* package that provides functionality to calculate multigroup models, partial invariance, visualizations for (non)-invariance, effect sizes for models and parameters, and potential replication rates compared to random models. Readers will learn how to interpret the impact and size of the proposed non-invariance in models with a focus on potential replication and how to plan for registered reports.

Keywords: multigroup confirmatory factor analysis, measurement invariance, visualization, effect size

visualizemi: Visualization, Effect Size, and Replication of Measurement Invariance for Registered Reports

Psychological assessments play a critical role in our ability to measure and analyze constructs to support theories and experimental hypotheses. Defining and creating assessments to validly and reliability measure constructs is often difficult because phenomenon, such as anxiety, are often not directly observable. Instead, we use surveys and questionnaires to indirectly assess the underlying construct (DeVellis & Thorpe, 2022). Latent variable modeling (i.e., structural equation modeling) is a popular tool for the validation of developed survey instruments to verify scale dimensionality, structure, and model fit. A simple search for scale development reveals thousands of articles in psychology that examine new and previously published work, thus, illustrating the interest in both measurement and the use of validation techniques. Unfortunately, except in specialty journals, much of the validity evidence and/or development for measures used in empirical studies is not reported within the journal article (Barry et al., 2014; Weidman et al., 2017). Without this information, it is difficult to interpret individual study conclusions, as validity information allows for judgment of usefulness of the measured values (Flake & Fried, 2020). Further, the current focus on replication (Makel et al., 2012; Makel & Plucker, 2014; Zwaan et al., 2018), reproducibility (Nelson et al., 2018), and the credibility of our results (Vazire et al., 2022) has demonstrated questionable measurement practices - decisions that researchers make like survey selection and scoring that impact the results of the study (Flake & Fried, 2020). Transparent reporting of the use and creation of scales can improve both interpretation and reproducibility when using surveys developed to measure latent constructs (Shadish et al., 2001).

A secondary concern for developed measures is the potential for differential responding and assessment within target populations. For example, Trent et al. (2013) examined for potential variability in the Revised Child Anxiety and Depression Scale in

White and Black youths (Chorpita et al., 2000). They found that the scale mostly functioned the same for both White and Black individuals but differences in averages on individual items could potentially affect the scoring and interpretation of the scale results. This comparison of sub-populations is the test of measurement invariance (Meredith, 1993). Invariance or equivalence implies that the scale operates in the same fashion for each sub-group, and thus, differences in the final latent variable scores can be interpreted as differences in populations. Non-invariance suggests that individuals respond or interpret items differently, and thus, differences in scores may represent different scores on the latent variable in the population or differences in measurement. Non-invariant measurement may lead to misleading results when making group comparisons, and assessing invariance has become a popular technique in scale development (Van De Schoot et al., 2015).

Measurement invariance has been explored and implemented for the last fifty years (Jöreskog, 1971; Sörbom, 1978) and implemented in the most popular structural equation modeling programs (Boker et al., 2011; Jöreskog & Sörbom, 2001; Rosseel, 2012). Byrne et al. (1989) extended the ideas of multigroup testing by suggesting partial invariance (followed by Meredith, 1993). Partial invariance occurs when non-invariance is found but can be attributed to only a few parameter estimate differences between groups (i.e., items 1 and 2 have different factor loadings but all others are the same). This testing provided an advantage to understand where the potential non-invariance may occur for further study and interpretation guidelines. As the field pushes back against favoring cutoff criteria and rules of thumb (Marsh et al., 2004; Putnick & Bornstein, 2016), an effect size measure for translating “how much” non-invariance was developed d_{MACS} (Nye & Drasgow, 2011). This effect size examines the differences in observed variables between the two groups for both the factor loading and the item intercept; thus, any differences in either or both will increase the effect size for non-invariance (Stark et al., 2006).

With d_{MACS} and measurement invariance testing, researchers can begin to quantify

how and where their construct measurement may vary between groups. Yet, given the large number of studies that show non-invariance, it is clear that equivalence can be hard to meet. It is difficult to know if non-invariance occurs because of random sampling error, true population differences, or differences in replication and reproducibility of the construct in a new sample. The field of psychology is increasingly interested in pre-registration (i.e., registering plans for a study before data collection, Nosek et al., 2018) and the promotion of transparency in study design, implementation, and analysis (Mayo-Wilson et al., 2021), in addition to supporting replication studies (Zwaan et al., 2018). Registered (replication) reports provide an advantageous avenue for the pre-registration of measurement tests, as they allow a researcher the ability to have their study accepted in principle, regardless of the results of a test of construct validity, reliability, or measurement invariance (Hobson, 2019; Nosek & Lakens, 2014). However, there are few tools that can provide effect size measures for models, individual parameters, or visualization for researchers to plan for future studies. d_{MACS} provides the opportunity to begin to think about the smallest effect size of interest or the smallest meaningful effect size for measurement invariance and replication (Anvari & Lakens, 2021; i.e., two studies with overlapping confidence intervals “replicate,” even if the test of measurement invariance does not, Lakens, 2017). As mentioned, d_{MACS} has only really been explored for a combined intercept and loadings, and while useful, does not necessarily allow a researcher to pinpoint specific issues within an observed variable.

Therefore, purpose of this manuscript is to describe an *R* package, **visualizemi**, that provides functionality to calculate multigroup confirmatory factor analysis, partial invariance tests, visualizations of the size of non-invariance, and potential effect sizes for overall models and individual parameters. No known visualization techniques have been proposed for measurement invariance. By creating panel visualizations, we can supplement a researcher’s ability to judge the strength of the non-invariance differences and effect size for each item. The proposed effect sizes demonstrate the likelihood of replication with a similar sample as compared to a randomly assigned group model, thus, illustrating what type of measurement

one might expect to find, and how different that is from random chance. Within this technique, the individual parameter effect sizes can be calculated: both the group differences within a model as compared to random and the likelihood of a parameter replication compared to random groups. Coupled with other indicators (i.e., fit indices differences, d_{MACS}), we can move toward a better understanding of how much measurement non-invariance is meaningful. This tutorial and package will help researchers plan future studies and aid in the ability to estimate a smallest effect of interest for measurement invariance studies, rather than relying on fit indices and rules of thumb alone.

By the end of this tutorial manuscript, readers will:

1. Learn how to use *visualizemi* to analyze multigroup confirmatory factor analysis, examine partial for invariance, and create visualizations of parameters.
2. Learn how to estimate the potential replication of multigroup models and their parameters using bootstrapping compared to a random group model.
3. Be able to calculate and interpret effect sizes for model and parameter replication, as well as parameter group differences.
4. Understand the impact of measurement variability on replication and generalizability.

The tutorial will start with a general overview of relevant topics to orient readers to invariance testing and MGCFA effect sizes. Next, the reader will learn about the **visualizemi** package functions for 1) running the multigroup analysis, 2) running a partial invariance analysis, 3) plotting the partial invariance, 4) estimating replication and effect sizes at the model level, and 5) estimating replication and effect sizes at the parameter level. Last, data from Aiena et al. (2014) and Chen et al. (2020) examining the measurement invariance of the RS-14 (Wagnild, 2009) will be used to demonstrate the application of the package on real data. The *visualizemi* package vignette includes an additional tutorial walk through.

Terminology

MGCFA

Multigroup confirmatory factor analysis (MGCFA) was proposed as a method to examine differences in scale functioning across groups (Brown, 2015) using structural equation modeling and confirmatory factor analysis (CFA). The goal of MGCFA is to determine if groups are invariant or show the same response patterns on the scale. MGCFA is often performed in sequential steps to determine the location and impact of potential non-invariance. The most common procedure starts by examining overall scale structure for each group individually to show that the groups can be combined into one nested model (i.e., both models at least converge). The combined model, often described as the *configural* model, creates one CFA for both groups that allows each group's estimated parameters to vary. Equality constraints between group parameters are then added sequentially to the model (Brown, 2015). If parameters are found to be equivalent between groups, these models are considered "invariant", suggesting that any subtle differences in the parameter estimates should not effect overall scoring and assessment.

The first equality constraint added is usually the item factor loadings (*metric* model) which requires items to be related to the latent variable at the same strength across groups. The intercepts are then constrained to be equal across groups (*scalar* model) to determine the invariance of item averages. The item variances (*strict* model) can then be examined to determine if the general variation in item answers are equal across groups. Other parameter equality constraints can be set at the latent variable level (latent means, variance, covariances), but the focus is generally on the observable variables and their parameter estimations. Non-invariance is determined by examining differences in fit indices between models with constraints versus without in sequential order (Cheung & Rensvold, 2002).

Partial Invariance

If a specific step within the MGCFA testing framework shows non-invariance, partial invariance is often used to investigate where and “how much” invariance occurs (i.e., how many items, rather than how big the differences are Byrne et al., 1989; Meredith, 1993). Similar to *post hoc* follow up tests in ANOVA, each equality constraint for that model is examined one at a time by allowing groups to vary. If the model fit improves when groups are allowed to have separate parameter estimates, the item for these groups is considered non-invariant. When only a few parameters are found that impact invariance, models are considered partially invariant, implying that most, but not all parameters are equal across groups. The researcher then interprets the impact of those items and parameters on overall scores and assessment usefulness. Partial invariance investigation is a useful tool for finding specific items that vary between groups, but does not fully explain the effect size of the difference between groups on those items and the impact on the overall model.

Effect Size: d_{MACS}

d_{MACS} was developed from Differential Item Functioning (DIF) measurement in Item Response Theory (IRT, Stark et al., 2004) wherein the effect size (DIF) portrays the group differences that lead to overall item score differences. Item scores can be mathematically defined as:

$$\hat{X}_{iR} = \tau_{iR} + \lambda_{iR}\xi$$

i indicates an individual item, and R indicates the reference group (Nye & Drasgow, 2011). Therefore, an individual item is predicted by the intercept of the item (τ) added to the item loading (λ) times the latent variable score (ξ). Nye and Drasgow (2011) demonstrated that the terminology and equations from IRT and DIF can be used to create a measure of item functioning differences within CFA, d_{MACS} . d_{MACS} is then the difference of reference group versus focal group (e.g., group 1 versus group 2) divided by a pooled

standard deviation similar to formulas for d proposed by Cohen (2013). Therefore, if τ or λ are individually (or both) different across groups, it will impact all the predicted X_{iR} scores, and thus, impact d_{MACS} values. While d_{MACS} represents a necessary step for development of effect sizes within CFA, it does not separate the differences in parameters between groups in a way that can be paired with traditional MGCFA testing steps and partial invariance. In addition to the missing effect sizes at model and parameter levels, no effect size to date gives the researcher a feeling for potential replication of the invariant or non-invariant items. Last, in line with general suggestions by Cumming (2012) and Cumming and Calin-Jageman (2016), the visualization of effect sizes in MGCFA would be an added tool for researchers to gauge the size of group differences.

Package Functions

The code for this manuscript can be found at <https://osf.io/wev5f/>. This tutorial was registered at <https://osf.io/vwf4d>, and the example provided at the end of the manuscript was added after that registration. The *R* package and replication/effect sizes was added after the original manuscript submission. The simulation study used to design plotting functions and test effect sizes can be found in the supplemental materials, along with worked code examples.

MGCFA: `mgcfa()` Function

First, we would create our model code in *lavaan* syntax (Rosseel, 2012). The *visualizemi* package does generally require raw data for bootstrapping purposes, and an example of how to simulate data from models and covariance/correlations tables that sometimes are provided in manuscripts (rather than the raw data) is provided in the supplemental documentation. The `mgcfa()` function is designed to flexibly allow you to leverage *lavaan*'s package functions to calculate multiple measurement steps at once. You would include:

- 1) the model syntax in the `model` argument.

- 2) the dataframe in the `data` argument of our function
- 3) the name of the grouping variable in quotes for `group`.
- 4) and the equality constraints you would like to impose in order in `group.equal`.
- 5) ... any other *lavaan* arguments you would like to use such as `meanstructure` or `estimator`.

The following output is saved:

- 1) `model_coef`: The parameter estimates for each model with the model step included in a *model* column. This set of coefficients can be used for other functions. This dataframe is created with *broom*'s `tidy()` function if you wish to recreate this table without running the `mgcfa()` function (Robinson et al., 2023).
- 2) `model_fit`: The model fit indices from `fitmeasures()` to review for overall model fit and invariance judgments. The name of the model is included in a *model* column.
- 3) `model_overall`: A saved *lavaan* fitted model of all groups together without any equality constraints or grouping variables. These objects can be used with any function that normally takes a saved model: `parameterEstimates()`, `modificationIndices()`, `semPlot::semPaths()`, and so on (Epskamp, 2022).
- 4) `group_models`: A list of saved fitted models for each group separately.
- 5) `model_configural`: A saved fitted model for the configural model that nests together each group into one model with no other constraints.
- 6) `invariance_models`: A list of saved fitted models that consecutively adds `group.equal` constraints.

Partial Invariance: `partial_mi()` Function

The `partial_mi()` function aids in the calculation of partial invariance for a specific step of the MGCFA process. The function includes the following arguments:

- 1) `saved_model`: The saved *lavaan* model with the equality constraints at the level of

measurement invariance you would like to examine for partial invariance.

2) **data**: The dataframe where the model was estimated.

3) **model**: The model syntax for the overall model.

4) **group**: The grouping variable column in the dataframe.

5) **group.equal**: The equality constraints including in your original multigroup tests.

6) **partial_step**: The level of partial invariance you wish to test.

In this function, each parameter with the appropriate *lavaan* syntax is relaxed individually (i.e., ~1 for intercepts, ~~ for residuals, etc.). The fitted models are saved in the **models** output, and the **fit_table** output includes all fit indices for each model to investigate potential areas of partial invariance based on the researcher's desired criterion.

Visualization of Invariance: **plot_mi()** Function

Once we know which items are non-invariant, the **model_coef** output from the **mgcfa()** can be used directly in **plot_mi()**. The plot outputs will be described below. First, here are the arguments for the function:

1) **data_coef**: A tidy dataframe of the parameter estimates from the models. This function assumes you have used **broom::tidy()** on the saved model from *lavaan* and added a column called "model" with the name of the model step (Robinson et al., 2023). This function will only run for models that have used the grouping function (i.e., configural, metric, scalar, and strict or other combinations/steps you wish to examine).

2) **model_step**: Which model do you want to plot? You should match this name to the one you want to extract from your model column in the **data_coef**.

3) **item_name**: Which observed variable from your model syntax do you want to plot? Please list this variable name exactly how it appears in the model.

4) **x_limits**: What do you want the x-axis limits to be for your invariance plot? The default option is to assume the latent variable is standardized, and therefore, -1 to 1 is recommended. Use only two numbers, a lower and upper limit. This value also

constrains the latent mean diagram to help zoom in on group differences because the scale of latent means is usually centered over zero. You can use this parameter to zoom out to a more traditional histogram using `c(-2, 2)`.

5) **y_limits**: What do you want the y-axis limits to be for your invariance plot? Given that the latent variable is used to predict the observed values in the data, you could use the minimum and maximum values found in the data. If that range is large, consider reducing this value to be able to visualize the results (i.e., otherwise it may be too zoomed out to judge group differences). Use only two numbers, a lower and upper limit.

6) **conf.level**: What confidence limit do you want to plot? Use $1 - \alpha$.

7) **model_results**: In this argument, include the saved *lavaan* output for the model listed in the **model_step** argument.

8) **lv_name**: Include the name of the latent variable, exactly how it is listed in your *lavaan* syntax. You should plot the latent variable that the **item_name** is linked to. If you have items that load onto multiple latent variables, you will need to make multiple plots.

9) **plot_groups**: If you include more than two groups in a multigroup model, the automatic assumption is that you want the first two groups for this visualization. If not, include the names of the groups here to plot.

The outputs from this function are several *ggplot2* objects that can be edited or saved directly using *ggplot2* functionality (Wickham, 2016).

1) **complete**: The output from this model can be found in Figure 1. On the left-hand side, the item invariance is plotted, and on the right-hand side, the latent mean distributions for the two groups are plotted. In the item invariance sub-plot, the visualization includes all three components traditionally seen in MGCFA testing steps: loadings, intercepts, and residuals. Each visualization element was designed to match the

traditional visualization for that type of output. All parameter estimates are plotted on the unstandardized estimates and their confidence interval based on the standard error of the estimate. All plots are made with *ggplot2* and *cowplot* (Wilke, 2020).

- 2) **intercept:** Only the left-hand side of the complete plot designed to represent intercepts and factor loadings. Factor loadings represent the slope of the regression equation for the latent variable predicting the scores on the observed variable ($\hat{Y} \sim b_0 + b_1X + \epsilon$). The y-axis indicates the observed variable scores, and here, the plot includes the entire range of the scale of the data this simulated item. The ribbon bands around the plotted slopes indicate the confidence interval for that estimate. In this plot, while the coefficients for each group are not literally equal, the overlapping and parallel slope bands indicate they are not different practically.

The item intercepts (b_0) are plotted on the middle line where they would cross the y-axis at a latent variable score of zero. These are represented by a dot with a set of confidence error bars around the point. In this invariant depiction, the overlap in the intercepts is clear, indicating they are not different. You can use `y_limits` to zoom in on the graph if these are too small to be distinguishable.

- 3) **mean:** The right-hand side of the complete plot graphing the latent variable means and density from the data. The latent variable is shown on the x-axis using standardized values (i.e., z-scores) where -1 indicates one standard deviation below the mean for the latent variable, 0 indicates the mean for the latent variable and so on. The lines indicate the means of the latent variables from the simulated dataset. Group labels are represented in the figure caption on the bottom. Group 1 is usually the group that is alphabetically first in the data set or whichever group is the first that appears when using the `levels()` command.

- 4) **variance:** A split geom violin plot indicating the variance distribution of the plotted

item. Residuals are trickier to plot, as they are the left over error when predicting the observed variables ϵ . It is tempting to plot this value as the confidence band around the slope, however, that defeats the purpose of understanding that the slopes are estimated separately from the residuals, and both have an associated variability around their parameter estimate. Therefore, residuals are represented in the inset picture at the bottom right of the item invariance plot. The black bars represent the estimated residual for each group. The distributions are plotted to represent the normal spread of values using the standard error of the residuals. The violin plot allows for direct comparison of those residuals and their potential distributions. Note that the placement has nothing to do with the x or y-axis and is designed to always show in the same location, regardless of size/value. The plots are included separately so they can be arranged in a different fashion if desired.

Model Replication and Effect Sizes: `bootstrap_model()` Function

The `bootstrap_model` function in *visualizemi* was designed to estimate the likely replication of overall model invariance with the assumption that the data used for the estimation represents the larger population. The following arguments are used:

- 1) **saved_configural**: a saved fitted model at the configural level with no equality constraints. This model should include all other lavaan settings you would like to use, such as estimator or ordered.
- 2) **data**: The dataframe where the model was estimated.
- 3) **model**: The model syntax for the overall model.
- 4) **group**: The grouping variable column in the dataframe.
- 5) **nboot**: The number of bootstraps to run.
- 6) **invariance_index**: The fit index you would like to use to determine invariance. Please use options and labeling from *lavaan* - see `fitmeasures()` for options.
- 7) **invariance_rule**: The invariance difference score you would like to use as your rule.

8) `group.equal`: The equality constraints including in your original multigroup tests.

The data included in this function will be sampled, with replacement, at the same size as the current dataset, and the included invariance equality constraints are estimated. Each step will be compared to the previous step using the invariance index and comparison rule entered. The output is a dataframe of the proportion of non-invariant bootstraps from the real data and the same bootstrapped dataset with the group labels randomly assigned. The effect size comparison of proportions, h , for non-invariant comparisons:

$$h_{nmi} = 2 \times (\text{asin}\sqrt{p_{data}} - \text{asin}\sqrt{p_{random}})$$

The alternative, h_{mi} , for effect size of measurement invariance replication would simply be the inverse sign of h_{nmi} and is also included in the table. Two additional columns h_{nmi_p} and h_{mi_p} represent the h values divided by the upper bound of h (i.e., π), to help with interpretation of the effect size (thus, bounding h to -1 to 1).

Parameter Replication and Effect Sizes: `bootstrap_partial()` Function

After examining the overall model potential replication effect size, the individual parameters within a model can be bootstrapped for partial invariance to with that parameter relaxed (overall partial model statistics) and the difference in group parameter estimates (parameter effect size). This function uses arguments seen in other functions, so they will not be repeated here. The general setup consists of using the model you think could be partially invariant in the `saved_model` argument and the fit index for comparison for the model with less constraints in `invariance_compare`. The `partial_step` argument will be used to determine which operation syntax (i.e. `=~` for loadings) to relax for modeling.

The saved output includes several dataframes and plots. The first is the `boot_DF` which is the summary of each bootstrapped run in a dataframe for plotting or summarization. This dataframe includes the estimate for each parameter (`term`) separated by group and type

(`boot_1`, `boot_2` are the bootstrapped estimates for group 1 and group 2, while the same `random` columns indicate the randomly assigned groups). The fit index used to determine invariance is included for bootstrapped and random estimates, and then the differences between groups and if they were “invariant” or not given the researcher supplied rule.

Next, the `boot_summary` includes a summarized form of the bootstrapped results separated by bootstrapped data versus randomized data and then invariant/non-invariant outcomes. The d_s for between groups Cohen’s d is included (Lakens, 2013). Effect sizes are only calculated when the number of bootstrapped estimates is at least 10% of the data - therefore, you would not receive effect sizes with almost no bootstrapped runs. This dataframe should be used to determine which parameter may be different and the effect size potential between groups in a replication of the study. The `boot_effects` table creates a summary similar to the overall model replication table based on the proportion of runs that were considered invariant versus not for each parameter.

Plots of the results from dataframes can be found within the `bootstrap_partial()` function. Figure 2 shows the difference between parameters for groups in the bootstrapped and randomly assigned group runs in simulated data. Figure 3 shows the density plot of the estimates for each group organized by bootstrapped and randomly assigned groups and the invariance decision for each bootstrapped run. Last, Figure 4 indicates the d_s value between groups with an indication of the number of data points in each estimate (i.e., dot size). These visualizations should allow a researcher to understand the likelihood of replication for each parameter, as well as the potential size of the differences. Therefore, one could indicate a specific smallest effect size of interest, rather than a invariance cut-off rule of thumb when planning a replication or registered report.

An Example Analysis

Aiena et al. (2014) examined the RS-14 (Wagnild, 2009) exploring the factor structure of the Resiliency Scale in a clinical sample receiving treatment services and a

college student sample. Measurement invariance was calculated for differences separately for these samples for gender and race finding a partially invariant models with a few item intercepts or residuals that differed between groups. Aiena et al. (2014) did not compare the clinical to the student sample for measurement invariance, and it is reasonable to expect potential differences in these two populations. This example will demonstrate the procedure for researchers who wish to use partial invariance steps and how to interpret real, messy data.

Table 1 indicates the results after running the one-factor model. There are several guidelines for assessing a degradation in model fit (Cao & Liang, 2022; Cheung & Rensvold, 2002; Counsell et al., 2020; Jin, 2020; Putnick & Bornstein, 2016) but for the purposes of this illustration $\Delta CFI > .01$ will be used. Table 1 indicates that fit was degraded when the constraint on equal item intercepts was added. The code online provides an example of testing each item individually by relaxing the constraints and recalculating the CFI. If these items bring the CFI value back up to $\Delta CFI \leq .01$ from the metric model, then the model would be considering partially invariant at the scalar level. It seems unlikely that the residuals will show invariance, even if partial scalar invariance can be found, as the drop in fit on the residual model is quite large.

The partial invariance results indicated that RS6 and RS7 are potential items that could be relaxed to improve model fit and create a partial scalar invariant model (i.e., by picking the largest CFI values). By examining our estimates, we can see that item seven on the RS-14 is estimated at nearly 5 points for the clinical sample, while the student sample has a lower mean around 4.5 points. Generally, students show higher means on the items of the RS-14, but when all loadings and other intercepts are constrained to be equal, and this one item is relaxed, this pattern flips so that clinical groups show higher item intercepts. Given the scale is a 1-7 Likert type scale, .5 a point represents a potentially sizable change on the scale. Item seven covers perseverance after hardship, and all items can be found in the user manual for the scale at www.resiliencecenter.com. The effect size from d_{MACS}

suggests a small to medium effect, 0.28. See Figure 5 for the difference between item intercepts and latent means. We repeat this process for the RS6, as the CFI for our model with only RS7 does not achieve the levels of partial invariance for our ΔCFI criterion (i.e., $\leq .01$ downward change in fit: metric CFI = .925, partial scalar CFI = .914). Again, we see about a half-point difference between our clinical and student samples for item 6, which is about drive to achieve. The CFI for this model does meet the requirements for partial invariance, .917. The effect size is approximately the same at $d_{MACS} = 0.28$. See Figure 6 shows the difference between item intercepts and latent means.

Next, we would examine our replication potential for this model. Given our current results, we may not expect our intercepts to replicate. Given the order of desired steps in the `group.equal` argument, the boot function will select the first non-invariant step (as defined by the user) in the calculation of the effect size for potential replication. In our output, we do not see a loadings effect size, and this result occurs when *none* of the bootstrapped or random results are non-invariant. Therefore, we would expect the loadings to replicate (and the effect size would be 0 difference between bootstrapped and random, both showing invariance). The intercepts show a large (i.e., close to the max possible value) non-invariant effect, $h_{nmi} = 3.05$ ($h_{mni_p} = 0.97$), and therefore, we should not expect this model to show invariance in a replication.

Next, we would examine the strength of the effects of replication on each parameter at the intercept level. By examining Table 2, it is clear that most of the item means are unlikely to replicate, even though two particular items can be used to create partial invariance. Figures 7 and 8 display the three plots provided in the `bootstrap_partial()` function. In general, we should expect $M_D = 0.11$ when items are invariant and $M_D = 0.25$ when items are not invariant. The effect size of non-invariant items ranges from $d_s \sim 0.43$ to 0.59.

The density plot shown at the bottom of Figure 7 illustrates the likely reasons for the differences found in the top plots. It appears that many items show a bimodal distribution

within group 1 (Clinical Sample) and when items are invariant, the intercept averages to the same intercept as group 2 (Student Sample). In non-invariant estimates, the same bimodal distributions are found, but they are more extreme than the student samples, and therefore, item show different averages due to the presence of two separate means of data. Further, some items also appear to show two separate student item averages within the data. This result suggests that it would be fruitful to understand a potential predictor of these differences or other confounding variable that separates these samples, creating differences in item averages.

In summary, if one were planning a replication, the prediction would be that item intercepts would likely not replicate, with a large effect size (i.e., it is easy to judge h_{nmi_p} close to the max of one as large). While this study found partial invariance by relaxing constraints on two individual items, bootstrapped partial invariance indicates that any item could potentially be problematic with an effect size averaging $d_s \sim 0.50$ difference in means. While d_{MACS} values represented a “small” effect based on previous publications, this effect may be muted by examining both loadings and intercepts. The results here suggest that the effect is driven by intercepts. The overall average score on items is high: $M_M = 5.04$ ($M_{SD} = 1.72$). Given the mean and standard deviation, a $d_s \sim 0.50$ represents 0.86 or nearly one whole point on the scale. A researcher could decide that at least $d_s = 0.33$ or at least a third of a standard deviation would be an important change and set that as their smallest effect size of interest for invariance replication and extension. Further, a newly planned study should investigate what variables may predict when and why samples separate into bimodal representations for item means.

An Example Extension

One benefit of the open science movement on scale development is the publication of datasets or covariance tables with published articles. We can extend our examination of potential replication on other variables that may effect assessment of underlying phenomena.

For example, scale translation across languages is not only impacted by the literal conversion of concepts from one language to another, but also the cultural and societal norms of the target population (Cha et al., 2007; Chang et al., 1999; Yu et al., 2004). The RS14 was tested in Chinese speaking samples in Chen et al. (2020) across five different large samples and determined that the scale showed good psychometric properties for use within Chinese speaking samples as a one-factor model of resiliency. Given these results, another researcher may assume that the models would be comparable between English speaking (i.e., likely United States) and Chinese speaking samples. With the published data, we can use *visualizemi* to determine if the RS14 is invariant across language/culture, and if the results would likely replicate if tested on a new set of samples, and what, if any, effect size differences are found in parameters. The code used for this analysis is presented in the supplemental materials.

The Aiena et al. (2014) data used in our previous example was first filtered to only college students, as we have already noted that clinical samples show slightly different intercepts for at least two of the items from student samples. The Chen et al. (2020) data also included college students, which allows us to test a comparable sample that varies on translation and culture. The test of measurement invariance indicated that the factor structure and loadings were invariant across groups. Much like our test of clinical versus student samples, the results indicated that the intercepts were not consistent across groups. Within the English clinical/student sample, partial invariance could be achieved by relaxing two item intercept constraints (item 6 and 7). To achieve “partial” invariance between the Chinese and English speaking samples, we would need to relax more than half of the items (specifically, eight items: 1, 2, 3, 4, 10, 11, 12, 14), and it would be difficult to suggest partial invariance given this finding. d_{MACS} values range from 0.26 to 0.32 for the eight items which could be interpreted a small to medium given Nye et al. (2019)’s simulation study. Figure 9 portrays the results from the second item on the RS14 (*life accomplishments*).

The results of model bootstrapping indicated that the effect of the loadings was likely to replicate (only invariant results were found), but the intercepts were never found to be invariant compared to a randomized sample ($h_{nmi_p} = 1$). Therefore, we would expect that these differences are persistent, either due to the adaptation or cultural differences in resiliency across samples. The bootstrapped partial invariance demonstrates that each item intercept has a medium to large difference between the two samples as shown in Figure 10, which may explain why full or partial invariance cannot be achieved. This result does not invalidate either version of the scale, but informs researchers of potential baseline differences in item responding for the two samples. Therefore, careful interpretation should be made when comparing these two samples in other instances, as differences in latent means may be the default finding, but with these results one may determine if their results are different from expectations of general scale responding.

Discussion

In this tutorial, we examined how to use multiple tools to examine measurement invariance and its potential replication. Model fit comparisons and statistics can be paired with the proposed effect size measures, and visualizations to examine individual items and the overall latent mean scores. The impact of potential replication was estimated on the overall model and the individual parameters. Using real data, the effect of two non-invariant item intercepts was examined and visualized. This tutorial manuscript has provided a concrete way to plan for pre-registration and/or registered reports. Researchers could simulate results based on published or previously collected data to determine the likelihood and size of potential replication. They could plan and pre-register a smallest effect of interest. For example, we may determine that an h_{nmi_p} value above .20 represents an important level of non-invariance for our model overall, while $h_{nmi_p} > .30$ for any individual parameter warrant caution against invariance for groups. Others have begun to discuss the importance of focusing on effects in the scale of the data and their practical importance (Anvari & Lakens, 2021; Cumming, 2012).

From the example, our interpretation may be that the difference between group's latent means is large, as a 0.72 change on a 7 point scale is approximately 10% more resiliency for students when compared to the clinical sample. Practically, 10% in resiliency for an area of the United States (Mississippi) often hit with natural disasters (hurricanes, tornadoes, floods) and high levels of poverty would be very important. Even the smaller difference of .5 point on each individual item could translate into increases in resiliency, and these results may elucidate avenues for further exploration into areas of focus within resiliency, given the items. The secondary example showed that we can extend these results to other samples to examine other potentially impactful variables on assessment. The findings replicate in the sense that the scale shows the same invariant issues with intercepts on a Chinese versus English sample comparison. However, in this analysis, it is clear that the differences in items averages are much larger and across all items, rather than a few.

What do the results of a study on measurement invariance with these results tell us about replication, generalizability, and validity overall? If a researcher decides their effects are large, they should likely caution against suggesting that these scores are directly comparable without weighting or other adjustment. Let's consider a scenario wherein the change metric between models picked (i.e., ΔCFI , $\Delta RMSEA$) indicates a "significant" change in model fit. However, if both the effect size and a visual inspection of the invariance indicates a small difference, we may decide to lessen the practical importance of those results, much like "just significant" p -values with small effect sizes are treated now. The results from our Chinese versus English comparison show us the other scenario: non-invariant results that clearly indicate differences with a large effect size on both replication and item average differences. Overall, given that the goal of measurement invariance is to compare parameter *estimates*, we should expect some differences across samples due to the nature of sampling and estimation. It may be that many of the published models presented represent these effects - small variations between groups due to sampling error or other small crud - but do not represent a fundamental problem with the measurement or generalizability of the results.

541 The `visualizemi` package is one useful tool to help sort out if findings are this small
542 sampling error or differences in samples due to other variables.

References

- Aiena, B. J., Baczwaski, B. J., Schulenberg, S. E., & Buchanan, E. M. (2014). Measuring Resilience With the RS-14: A Tale of Two Samples. *Journal of Personality Assessment*, 97(3), 291–300. <https://doi.org/10.1080/00223891.2014.951445>
- Akaike, H. (1998). *Information theory and an extension of the maximum likelihood principle* (E. Parzen, K. Tanabe, & G. Kitagawa, Eds.; pp. 199–213). Springer New York. http://link.springer.com/10.1007/978-1-4612-1694-0_15
- Anvari, F., & Lakens, D. (2021). Using anchor-based methods to determine the smallest effect size of interest. *Journal of Experimental Social Psychology*, 96, 104159. <https://doi.org/10.1016/j.jesp.2021.104159>
- Barry, A. E., Chaney, B., Piazza-Gardner, A. K., & Chavarria, E. A. (2014). Validity and Reliability Reporting Practices in the Field of Health Education and Behavior: A Review of Seven Journals. *Health Education & Behavior*, 41(1), 12–18. <https://doi.org/10.1177/1090198113483139>
- Beaujean, A. A. (2014). *Latent variable modeling using r: A step by step guide*. Routledge/Taylor & Francis Group.
- Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, 107(2), 238–246. <https://doi.org/10.1037/0033-2909.107.2.238>
- Bentler, P. M. (1995). *EQS structural equations program manual*.
- Boker, S., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T., Spies, J., Estabrook, R., Kenny, S., Bates, T., Mehta, P., & Fox, J. (2011). OpenMx: An Open Source Extended Structural Equation Modeling Framework. *Psychometrika*, 76(2), 306–317. <https://doi.org/10.1007/s11336-010-9200-6>
- Brown, T. A. (2015). *Confirmatory factor analysis for applied research* (Second edition). The Guilford Press.
- Byrne, B. M., Shavelson, R. J., & Muthén, B. (1989). Testing for the equivalence of factor covariance and mean structures: The issue of partial measurement invariance.

- Psychological Bulletin*, 105(3), 456–466. <https://doi.org/10.1037/0033-2909.105.3.456>
- Cao, C., & Liang, X. (2022). The impact of model size on the sensitivity of fit measures in measurement invariance testing. *Structural Equation Modeling: A Multidisciplinary Journal*, 29(5), 744–754. <https://doi.org/10.1080/10705511.2022.2056893>
- Cha, E.-S., Kim, K. H., & Erlen, J. A. (2007). Translation of scales in cross-cultural research: issues and techniques. *Journal of Advanced Nursing*, 58(4), 386–395. <https://doi.org/10.1111/j.1365-2648.2007.04242.x>
- Chang, A. M., Chau, J. P. C., & Holroyd, E. (1999). Translation of questionnaires and issues of equivalence. *Journal of Advanced Nursing*, 29(2), 316–322. <https://doi.org/10.1046/j.1365-2648.1999.00891.x>
- Chen, W., Xie, E., Tian, X., & Zhang, G. (2020). Psychometric properties of the Chinese version of the Resilience Scale (RS-14): Preliminary results. *PLOS ONE*, 15(10), e0241606. <https://doi.org/10.1371/journal.pone.0241606>
- Cheung, G. W., & Rensvold, R. B. (2002). Evaluating Goodness-of-Fit Indexes for Testing Measurement Invariance. *Structural Equation Modeling: A Multidisciplinary Journal*, 9(2), 233–255. https://doi.org/10.1207/s15328007sem0902_5
- Chorpita, B. F., Yim, L., Moffitt, C., Umemoto, L. A., & Francis, S. E. (2000). Assessment of symptoms of DSM-IV anxiety and depression in children: a revised child anxiety and depression scale. *Behaviour Research and Therapy*, 38(8), 835–855. [https://doi.org/10.1016/S0005-7967\(99\)00130-8](https://doi.org/10.1016/S0005-7967(99)00130-8)
- Cohen, J. (2013). *Statistical Power Analysis for the Behavioral Sciences*. Routledge. <https://doi.org/10.4324/9780203771587>
- Cohen, J., Cohen, P., West, S. G., & Aiken, L. (2003). *Applied multiple regression / correlation analysis for the behavioral sciences* (3rd ed.). Lawrence Erlbaum Associates.
- Counsell, A., Cribbie, R. A., & Flora, D. B. (2020). Evaluating equivalence testing methods for measurement invariance. *Multivariate Behavioral Research*, 55(2), 312–328. <https://doi.org/10.1080/00273171.2019.1633617>

- Cumming, G. (2012). *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge.
- Cumming, G., & Calin-Jageman, R. (2016). *Introduction to the New Statistics* (0th ed.). Routledge. <https://doi.org/10.4324/9781315708607>
- DeVellis, R. F., & Thorpe, C. T. (2022). *Scale development: Theory and applications* (Fifth edition). SAGE Publications, Inc.
- Dueber, D. (2023). *Dmacs*. <https://github.com/ddueber/dmacs>
- Epskamp, S. (2022). *semPlot: Path diagrams and visual analysis of various SEM packages' output*. <https://CRAN.R-project.org/package=semPlot>
- Flake, J. K., & Fried, E. I. (2020). Measurement Schmeasurement: Questionable Measurement Practices and How to Avoid Them. *Advances in Methods and Practices in Psychological Science*, 3(4), 456–465. <https://doi.org/10.1177/2515245920952393>
- Hobson, H. (2019). Registered reports are an ally to early career researchers. *Nature Human Behaviour*, 3(10), 1010–1010. <https://doi.org/10.1038/s41562-019-0701-8>
- Jin, Y. (2020). A note on the cutoff values of alternative fit indices to evaluate measurement invariance for ESEM models. *International Journal of Behavioral Development*, 44(2), 166–174. <https://doi.org/10.1177/0165025419866911>
- Jöreskog, K. G. (1971). Simultaneous factor analysis in several populations. *Psychometrika*, 36(4), 409–426. <https://doi.org/10.1007/BF02291366>
- Jöreskog, K. G., & Sörbom, D. (2001). *LISREL 8: user's reference guide* (2. ed., updated to LISREL 8). SSI Scientific Software Internat.
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, 4. <https://doi.org/10.3389/fpsyg.2013.00863>
- Lakens, D. (2017). Equivalence Tests. *Social Psychological and Personality Science*, 8(4), 355–362. <https://doi.org/10.1177/1948550617697177>
- Makel, M. C., & Plucker, J. A. (2014). Facts Are More Important Than Novelty:

Replication in the Education Sciences. *Educational Researcher*, 43(6), 304–316.

<https://doi.org/10.3102/0013189X14545513>

Makel, M. C., Plucker, J. A., & Hegarty, B. (2012). Replications in Psychology Research:

How Often Do They Really Occur? *Perspectives on Psychological Science*, 7(6), 537–542.

<https://doi.org/10.1177/1745691612460688>

Marsh, H. W., Hau, K.-T., & Wen, Z. (2004). In search of golden rules: Comment on

hypothesis-testing approaches to setting cutoff values for fit indexes and dangers in

overgeneralizing hu and bentler's (1999) findings. *Structural Equation Modeling: A*

Multidisciplinary Journal, 11(3), 320–341. https://doi.org/10.1207/s15328007sem1103_2

Mayo-Wilson, E., Grant, S., Supplee, L., Kianersi, S., Amin, A., DeHaven, A., & Mellor, D.

(2021). Evaluating implementation of the transparency and openness promotion (TOP)

guidelines: The TRUST process for rating journal policies, procedures, and practices.

Research Integrity and Peer Review, 6(1), 9. <https://doi.org/10.1186/s41073-021-00112-8>

Meredith, W. (1993). Measurement invariance, factor analysis and factorial invariance.

Psychometrika, 58(4), 525–543. <https://doi.org/10.1007/BF02294825>

Nelson, L. D., Simmons, J., & Simonsohn, U. (2018). Psychology's renaissance. *Annual*

Review of Psychology, 69(1), 511–534.

<https://doi.org/10.1146/annurev-psych-122216-011836>

Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration

revolution. *Proceedings of the National Academy of Sciences of the United States of*

America, 115(11), 2600–2606. <https://doi.org/10.1073/pnas.1708274114>

Nosek, B. A., & Lakens, D. (2014). Registered Reports: A Method to Increase the

Credibility of Published Results. *Social Psychology*, 45(3), 137–141.

<https://doi.org/10.1027/1864-9335/a000192>

Nye, C. D., Bradburn, J., Olenick, J., Bialko, C., & Drasgow, F. (2019). How Big Are My

Effects? Examining the Magnitude of Effect Sizes in Studies of Measurement Equivalence.

Organizational Research Methods, 22(3), 678–709.

651 <https://doi.org/10.1177/1094428118761122>

652 Nye, C. D., & Drasgow, F. (2011). Effect size indices for analyses of measurement
653 equivalence: Understanding the practical importance of differences between groups.

654 *Journal of Applied Psychology*, 96(5), 966–980. <https://doi.org/10.1037/a0022955>

655 Putnick, D. L., & Bornstein, M. H. (2016). Measurement invariance conventions and
656 reporting: The state of the art and future directions for psychological research.

657 *Developmental Review*, 41, 71–90. <https://doi.org/10.1016/j.dr.2016.06.004>

658 Robinson, D., Hayes, A., Couch [aut, S., cre, Software, P., PBC, Patil, I., Chiu, D., Gomez,
659 M., Demeshev, B., Menne, D., Nutter, B., Johnston, L., Bolker, B., Briatte, F., Arnold,

660 J., Gabry, J., Selzer, L., Simpson, G., ... Reinhart, A. (2023). *Broom: Convert*
661 *statistical objects into tidy tibbles*. <https://CRAN.R-project.org/package=broom>

662 Rosseel, Y. (2012). Llavaan: An r package for structural equation modeling. *Journal of*
663 *Statistical Software*, 48(1), 1–36. <https://doi.org/10.18637/jss.v048.i02>

664 Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2),
665 461–464. <https://www.jstor.org/stable/2958889>

666 Shadish, W. R., Cook, T. D., & Campbell, D. T. (2001). *Experimental and*
667 *quasi-experimental designs for generalized causal inference*. Houghton Mifflin.

668 Sörbom, D. (1978). An alternative to the methodology for analysis of covariance.
669 *Psychometrika*, 43(3), 381–396. <https://doi.org/10.1007/BF02293647>

670 Stark, S., Chernyshenko, O. S., & Drasgow, F. (2004). Examining the Effects of Differential
671 Item (Functioning and Differential) Test Functioning on Selection Decisions: When Are
672 Statistically Significant Effects Practically Important? *Journal of Applied Psychology*,
673 89(3), 497–508. <https://doi.org/10.1037/0021-9010.89.3.497>

674 Stark, S., Chernyshenko, O. S., & Drasgow, F. (2006). Detecting differential item
675 functioning with confirmatory factor analysis and item response theory: Toward a unified
676 strategy. *Journal of Applied Psychology*, 91(6), 1292–1306.

677 <https://doi.org/10.1037/0021-9010.91.6.1292>

- Steiger, J. H. (1990). Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research*, 25(2), 173–180.
https://doi.org/10.1207/s15327906mbr2502_4
- Trent, L. R., Buchanan, E., Ebesutani, C., Ale, C. M., Heiden, L., Hight, T. L., Damon, J. D., & Young, J. (2013). A measurement invariance examination of the revised child anxiety and depression scale in a southern sample: Differential item functioning between african american and caucasian youth. *Assessment*, 20(2), 175–187.
<https://doi.org/10.1177/1073191112450907>
- Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika*, 38(1), 1–10. <https://doi.org/10.1007/BF02291170>
- Van De Schoot, R., Schmidt, P., De Beuckelaer, A., Lek, K., & Zondervan-Zwijnenburg, M. (2015). Editorial: Measurement invariance. *Frontiers in Psychology*, 6.
<https://www.frontiersin.org/articles/10.3389/fpsyg.2015.01064>
- Vazire, S., Schiavone, S. R., & Bottesini, J. G. (2022). Credibility Beyond Replicability: Improving the Four Validities in Psychological Science. *Current Directions in Psychological Science*, 31(2), 162–168. <https://doi.org/10.1177/09637214211067779>
- Wagnild, G. (2009). A review of the resilience scale. *Journal of Nursing Measurement*, 17(2), 105–113. <https://doi.org/10.1891/1061-3749.17.2.105>
- Weidman, A. C., Steckler, C. M., & Tracy, J. L. (2017). The jingle and jangle of emotion assessment: Imprecise measurement, casual scale usage, and conceptual fuzziness in emotion research. *Emotion*, 17(2), 267–295. <https://doi.org/10.1037/emo0000226>
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York.
<https://ggplot2.tidyverse.org>
- Wilke, C. O. (2020). *Cowplot: Streamlined plot theme and plot annotations for 'ggplot2'*.
<https://CRAN.R-project.org/package=cowplot>
- Yu, D. S. F., Lee, D. T. F., & Woo, J. (2004). Issues and Challenges of Instrument Translation. *Western Journal of Nursing Research*, 26(3), 307–320.

705 <https://doi.org/10.1177/0193945903260554>

706 Zwaan, R. A., Etz, A., Lucas, R. E., & Donnellan, M. B. (2018). Making replication

707 mainstream. *Behavioral and Brain Sciences*, *41*, e120.

708 <https://doi.org/10.1017/S0140525X17001972>

Table 1*Model Fit for RS-14 Example*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	126,722.491	126,888.707	0.934	0.923	0.094	0.036
Group Clinical	52,961.421	53,099.720	0.919	0.904	0.090	0.044
Group Student	69,100.985	69,254.310	0.928	0.915	0.108	0.035
Configural	122,118.406	122,617.055	0.926	0.912	0.102	0.036
Loadings	122,144.532	122,566.010	0.925	0.918	0.098	0.043
Intercepts	122,544.109	122,888.415	0.911	0.910	0.103	0.052
Residuals	126,466.241	126,727.438	0.780	0.793	0.156	0.086

Table 2*Boot Partial Effects Results for RS-14 Intercepts*

Term	Non-Invariant	Random Non-Invariant	h_{nmi}	h_{nmi_p}
RS Intercept	0.989	0.013	2.703	0.860
RS1 Intercept	0.979	0.013	2.622	0.835
RS10 Intercept	0.981	0.013	2.636	0.839
RS11 Intercept	0.988	0.013	2.694	0.857
RS12 Intercept	0.986	0.013	2.676	0.852
RS13 Intercept	0.984	0.013	2.659	0.847
RS14 Intercept	0.986	0.013	2.676	0.852
RS2 Intercept	0.971	0.012	2.580	0.821
RS3 Intercept	0.981	0.012	2.646	0.842
RS4 Intercept	0.989	0.012	2.712	0.863
RS5 Intercept	0.976	0.013	2.602	0.828
RS6 Intercept	0.970	0.013	2.565	0.816
RS7 Intercept	0.961	0.013	2.515	0.801
RS8 Intercept	0.986	0.013	2.676	0.852
RS9 Intercept	0.978	0.013	2.615	0.832

Note. Non-Invariant and Random Non-Invariant columns represent the proportion of non-invariant simulations of out the total simulations, representing our non-replication rate. These values are converted into an effect size difference in the final two columns.

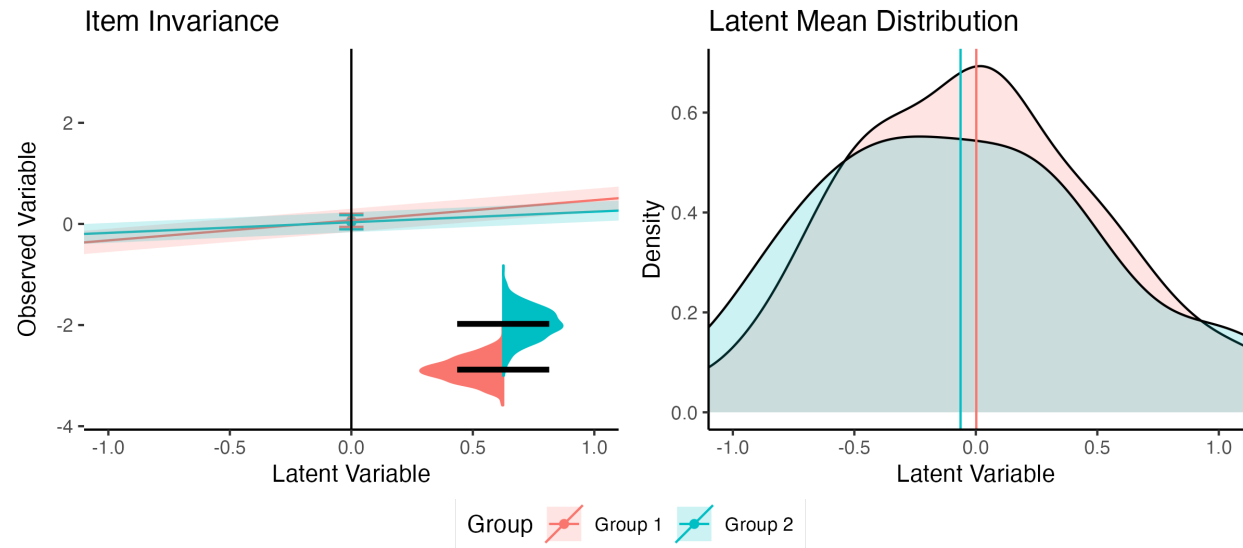


Figure 1

Invariant model visualization demonstrating the components of the `plot_mi()` function in `visualizemi`.

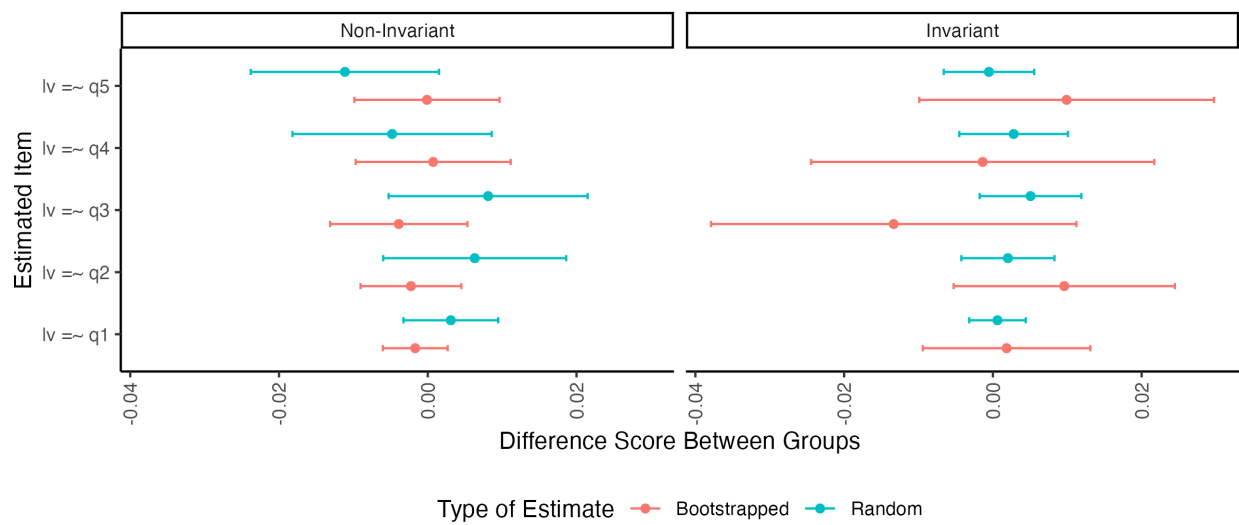


Figure 2

Visualization of the difference score between groups by parameter for invariant and non-invariant bootstrapped and randomly assigned group data on simulated data.

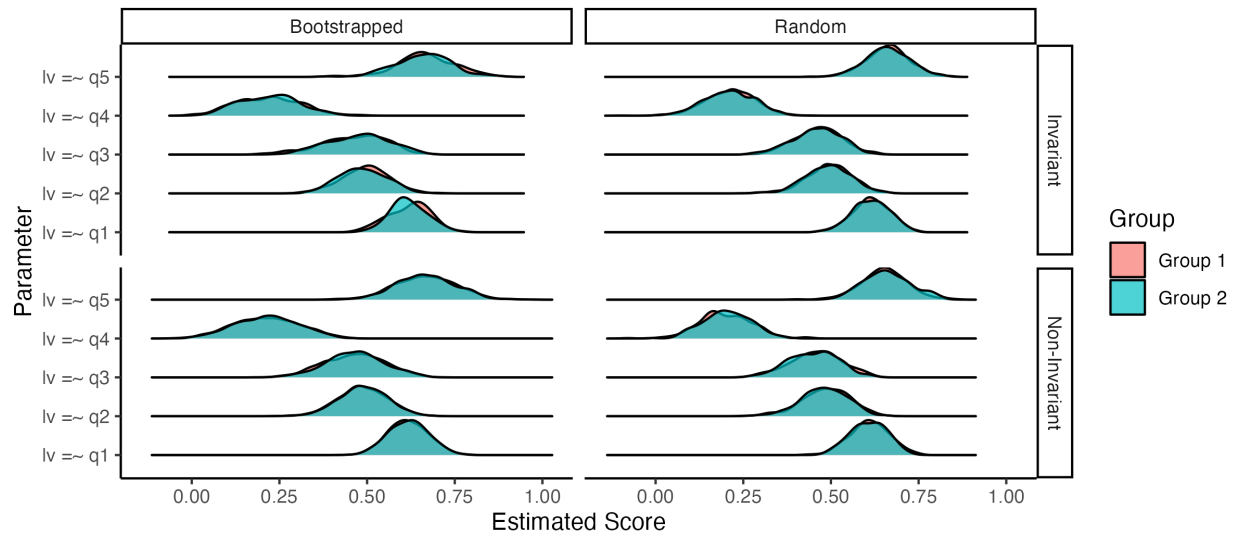
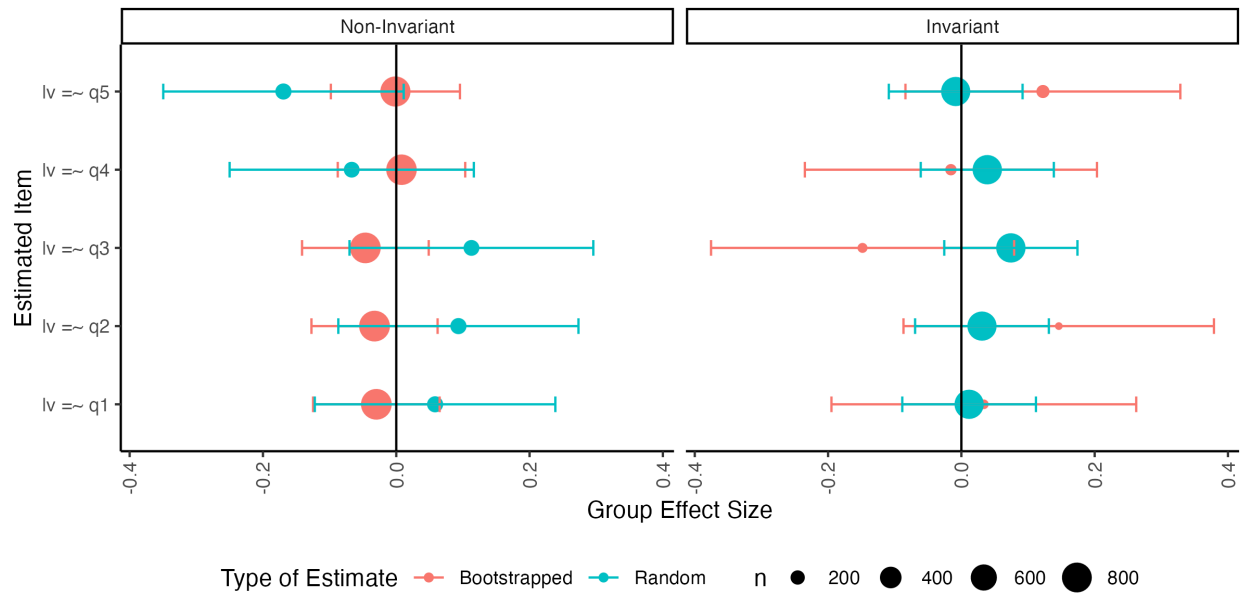
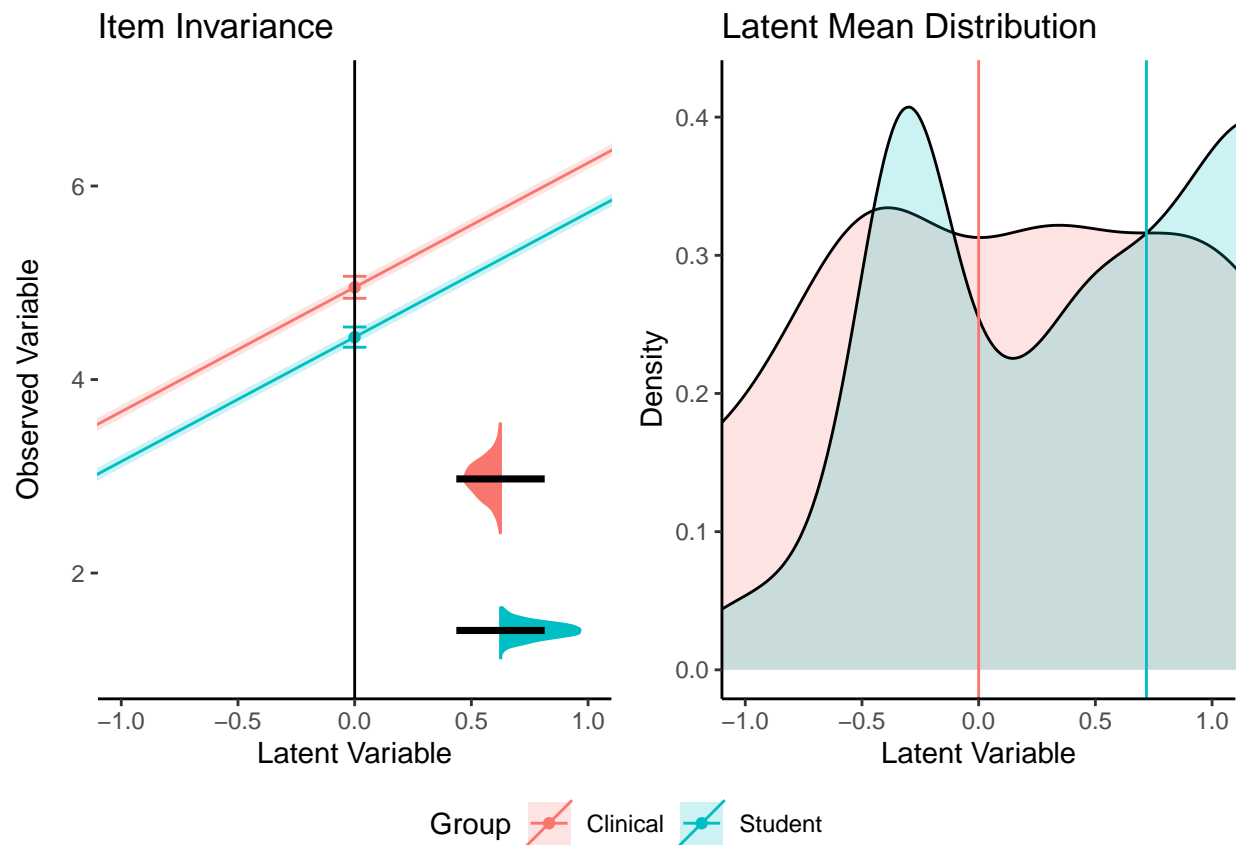


Figure 3

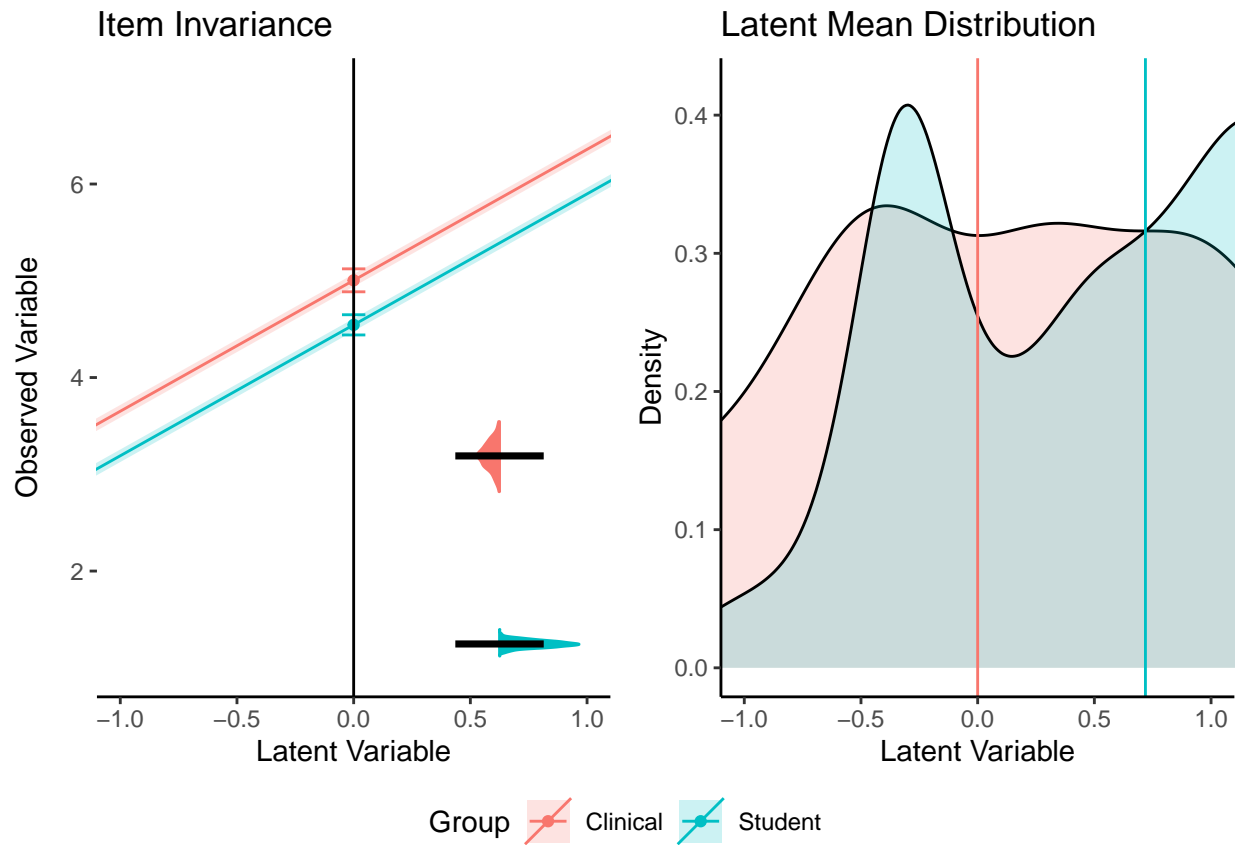
Visualization of the number of estimates for each group by bootstrapped and randomly assigned group runs by their invariance decision on simulated data.

**Figure 4**

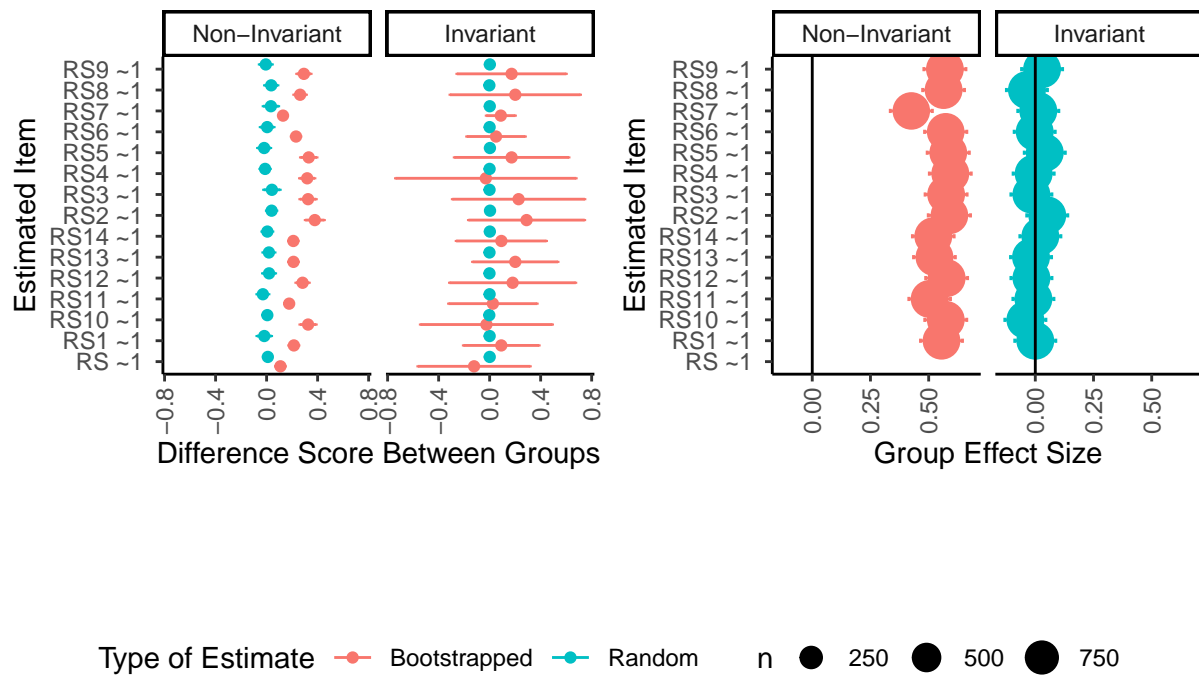
Visualization of effect size between groups by parameter for invariant and non-invariant bootstrapped and randomly assigned group data. The size of the dots indicate the number of data points for that estimate.

**Figure 5**

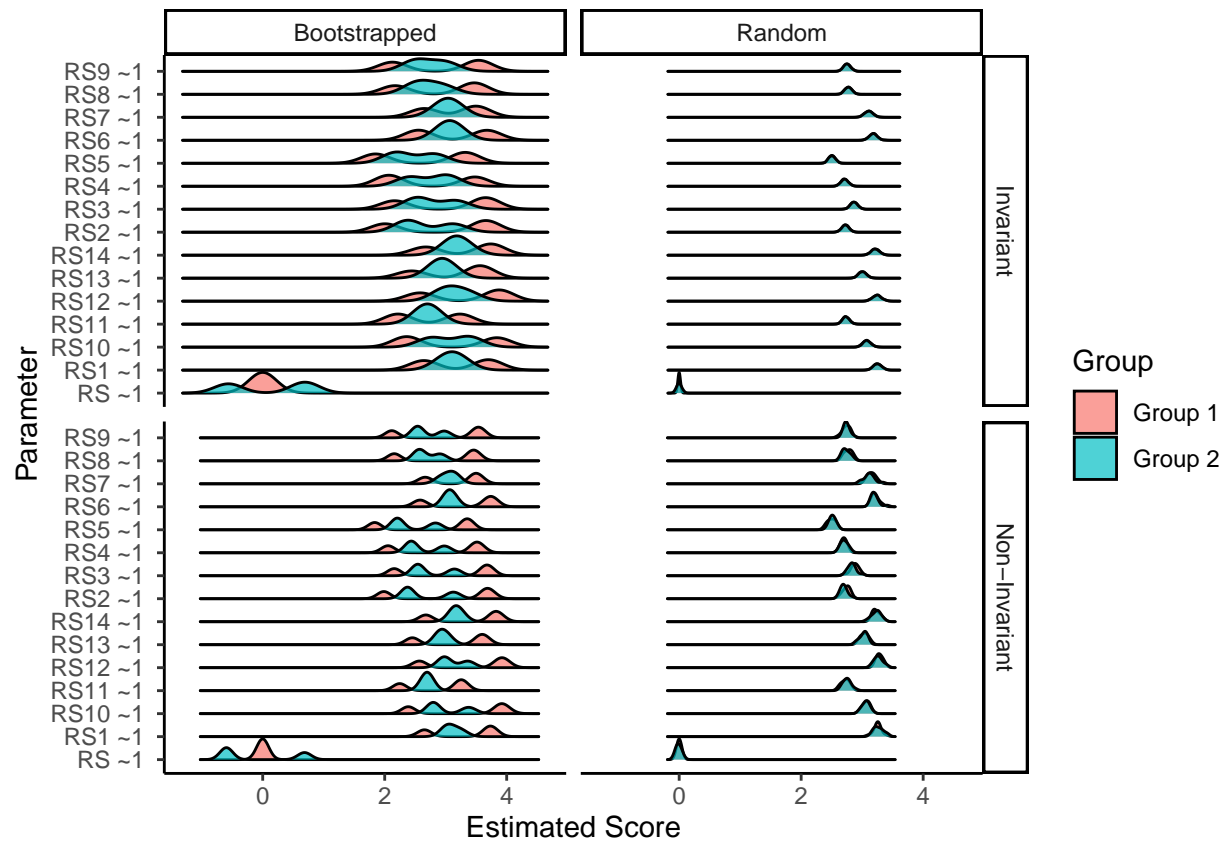
RS7 item non-invariance visualization showing differences in the item intercepts and latent variable.

**Figure 6**

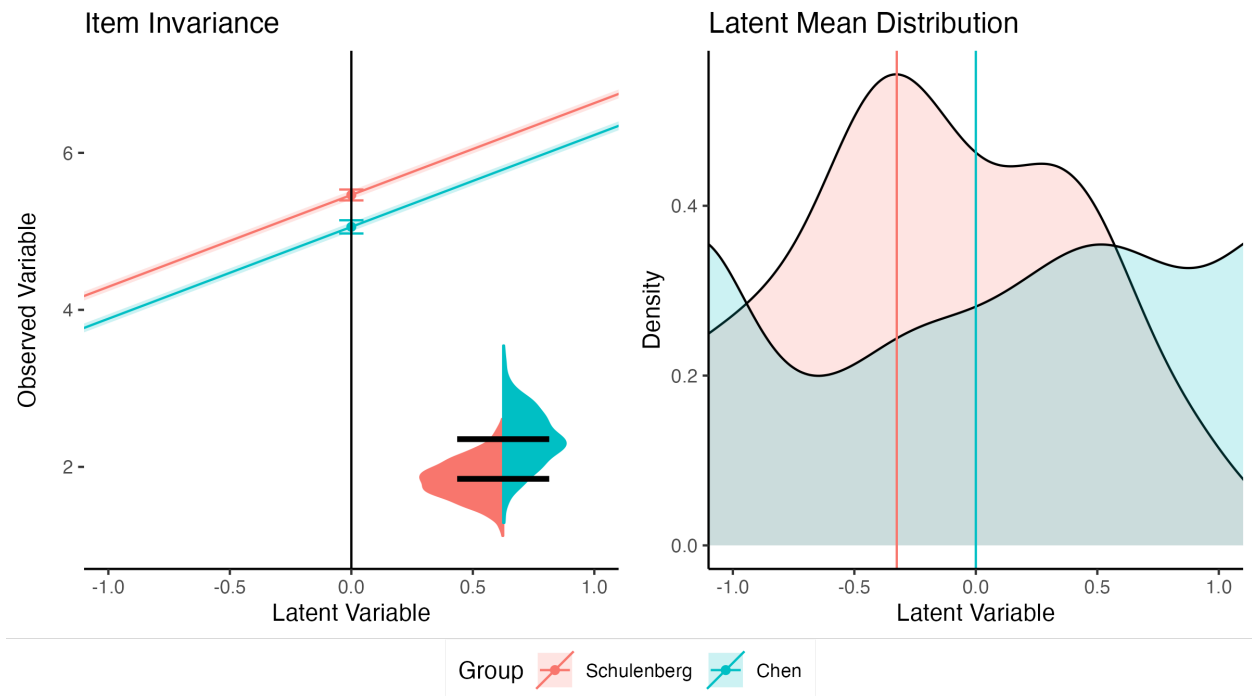
RS6 item non-invariance visualization similar results to RS7 with differences in item intercepts.

**Figure 7**

RS-14 scale invariance for item intercepts. The left panel indicates the raw score difference between groups and items, while the right panel indicates the effect size for group differences based on invariance.

**Figure 8**

RS-14 scale invariance density plots, illustrating invariant versus non-invariant bootstrapped and random runs for each parameter.

**Figure 9**

The differences in intercepts for the second item on the RS14 by language sample. The differences between intercepts are shown on the left-hand side with a clear separation between lines. The latent means also show a clear difference between groups where the English group appears to have lower scores overall than the Chinese group.

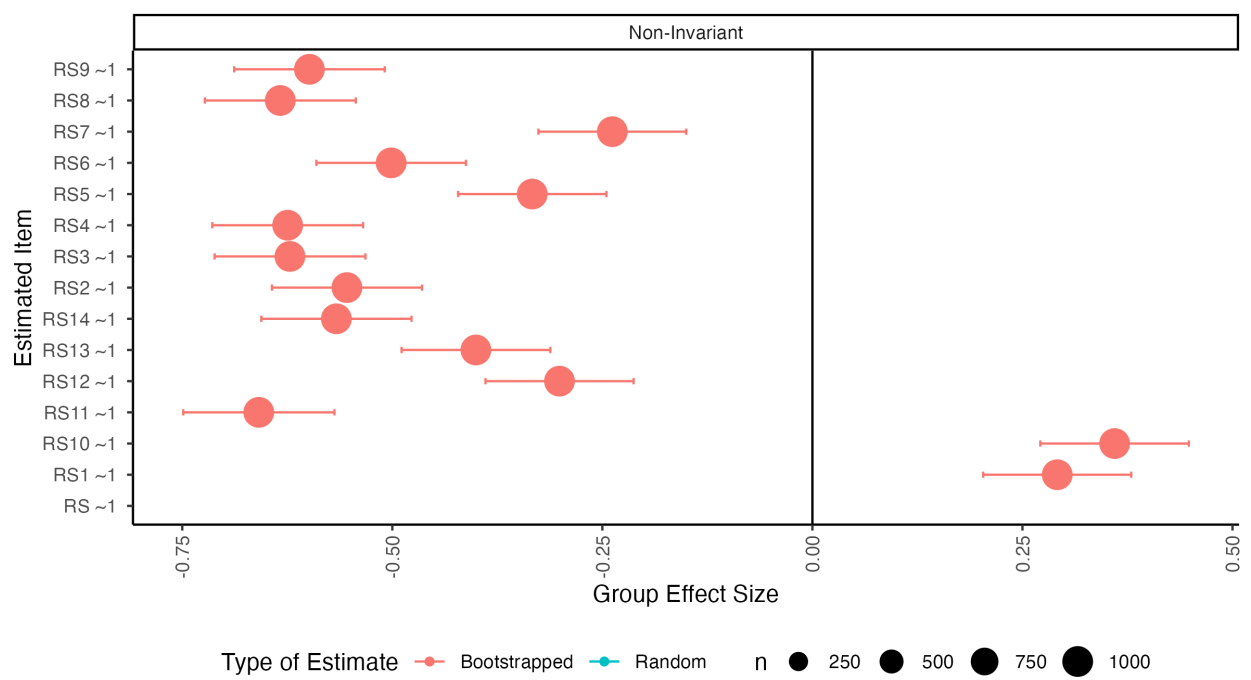


Figure 10

Effect size differences for item intercepts on the English versus Chinese samples for the RS14.

Appendix A

Code Examples

709 Simulating from Models

710 Here's an example of how to simulate directly from a `lavaan` model:

```
# first build your model
# this example is separate for each group
model.invariant.g1 <- "
# loadings
lv =~ .8*q1 + .4*q2 + .6*q3 + .3*q4 + .6*q5
# set the residual for invariance on q4
q4 ~~ 1*q4
# set the intercept for invariance on q4
q4 ~ 0*1
# set the intercept to zero for df purposes
q1 ~ 0*1
# allow the latent mean to be estimated
lv ~ 1"
model.invariant.g2 <- "lv =~ .77*q1 + .43*q2 + .58*q3 + .3*q4 + .61*q5
q4 ~~ 1*q4
q4 ~ 0*1
q1 ~ 0*1
lv ~ 1"

# simulate data invariant separately for each group
df.invariant <- bind_rows(
  # lavaan function
  simulateData(
    # model with estimates
```

```
model = model.invariant.g1,  
# how many data points  
sample.nobs = 250,  
# mean structure for mgcfa models  
meanstructure = T,  
# model type  
model.type = "cfa",  
# set seed for reproducibility  
seed = 1234) %>%  
# add a group label to the data  
mutate(group = "Group 1"),  
simulateData(  
  model = model.invariant.g2,  
  sample.nobs = 250,  
  meanstructure = T,  
  model.type = "cfa",  
  seed = 1234) %>%  
  mutate(group = "Group 2")  
)
```

711 **Simulating from Matrices**

712 Here's an example of how to simulate using **MASS** and covariance or correlation
713 matrices.

```
library(MASS)

# covariance matrix
university.cov <- lav_matrix_lower2full(
  c(169.00,
    73.710, 182.2500,
    73.229, 88.4250, 171.6100,
    63.375, 72.5625, 127.7250, 156.2500,
    42.120, 67.4325, 122.0265, 123.1875, 182.2500,
    57.226, 63.2610, 117.1926, 154.4250, 138.0240, 201.6400,
    30.875, 32.0625, 60.9805, 62.9375, 76.9500, 79.5910, 90.2500,
    36.075, 38.9610, 61.0722, 58.2750, 65.9340, 70.9290, 81.1965, 123.2100,
    18.096, 21.1410, 26.2131, 39.1500, 44.6310, 46.9452, 48.7635, 56.0106, 75.6900))

# give it names
rownames(university.cov) <-
  colnames(university.cov) <-
  c("class", "social", "learn", "chronic", "physical", "sex",
    "depression", "anxiety", "stress")

# means - you need standard deviation if you only have a correlation matrix
university.means <- c(3.4, 4.3, 3.7, 3.2, 4.5, 1.2, 4.0, 3.5, 4.2)

# use mass function
DF <- mvrnorm(n = 200, mu = university.means, Sigma = university.cov)
```

```
head(DF)
```

```

714 ##           class      social      learn  chronic  physical      sex
715 ## [1,]    9.332202  6.7442501   8.001600  6.771713 -10.833804  -3.837477
716 ## [2,]    4.873877 -0.1368433  11.690145 14.572699  13.356779  22.699918
717 ## [3,]   13.650244 -5.5841743  -2.295967 -3.914620  -6.552379 -11.711882
718 ## [4,]    7.644520 -1.6201790 -15.075033  4.010138   4.741793  11.704179
719 ## [5,]  -10.491240 14.6367273   2.951522 10.934949   1.153787   3.637487
720 ## [6,]    3.188521 -0.2078648 -11.655781 -8.085560 -12.482893 -18.914448
721 ##      depression    anxiety      stress
722 ## [1,]  -9.2071866  -9.0853468 -10.016408
723 ## [2,]  -5.7266089  -5.1086786  -3.757830
724 ## [3,] -12.3350189  -9.5855529 -10.855687
725 ## [4,]  -0.8850695  -8.8631134  -1.046865
726 ## [5,]  -3.4707411  -0.1493184  -2.300569
727 ## [6,]  -9.1500394   0.3167367   5.590050

```

MGCFA: `mgcfa()` Function

In this example, we make our example model using *lavaan* syntax. The `lv` latent variable predicts the five measured variables, which are present as columns in our `df.invariant` data set.

lavaan automatically sets the mean (i.e., the intercept) for latent variables to zero. If we wish to visualize the impact of the changes in parameter estimates across groups on the latent means, we need to allow the latent mean estimation with `lv ~ 1`. However, adding this estimation into our model will create a non-identified model. To solve this problem, you can set one of the intercepts of another variable to a value to scale the model. Here we will set the scale of the model by using `q1 ~ 0*1`, thus, scaling the expected means to zero. With simulation, this step is easy to know which variable to pick - we set the intercept on the variable we know did not show differences. In real data, you may wish to run the model steps *without* setting this option, examine the results of a configural or separate models, and then add the option for the values most similar. Additionally, you could complete partial invariance steps to determine which value appears most consistent to fix the estimate.

```
# create lavaan model
model.overall <- "
# overall one-factor model
lv =~ q1 + q2 + q3 + q4 + q5
# set the intercept (mean) of q1 to zero
q1 ~ 0*1
# allow the lv intercept to be freely estimated
lv ~ 1"
# look at the data
head(df.invariant)
```

##	q1	q2	q3	q4	q5	group
743						

```

744 ## 1 -0.8903542 -0.81707530  0.06137292 -1.3236407 -1.7916418 Group 1
745 ## 2  1.1054521 -0.03540948 -0.81299606  1.0028340 -0.1909127 Group 1
746 ## 3  1.4555852  1.54083484  1.59084213 -0.3345967 -0.6865496 Group 1
747 ## 4 -1.8745187 -1.27880245 -2.53565792 -1.0024193 -1.6253249 Group 1
748 ## 5 -0.4449517 -0.17782974  1.05507079 -1.2615705  1.7536428 Group 1
749 ## 6  0.2278813  0.71348845  1.63251893  0.6449847 -1.0055700 Group 1

```

750 The `mgcfa()` function is designed to flexibly allow you to leverage *lavaan*'s package
 751 functions to calculate multiple measurement steps at once. You would include:

- 752 1) the model syntax in the `model` argument
- 753 2) the dataframe in the `data` argument of our function
- 754 3) the name of the grouping variable in quotes for `group`
- 755 4) and the equality constraints you would like to impose in order in `group.equal`
- 756 5) ... any other *lavaan* arguments you would like to use such as `meanstructure` or
 757 `estimator`.

758 Note: you can also use `sample.cov`, `sample.mean`, `sample.nobs` in this step for
 759 estimation of multigroup models, but simulated dataframes are needed for bootstrapping
 760 replication estimates.

```

# run our mgcfa function to run all models
results.invariant <-

# name of the saved model syntax
mgcfa(model = model.overall,

      # name of the dataframe
      data = df.invariant,

      # name of the grouping variable
      group = "group",

      # equality constraints to impose in order

```



```

group.equal = c("loadings", "intercepts", "residuals"),
# other options to send to lavaan cfa function
meanstructure = T)

# what is saved for you
names(results.invariant)

```

```

761 ## [1] "model_coef"          "model_fit"          "model_overall"
762 ## [4] "group_models"        "model_configural"   "invariance_models"

```

763 1) `model_coef`: The parameter estimates for each model with the model step included in
 764 a *model* column. This set of coefficients can be used for other functions. This
 765 dataframe is created with *broom*'s `tidy()` function if you wish to recreate this table
 766 without running the `mgcfa()` function (Robinson et al., 2023).

```
results.invariant$model_coef[1:10 , ]
```

```

767 ## # A tibble: 10 x 12
768 ##   term      op estimate std.error statistic  p.value std.lv std.all model
769 ##   <chr>    <chr>   <dbl>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl> <chr>
770 ## 1 "lv =~ q1" =~      1        0      NA      NA      0.803   0.616 Over~
771 ## 2 "lv =~ q2" =~    0.655    0.0880    7.44 9.77e-14 0.526   0.493 Over~
772 ## 3 "lv =~ q3" =~    0.640    0.0895    7.15 8.83e-13 0.514   0.463 Over~
773 ## 4 "lv =~ q4" =~    0.277    0.0749    3.69 2.24e- 4 0.222   0.209 Over~
774 ## 5 "lv =~ q5" =~    0.955    0.117     8.13 4.44e-16 0.766   0.656 Over~
775 ## 6 "q1 ~1 "  ~1      0        0      NA      NA      0        0      Over~
776 ## 7 "lv ~1 "  ~1    -0.0305   0.0582   -0.524 6.00e- 1 -0.0380 -0.0380 Over~
777 ## 8 "q1 ~~ q1" ~~    1.05    0.0995   10.6 0        1.05    0.620 Over~
778 ## 9 "q2 ~~ q2" ~~    0.860    0.0653   13.2 0        0.860    0.757 Over~

```

```

779 ## 10 "q3 ~~ q3" ~~      0.966      0.0711      13.6      0      0.966      0.785 Over~
780 ## # i 3 more variables: block <int>, group <int>, label <chr>

```

2) `model_fit`: The model fit indices from `fitmeasures()` to review for overall model fit and invariance judgments. The name of the model is included in a *model* column.

```
head(results.invariant$model_fit)
```

```

783 ## # A tibble: 6 x 18
784 ##   agfi    AIC    BIC    cfi  chisq  npar  rmsea rmsea.conf.high    srmr    tli
785 ##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl>   <dbl> <dbl>
786 ## 1 0.998 7516. 7580. 1      0.650   15 0      0      0.00616 1.04
787 ## 2 0.948 3766. 3819. 0.976 7.79    15 0.0473 0.108 0.0312 0.953
788 ## 3 0.974 3768. 3820. 1      4.48    15 0      0.0831 0.0210 1.01
789 ## 4 0.961 7533. 7660. 0.991 12.3    30 0.0301 0.0785 0.0261 0.982
790 ## 5 0.965 7528. 7638. 0.994 15.4    26 0.0200 0.0660 0.0330 0.992
791 ## 6 0.969 7522. 7615. 1      17.3    22 0      0.0542 0.0352 1.00
792 ## # i 8 more variables: converged <lgl>, estimator <chr>, ngroups <int>,
793 ## #   missing_method <chr>, nobs <int>, norig <int>, nexcluded <int>, model <chr>

```

3) `model_overall`: A saved *lavaan* fitted model of all groups together without any equality constraints or grouping variables. These objects can be used with any function that normally takes a saved model: `parameterEstimates()`, `modificationIndices()`, `semPlot::semPaths()`, and so on (Epskamp, 2022).

```
class(results.invariant$model_overall)
```

```

798 ## [1] "lavaan"
799 ## attr(,"package")
800 ## [1] "lavaan"

```

801 4) `group_models`: A list of saved fitted models for each group separately.

```
names(results.invariant$group_models)
```

802 `## [1] "model.Group 1" "model.Group 2"`

803 5) `model_configural`: A saved fitted model for the configural model that nests together
804 each group into one model with no other constraints.

```
class(results.invariant$model_configural)
```

805 `## [1] "lavaan"`

806 `## attr(,"package")`

807 `## [1] "lavaan"`

808 6) `invariance_models`: A list of saved fitted models that consecutively adds
809 `group.equal` constraints.

```
names(results.invariant$invariance_models)
```

810 `## [1] "model.loadings" "model.intercepts" "model.residuals"`

Partial Invariance: `partial_mi()` Function

The `partial_mi()` function aids in the calculation of partial invariance for a specific step of the MGCFA process. The function includes the following arguments:

- 1) `saved_model`: The saved *lavaan* model with the equality constraints at the level of measurement invariance you would like to examine for partial invariance.
- 2) `data`: The dataframe where the model was estimated.
- 3) `model`: The model syntax for the overall model.
- 4) `group`: The grouping variable column in the dataframe.
- 5) `group.equal`: The equality constraints including in your original multigroup tests.
- 6) `partial_step`: The level of partial invariance you wish to test.

```
partial.invariant <-
  partial_mi(
    # saved model output with constraints
    saved_model = results.invariant$invariance_models$model.residuals,
    # dataframe from model
    data = df.invariant,
    # model syntax
    model = model.overall,
    # group column name
    group = "group",
    # group equality constraints from your mgcfa
    group.equal = c("loadings", "intercepts", "residuals"),
    # which step you want to examine for partial invariance
    partial_step = "residuals"
  )

names(partial.invariant)
```

```
821 ## [1] "models"      "fit_table"
```

822 In this function, each parameter with the appropriate *lavaan* syntax is relaxed
 823 individually (i.e., ~1 for intercepts, ~~ for residuals, etc.). The fitted models are saved in the
 824 `models` output, and the `fit_table` output includes all fit indices for each model to
 825 investigate potential areas of partial invariance based on the researcher's desired criterion.

```
names(partial.invariant$models)
```

```
826 ## [1] "q1 ~~ q1" "q2 ~~ q2" "q3 ~~ q3" "q4 ~~ q4" "q5 ~~ q5" "lv ~~ lv"
```

```
head(partial.invariant$fit_table %>%  
      dplyr::select(free.parameter, cfi, rmsea))
```

```
827 ## # A tibble: 6 x 3  
828 ##   free.parameter cfi      rmsea  
829 ##   <chr>         <lvn.vctr> <lvn.vctr>  
830 ## 1 q1 ~~ q1      0.9902679  0.02108648  
831 ## 2 q2 ~~ q2      0.9868905  0.02447336  
832 ## 3 q3 ~~ q3      0.9958241  0.01381266  
833 ## 4 q4 ~~ q4      1.0000000  0.00000000  
834 ## 5 q5 ~~ q5      0.9868088  0.02454944  
835 ## 6 lv ~~ lv      0.9906154  0.02025143
```

836 Note: the `partial_step` function is used to determine which types of `op` or
 837 operators to freely estimate between groups. If one chooses residuals, you will also freely
 838 estimate the residual for the latent variable or any other residuals found in the model. These
 839 items may be ignored if they were not meant to be included.

Visualization of Invariance: `plot_mi()` Functions

Once we know which items are non-invariant, the `model_coef` output from the `mgcfa()` can be used directly in `plot_mi()`. The plot outputs will be described below. First, here are the arguments for the function:

- 1) `data_coef`: A tidy dataframe of the parameter estimates from the models. This function assumes you have used `broom::tidy()` on the saved model from *lavaan* and added a column called “model” with the name of the model step (Robinson et al., 2023). This function will only run for models that have used the grouping function (i.e., configural, metric, scalar, and strict or other combinations/steps you wish to examine).
- 2) `model_step`: Which model do you want to plot? You should match this name to the one you want to extract from your model column in the `data_coef`.
- 3) `item_name`: Which observed variable from your model syntax do you want to plot? Please list this variable name exactly how it appears in the model.
- 4) `x_limits`: What do you want the x-axis limits to be for your invariance plot? The default option is to assume the latent variable is standardized, and therefore, -1 to 1 is recommended. Use only two numbers, a lower and upper limit. This value also constrains the latent mean diagram to help zoom in on group differences because the scale of latent means is usually centered over zero. You can use this parameter to zoom out to a more traditional histogram using `c(-2, 2)`.
- 5) `y_limits`: What do you want the y-axis limits to be for your invariance plot? Given that the latent variable is used to predict the observed values in the data, you could use the minimum and maximum values found in the data. If that range is large, consider reducing this value to be able to visualize the results (i.e., otherwise it may be too zoomed out to judge group differences). Use only two numbers, a lower and upper limit.
- 6) `conf.level`: What confidence limit do you want to plot? Use $1 - \alpha$.

7) `model_results`: In this argument, include the saved *lavaan* output for the model listed in the `model_step` argument.

8) `lv_name`: Include the name of the latent variable, exactly how it is listed in your *lavaan* syntax. You should plot the latent variable that the `item_name` is linked to. If you have items that load onto multiple latent variables, you will need to make multiple plots.

9) `plot_groups`: If you include more than two groups in a multigroup model, the automatic assumption is that you want the first two groups for this visualization. If not, include the names of the groups here to plot.

```
invariant.plot <-
  plot_mi(
    # output from model_coef
    data_coef = results.invariant$model_coef,
    # which model do you want to plot
    model_step = "Configural",
    # name of observed item
    item_name = "q4",
    # latent variable limits to graph
    x_limits = c(-1,1),
    # Y min and max in data
    y_limits = c(min(df.invariant$q4), max(df.invariant$q4)),
    # what ci do you want
    conf.level = .95,
    # what model results do you want
    model_results = results.invariant$model_configural,
    # which latent variable do you want
    lv_name = "lv"
  )
```

```
names(invariant.plot)
```

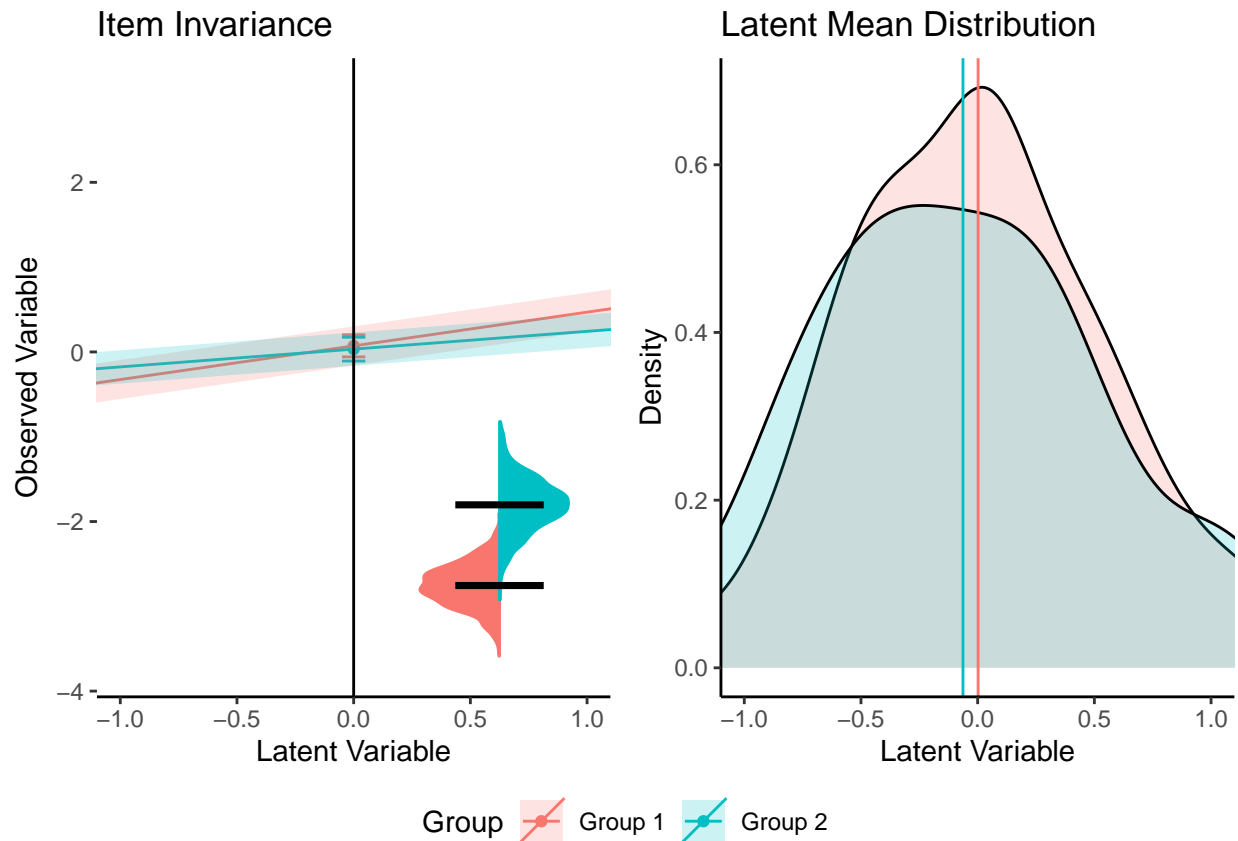
```
875 ## [1] "complete" "intercept" "mean" "variance"
```

876 The outputs from this function are several *ggplot2* objects that can be edited or saved
877 directly using *ggplot2* functionality (Wickham, 2016).

878 1) **complete**: The output from this model can be found in Figure 1. On the left-hand side,
879 the item invariance is plotted, and on the right-hand side, the latent mean distributions
880 for the two groups are plotted. In the item invariance sub-plot, the visualization
881 includes all three components traditionally seen in MGCFA testing steps: loadings,
882 intercepts, and residuals. Each visualization element was designed to match the
883 traditional visualization for that type of output. All parameter estimates are plotted
884 on the unstandardized estimates and their confidence interval based on the standard
885 error of the estimate. All plots are made with *ggplot2* and *cowplot* (Wilke, 2020).

886 2) **intercept**: Only the left-hand side of the complete plot designed to represent
887 intercepts and factor loadings. Factor loadings represent the slope of the regression
888 equation for the latent variable predicting the scores on the observed variable
889 ($\hat{Y} \sim b_0 + b_1X + \epsilon$). The y-axis indicates the observed variable scores, and here, the
890 plot includes the entire range of the scale of the data for item four. The coefficient (b_1)
891 for group 1 was 0.40, while the coefficient for group 2 was 0.21. The ribbon bands
892 around the plotted slopes indicate the confidence interval for that estimate. In this
893 plot, while the coefficients for each group are not literally equal, the overlapping and
894 parallel slope bands indicate they are not different practically.

895 The item intercepts (b_0) are plotted on the middle line where they would cross the
896 y-axis at a latent variable score of zero. These are represented by a dot with a set of

**Figure A1***Invariant Model Visualization*

confidence error bars around the point. The intercept for group 1 was 0.07, while the coefficient for group 2 was 0.03. In this invariant depiction, the overlap in the intercepts is clear, indicating they are not different. You can use `y_limits` to zoom in on the graph if these are too small to be distinguishable.

- 3) **mean:** The right-hand side of the complete plot graphing the latent variable means and density from the data. The latent variable is shown on the x-axis using standardized values (i.e., z-scores) where -1 indicates one standard deviation below the mean for the latent variable, 0 indicates the mean for the latent variable and so on. The lines indicate the means of the latent variables from the simulated dataset. Group labels are represented in the figure caption on the bottom. Group 1 is usually the group that is

907 alphabetically first in the data set or whichever group is the first that appears when
908 using the `levels()` command.

909 4) **variance:** A split geom violin plot indicating the variance distribution of the plotted
910 item. Residuals are trickier to plot, as they are the left over error when predicting the
911 observed variables ϵ . It is tempting to plot this value as the confidence band around
912 the slope, however, that defeats the purpose of understanding that the slopes are
913 estimated separately from the residuals, and both have an associated variability around
914 their parameter estimate. Therefore, residuals are represented in the inset picture at
915 the bottom right of the item invariance plot. The black bars represent the estimated
916 residual for each group (group 1: 0.91, group 2: 1.22). The distributions are plotted to
917 represent the normal spread of values using the standard error of the residuals. The
918 violin plot allows for direct comparison of those residuals and their potential
919 distributions. Note that the placement has nothing to do with the x or y-axis and is
920 designed to always show in the same location, regardless of size/value. The plots are
921 included separately so they can be arranged in a different fashion if desired.

Model Replication and Effect Sizes: `bootstrap_model()` Function

The `bootstrap_model` function in *visualizemi* was designed to estimate the likely replication of overall model invariance with the assumption that the data used for the estimation represents the larger population. The following arguments are used:

- 1) `saved_configural`: a saved fitted model at the configural level with no equality constraints. This model should include all other lavaan settings you would like to use, such as estimator or ordered.
- 2) `data`: The dataframe where the model was estimated.
- 3) `model`: The model syntax for the overall model.
- 4) `group`: The grouping variable column in the dataframe.
- 5) `nboot`: The number of bootstraps to run.
- 6) `invariance_index`: The fit index you would like to use to determine invariance. Please use options and labeling from *lavaan* - see `fitmeasures()` for options.
- 7) `invariance_rule`: The invariance difference score you would like to use as your rule.
- 8) `group.equal`: The equality constraints including in your original multigroup tests.

```
boot.model.invariant <-  
  bootstrap_model(  
    # saved configural model  
    saved_configural = results.invariant$model_configural,  
    # dataframe  
    data = df.invariant,  
    # model syntax  
    model = model.overall,  
    # group variable column in dataframe  
    group = "group",  
    # number of bootstraps  
    nboot = 1000,
```

```

# which fit index you would like to use
invariance_index = "cfi",
# what is your criterion for that fit index
invariance_rule = .01,
# what equality constraints are you testing
group.equal = c("loadings", "intercepts", "residuals")
)

```

937 The data included in this function will be sampled, with replacement, at the same
 938 size as the current dataset, and the included invariance equality constraints are estimated.
 939 Each step will be compared to the previous step using the invariance index and comparison
 940 rule entered. The output is a dataframe of the proportion of non-invariant bootstraps from
 941 the real data and the same bootstrapped dataset with the group labels randomly assigned.
 942 The effect size comparison of proportions, h , for non-invariant comparisons:

$$h_{nmi} = 2 \times (\text{asin}\sqrt{p_{data}} - \text{asin}\sqrt{p_{random}})$$

943 The alternative, h_{mi} , for effect size of measurement invariance replication would
 944 simply be the inverse sign of h_{nmi} and is also included in the table. Two additional columns
 945 h_{nmi_p} and h_{nmi_p} represent the h values divided by the upper bound of h (i.e., π), to help
 946 with interpretation of the effect size (thus, bounding h to -1 to 1).

Parameter Replication and Effect Sizes: `bootstrap_partial()` Function

After examining the overall model potential replication effect size, the individual parameters within a model can be bootstrapped for partial invariance to with that parameter relaxed (overall partial model statistics) and the difference in group parameter estimates (parameter effect size). This function uses arguments seen in other functions, so they will not be repeated here. The general setup consists of using the model you think could be partially invariant in the `saved_model` argument and the fit index for comparison for the model with less constraints in `invariance_compare`. This example examines the loadings in the invariant model, so `saved_model` uses the `mgcfa` output for equality constraints present on the loadings and compares that model to the configural model with no equality constraints on the loadings. The `partial_step` argument will be used to determine which operation syntax (i.e. `=~` for loadings) to relax for modeling.

```
boot.partial.invariant <-
  bootstrap_partial(
    # saved model you want to examine the partial loadings for
    saved_model = results.invariant$invariance_models$model.loadings,
    # the dataset
    data = df.invariant,
    # the model
    model = model.overall,
    # the group variable in the dataset
    group = "group",
    # number of bootstraps
    nboot = 1000,
    # which fit index you would like to use to determine partial invariance
    invariance_index = "cfi",
    # what is the invariance rule
    invariance_rule = ".01",
```

```

# what are we comparing the saved model fit index to
invariance_compare = fitmeasures(results.invariant$model_configural, "cfi"),
# what step are we using for invariance
partial_step = "loadings",
# what equality constraints should be imposed
group.equal = c("loadings")
)

```

```
names(boot.partial.invariant)
```

```

959 ## [1] "invariance_plot"          "effect_invariance_plot" "density_plot"
960 ## [4] "boot_DF"                 "boot_summary"          "boot_effects"

```

961 The saved output includes several dataframes and plots. The first is the `boot_DF`
 962 which the summary of each run in a dataframe for plotting or summarization. This
 963 dataframe includes the estimate for each parameter (`term`) separated by group and type
 964 (`boot_1`, `boot_2` are the bootstrapped estimates for group 1 and group 2, while the same
 965 `random` columns indicate the randomly assigned groups). The fit index used to determine
 966 invariance is included for bootstrapped and random estimates, and then the differences
 967 between groups and if they were “invariant” or not given the researcher supplied rule.

```
head(boot.partial.invariant$boot_DF)
```

```

968 ##      term    boot_1    boot_2  random_1  random_2  boot_fit random_fit
969 ## 1 lv =~ q1 0.4548783 0.49928877 0.4627486 0.4651391 0.9296990 1.0000000
970 ## 2 lv =~ q2 0.3599017 0.56241016 0.4100874 0.4980215 0.9441125 1.0000000
971 ## 3 lv =~ q3 0.4254283 0.33640233 0.4274329 0.3422124 0.9377130 1.0000000
972 ## 4 lv =~ q4 0.3930716 0.03320619 0.1380833 0.2628802 0.9750274 1.0000000
973 ## 5 lv =~ q5 0.7306414 0.73512673 0.7093891 0.7532471 0.9266587 1.0000000
974 ## 6 lv =~ q1 0.5537083 0.57086815 0.5732166 0.5475714 0.8958929 0.9814658

```

```

975 ##      boot_difference random_difference boot_index_difference
976 ## 1      -0.044410454      -0.002390463      FALSE
977 ## 2      -0.202508484      -0.087934027      FALSE
978 ## 3       0.089025927       0.085220565      FALSE
979 ## 4       0.359865463      -0.124796846      FALSE
980 ## 5      -0.004485377      -0.043857947      FALSE
981 ## 6      -0.017159815       0.025645271      FALSE
982 ##      random_index_difference
983 ## 1                          TRUE
984 ## 2                          TRUE
985 ## 3                          TRUE
986 ## 4                          TRUE
987 ## 5                          TRUE
988 ## 6                          TRUE

```

989 Next, the `boot_summary` includes a summarized form of the bootstrapped results
 990 from separated by bootstrapping versus random and invariant/non-invariant. The d_s for
 991 between groups Cohen's d is shown below, and the non-central confidence interval is
 992 included. Effect sizes are only calculated when the number of bootstrapped estimates is at
 993 least 10% of the data - therefore, you would not receive effect sizes with almost no
 994 bootstrapped runs. This dataframe should be used to determine which parameter may be
 995 different and at what size between groups in a replication of the study.

```

boot.partial.invariant$boot_summary %>%
  dplyr::select(term, d_boot, d_random)

```

```

996 ##      term      d_boot      d_random
997 ## 1  lv =~ q1 -0.029853316  0.058271662
998 ## 2  lv =~ q1  0.033742666  0.011640524

```

```

999 ## 3 lv =~ q2 -0.032613505 0.093288563
1000 ## 4 lv =~ q2 0.146200211 0.030925365
1001 ## 5 lv =~ q3 -0.046329761 0.112823117
1002 ## 6 lv =~ q3 -0.148330246 0.074265298
1003 ## 7 lv =~ q4 0.007851687 -0.066844445
1004 ## 8 lv =~ q4 -0.015702705 0.038884725
1005 ## 9 lv =~ q5 -0.001285809 -0.169307690
1006 ## 10 lv =~ q5 0.122405579 -0.008526218

```

1007 The `boot_effects` table creates a summary similar to the overall model replication
 1008 table based on the proportion of runs that were considered invariant versus not for each
 1009 parameter. Note that the effects match the overall results, such that simulated invariant
 1010 data appears to still show the likelihood that loadings may not replicate in a similar dataset.

```
boot.partial.invariant$boot_effects
```

```

1011 ##          term non_invariant random_non_invariant    h_nmi    h_mi    h_nmi_p
1012 ## 1 lv =~ q1          0.853          0.236 1.340078 -1.340078 0.4265601
1013 ## 2 lv =~ q2          0.858          0.237 1.351946 -1.351946 0.4303378
1014 ## 3 lv =~ q3          0.851          0.230 1.348639 -1.348639 0.4292851
1015 ## 4 lv =~ q4          0.840          0.229 1.320578 -1.320578 0.4203530
1016 ## 5 lv =~ q5          0.819          0.237 1.245789 -1.245789 0.3965468
1017 ##          h_mi_p
1018 ## 1 -0.4265601
1019 ## 2 -0.4303378
1020 ## 3 -0.4292851
1021 ## 4 -0.4203530
1022 ## 5 -0.3965468

```


Plots of the results from dataframes can be found within the `bootstrap_partial()` function. Figure 2 shows the difference between parameters for groups in the bootstrapped and randomly assigned group runs. Figure 3 shows the density plot of the estimates for each group organized by bootstrapped and randomly assigned groups and the invariance decision for each bootstrapped run. Last, Figure 4 indicates the d_s value between groups with an indication of the number of data points in each estimate (i.e., dot size). These visualizations should allow a researcher to understand the likelihood of replication for each parameter, as well as the potential size of the differences. Therefore, one could indicate a specific smallest effect size of interest, rather than a invariance cut-off rule of thumb when planning a replication or registered report.

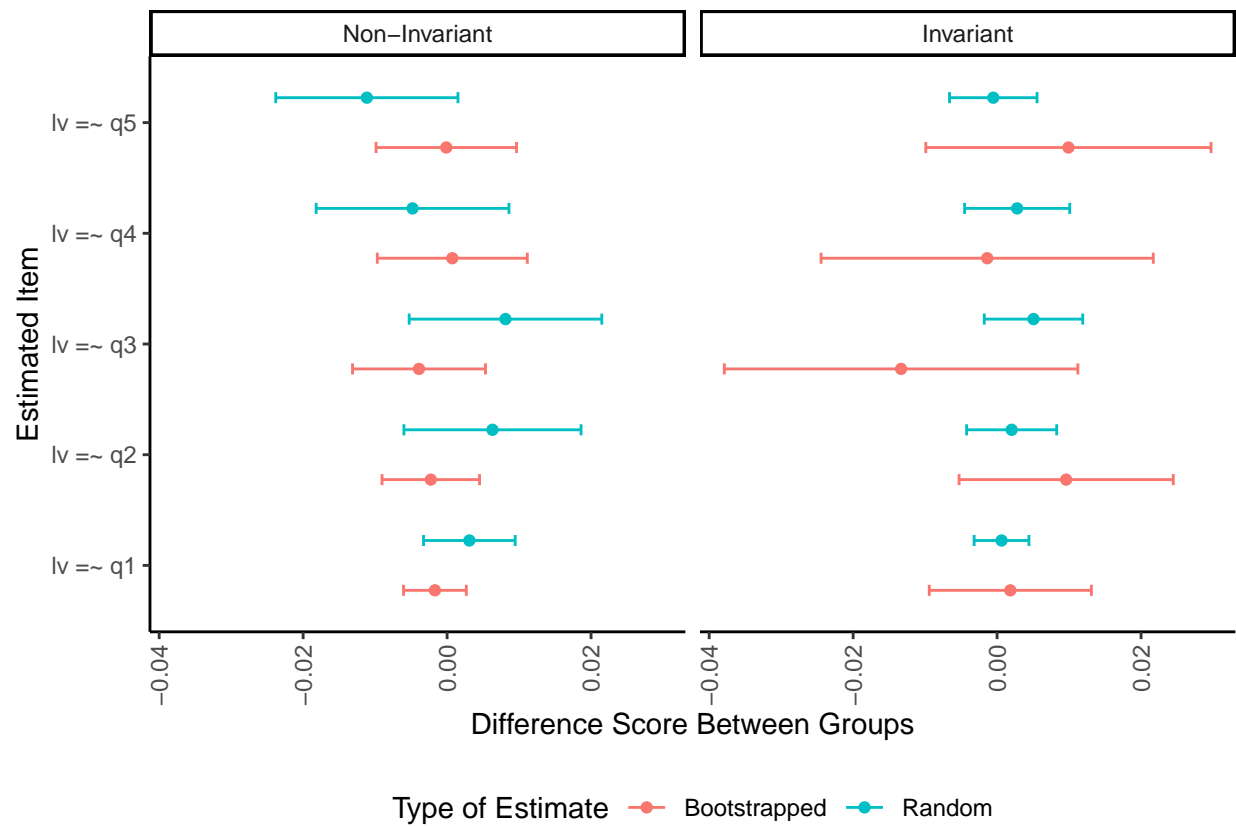


Figure A2

Visualization of the difference score between groups by parameter for invariant and non-invariant bootstrapped and randomly assigned group data.

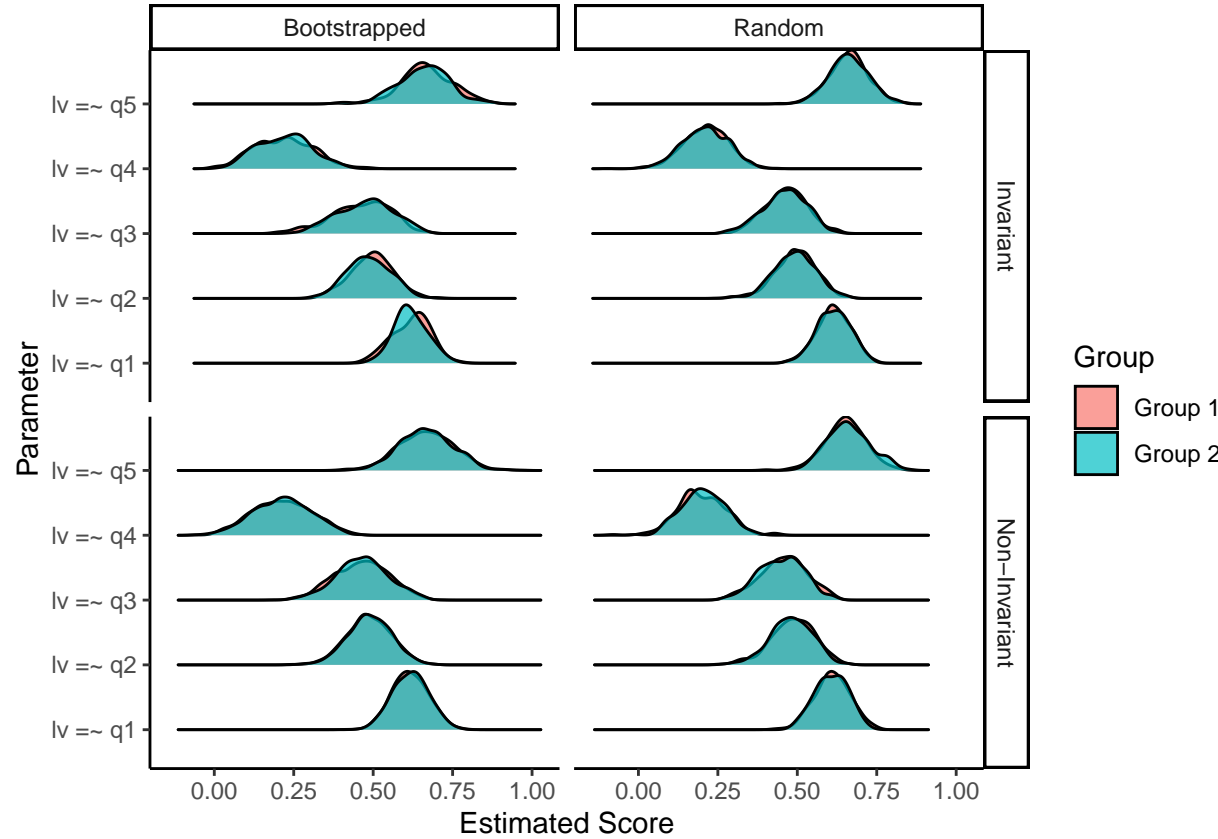
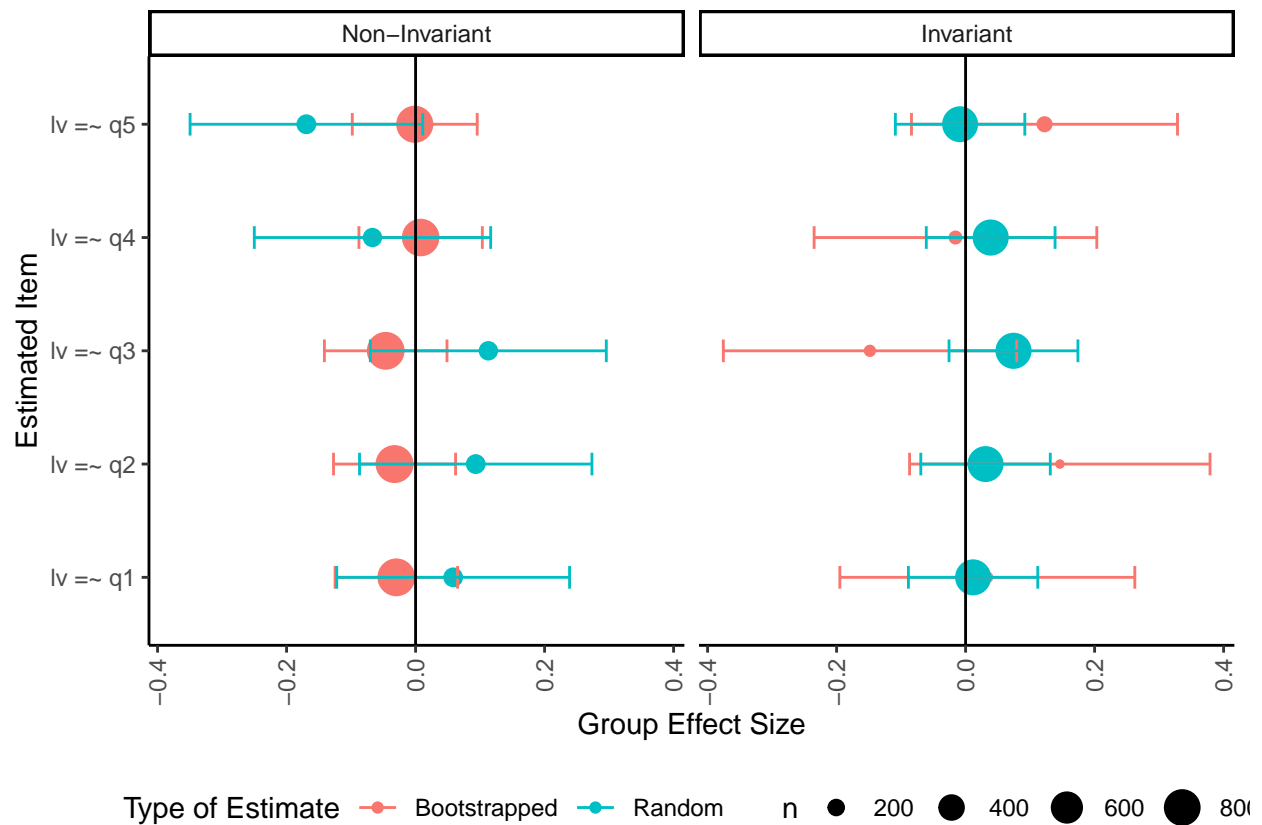


Figure A3

Visualization of the number of estimates for each group by bootstrapped and randomly assigned group runs by their invariance decision.

**Figure A4**

Visualization of effect size between groups by parameter for invariant and non-invariant bootstrapped and randomly assigned group data. The size of the dots indicate the number of data points for that estimate.

Appendix B

Simulation Study

The code for building and running simulations can be found in *simulate_boot_rr.Rmd*,
simulate_boot_partial.Rmd, and *simulate_combine.Rmd*.

Design and Analysis

Data was simulated using the `simulateData` function in the *R* package *lavaan* (Rosseel, 2012) assuming multivariate normality using a μ of 0 and σ of 1 for the data. This function allows you to write *lavaan* syntax for your model with estimated values to generate data for observed variables (see supplemental for examples). The data included two groups of individuals (“Group 1”, “Group 2”) for a multigroup confirmatory factor analysis ($n_{group} = 250$, $N = 500$). The latent variables were assumed to be continuous normal (the package functions do not require this assumption). The model consisted of five observed items predicted by one latent variable ($1v \sim q1 + q2 + q3 + q4 + q5$); however, the demonstration in this manuscript extends to multiple latent variables and other combinations of observed variables. Each item was assumed to be related to the latent variable with loadings approximately equal to .40 to .80, except when cases of non-invariance on the loadings was simulated.

The Brown (2015) steps of testing measurement invariance are demonstrated in this manuscript for illustration purposes, but in line with Stark et al. (2006) suggestions, the visualizations show the impact of loadings and intercepts together. A convenience function `mgcfa` is used for these steps or other measurement invariance test orders and combinations. Fit indices for the steps for multigroup models are presented in the appendix for comparison of cutoff rules of thumb (Cheung & Rensvold, 2002) to effect sizes and visualizations presented in this manuscript. Fit indices include Akaike Information Criterion (AIC, Akaike, 1998), Bayesian Information Criterion (BIC, Schwarz, 1978), Comparative Fit Index (CFI, Bentler, 1990), Tucker Lewis Index (TLI, Tucker & Lewis, 1973), root mean squared error of

approximation RMSEA (Steiger, 1990), and standardized root mean square residual (SRMR, Bentler, 1995).

The data was then simulated to represent invariance across all model steps, small, medium, and large invariance using d_{MACS} estimated sizes from Nye et al. (2019). While d_{MACS} is used primarily for an effect size of the (non)-invariance for intercepts and loadings together, a similar approach was taken for the estimation of small, medium, and large effects on the residuals. The effect size is presented for all models, calculated from the *dmacs* package (Dueber, 2023; Nye & Drasgow, 2011). Only one item in each model was manipulated from the invariant model to create the non-invariant models. Given the data was simulated with a z -score scaling, the loading values were simulated at .30 points apart (given d_{MACS} suggestions of .2, .4, .7), the intercepts at .25 points apart, and the residuals at .25 points apart. To plan a simulation for your own study, these values can be used to simulate small, medium, and large non-invariance effects by first converting data into z -score.

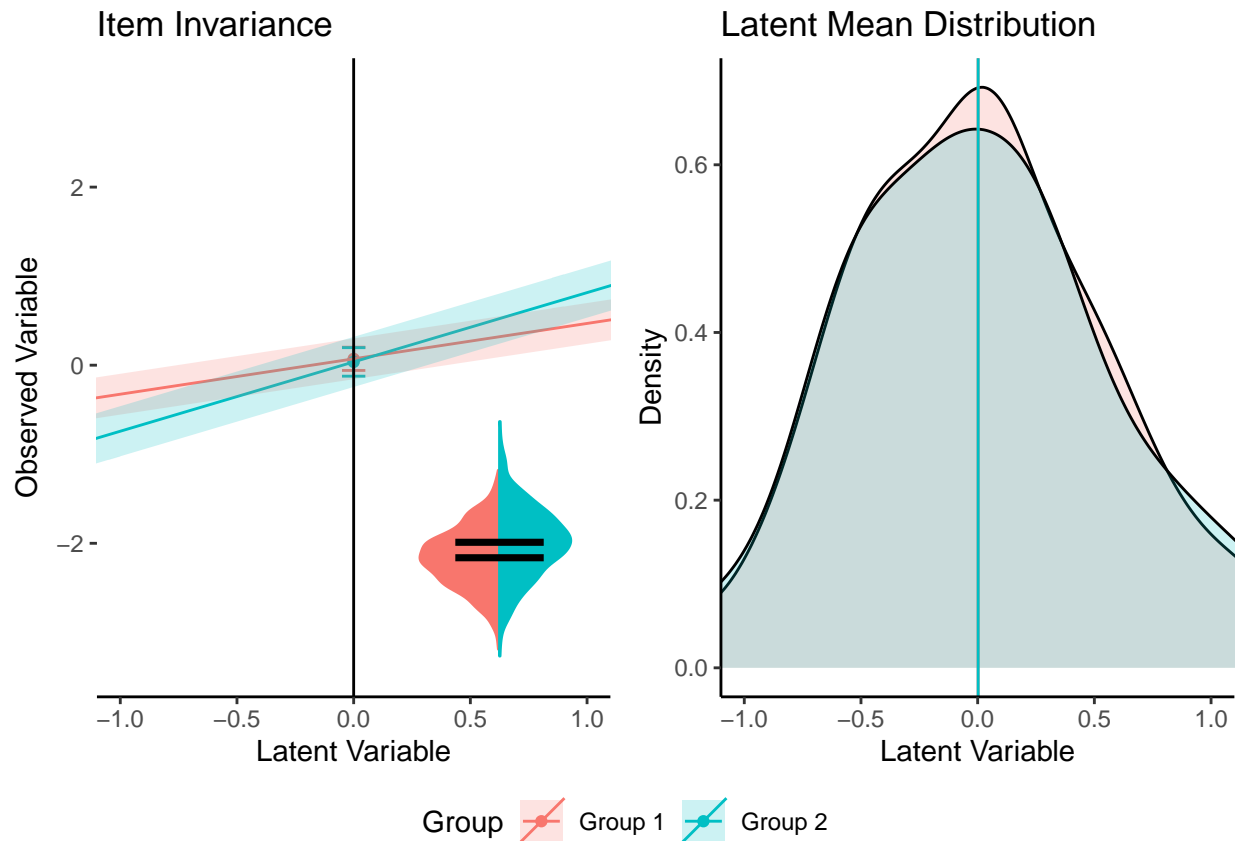
Visualize Parameter Differences

The d_{MACS} value for item 4 in the invariant model was 0.16, representing a nil or unimportant difference in this manuscript. It is important to note that while Nye et al. (2019) suggests specific sizes for small, medium, and large, each researcher should determine for themselves what effects represent. Figure B1 displays the results from the small ($d_{MACS} = 0.27$) difference in loadings, while Figure B2 displays the results from the medium ($d_{MACS} = 0.61$) difference in loadings, and Figure B3 shows the large ($d_{MACS} = 0.66$) differences. When investigating the slope values, we can clearly see the change in the loading for the second group (the only manipulated variable, although random data set generation may also change intercepts and residuals slightly). At the medium effect size, we see that the confidence bands do not overlap (at the edges), and at the large effect size, we can see a clear separation of two lines. Note that the intercepts in this model are estimated as equal so the loading representation will not literally separate, but the steepness of the lines is the

indicator of the difference between the slopes. You can imagine these lines are interpreted like a simple slopes analysis for interactions in regression (Cohen et al., 2003). When simple slopes for interactions are plotted, if they are parallel, there is no interaction, and if they cross, then there is an interaction. Here, we can use this same logic. If they are parallel, there is likely invariance (they are the same), and the further from parallel they become, the larger the effect size for the differences between group loadings.

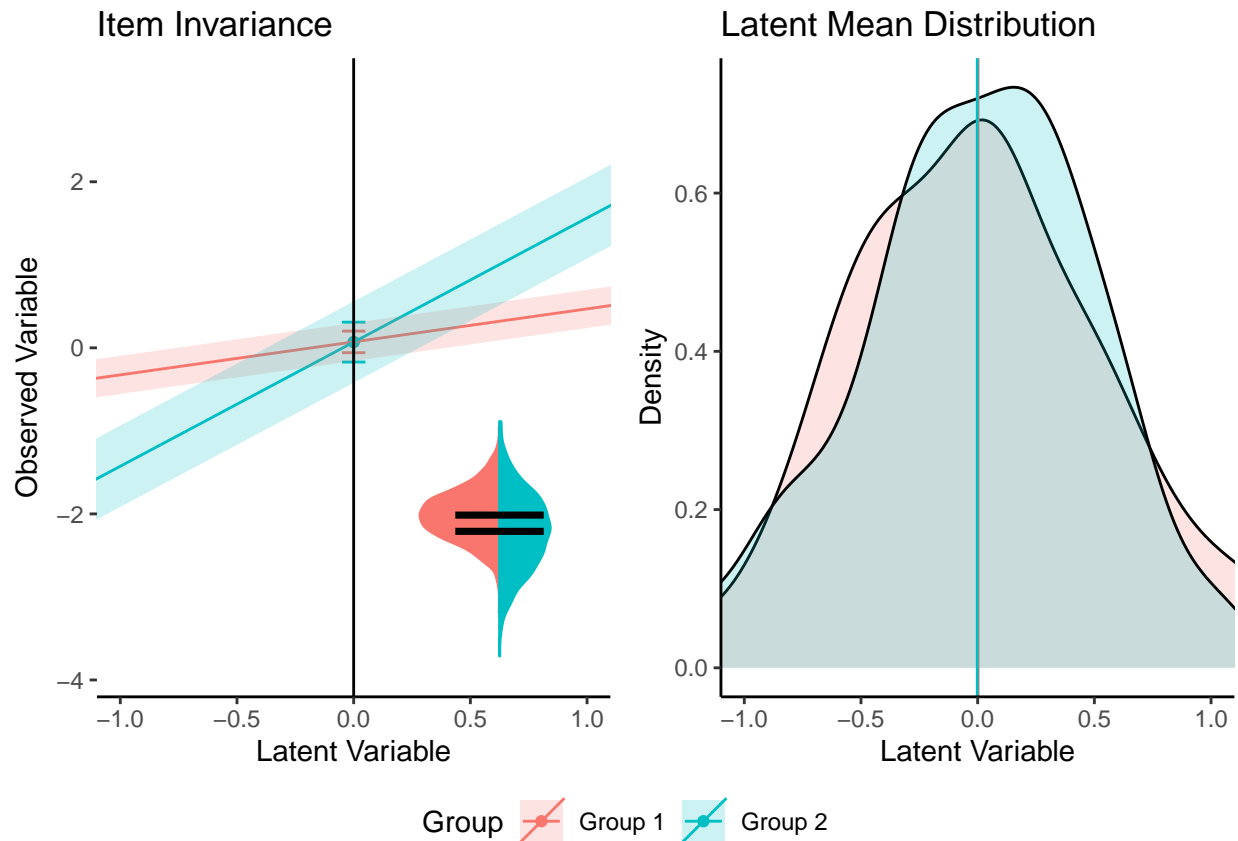
The latent means in Figure B3 do appear to show differences, albeit visually small. The latent means diagram shows the impact of any group differences that aren't constrained, and this image shows the configural model (as the metric model would force them to be equal). In the simulated model, the *only* manipulated parameter is question 4's loading. In real models, the differences may be larger due to other variation found in the parameter estimates. Therefore, once you discover items you believe would make a model "partially" invariant, you may wish to estimate that model and graph the item again using the partially invariant model to see only the effect of the non-invariant items. Additionally, consider that we set the scaling of the model to 0. The estimate for the lv mean in the large loading model was group 1: 0.00, and group 2: -0.06, which results in 0.06 difference in group means. The practical implications of this difference will depend on the research and interpretations of the researcher.

For intercepts, the small (Figure B4), medium (Figure B5), and large (Figure B6) depictions represent d_{MACS} values of 0.26, 0.47, and 0.70, respectively. Intercept differences can be clearly seen represented by the spacing out of the intercept locations (and thus, the overall line as well). While the changes in intercept do not appear to change the latent means, the caveat to this simulation is that only item four was manipulated. An example is provided below that demonstrates large changes in latent means.

**Figure B1**

Small Loadings Model Visualization

1107 Last, the effect of the residuals is plotted in small (Figure B7), medium (Figure B8),
 1108 and large (Figure B9) formats. While d_{MACS} values are not technically available for the
 1109 residuals, our models showed 0.19, 0.10, and 0.16, respectively. These differences in values
 1110 are variable due to the random generation of data sets for each measurement invariance
 1111 manipulation. At first glance, the differences in the small chart may seem large, because the
 1112 black lines are not touching, but notice that the distributions overlap, indicating a likely
 1113 small difference. The medium and large differences better illustrate differences in residuals
 1114 across groups. Further, the impact of the residuals on the shape of the latent mean

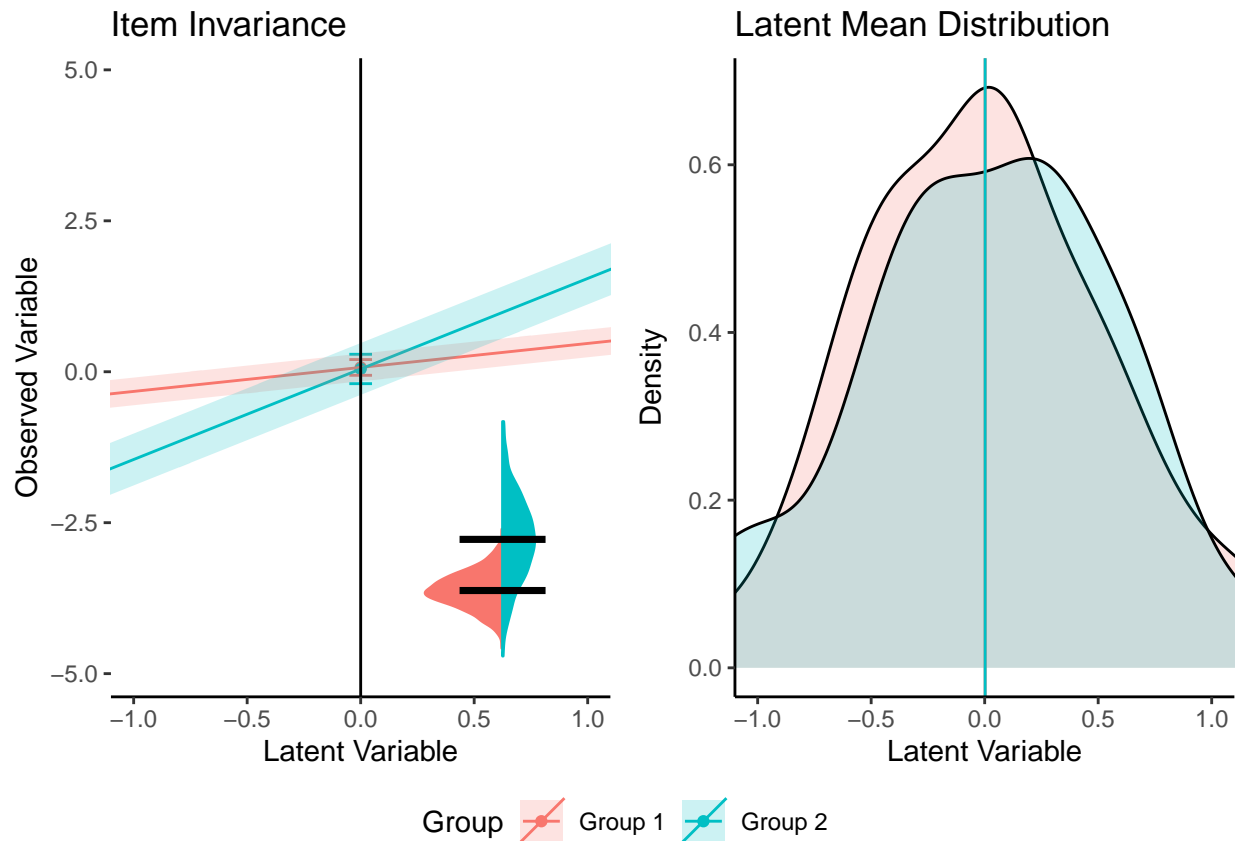
**Figure B2**

Medium Loadings Model Visualization

1115 distribution can also be seen (and unintentionally, in the first figures as well due to
 1116 random variation). The impact is due to the standard error of the residuals, as smaller
 1117 standard errors represent leptokurtic distributions (taller), and larger standard errors
 1118 represent platykurtic distributions (flatter). The effect size difference of the residuals does
 1119 not appear to change the effects in the latent means.

1120 Model Replication and Effect Size

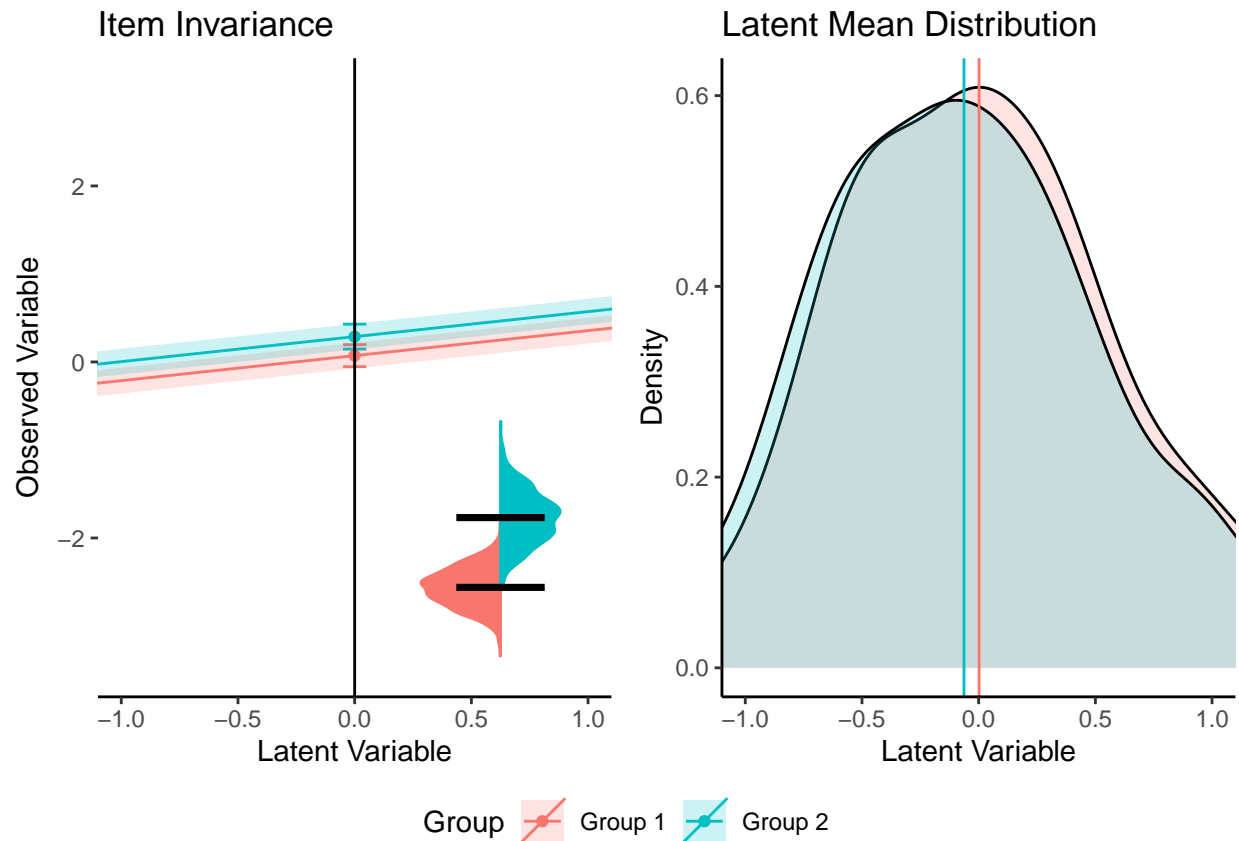
1121 Figure B10 portrays the h_{nmi_p} values by simulated non-invariance, strength of
 1122 non-invariance, and type of equality constraint. This image represents 100 simulations of
 1123 data by 1000 bootstrapped runs (averaged) to explore the expected pattern of results. The

**Figure B3**

Large Loadings Model Visualization

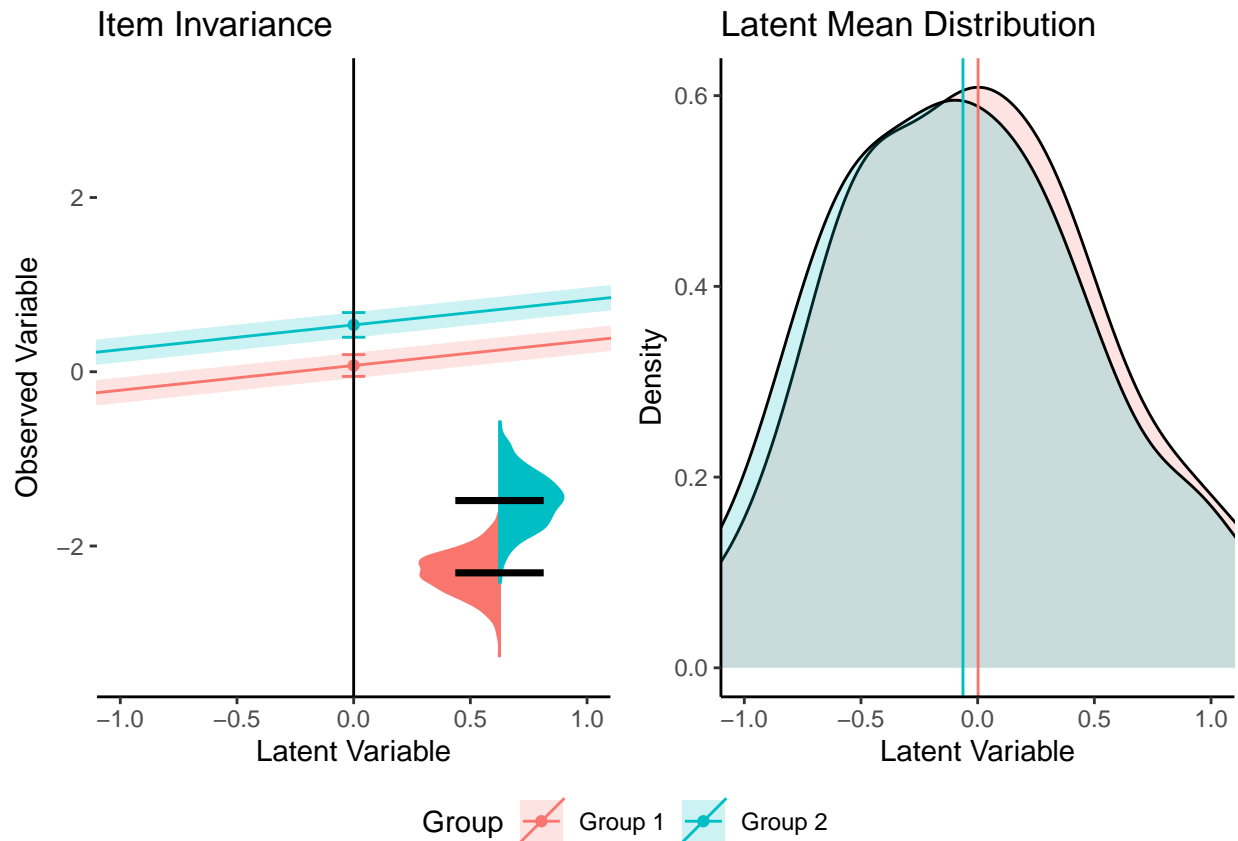
bars are arranged to show what a researcher might inspect when thinking about replication possibilities and their effect sizes (i.e., only three bars for each equality constraint would be calculated).

In the data that was simulated to be invariant between groups, effect sizes are still non-zero (loadings $h_{nmi_p} = 0.28$, intercepts $h_{nmi_p} = 0.06$, $h_{nmi_p} = 0.00$). This result mirrors the effects found in the literature - that often, many models fail to show invariance, and potentially not because measurement is poor but because of natural random variation in parameter estimates. This result also indicates the need to be able to identify if specific parameters are driving the differences, which is shown in the next section.

**Figure B4**

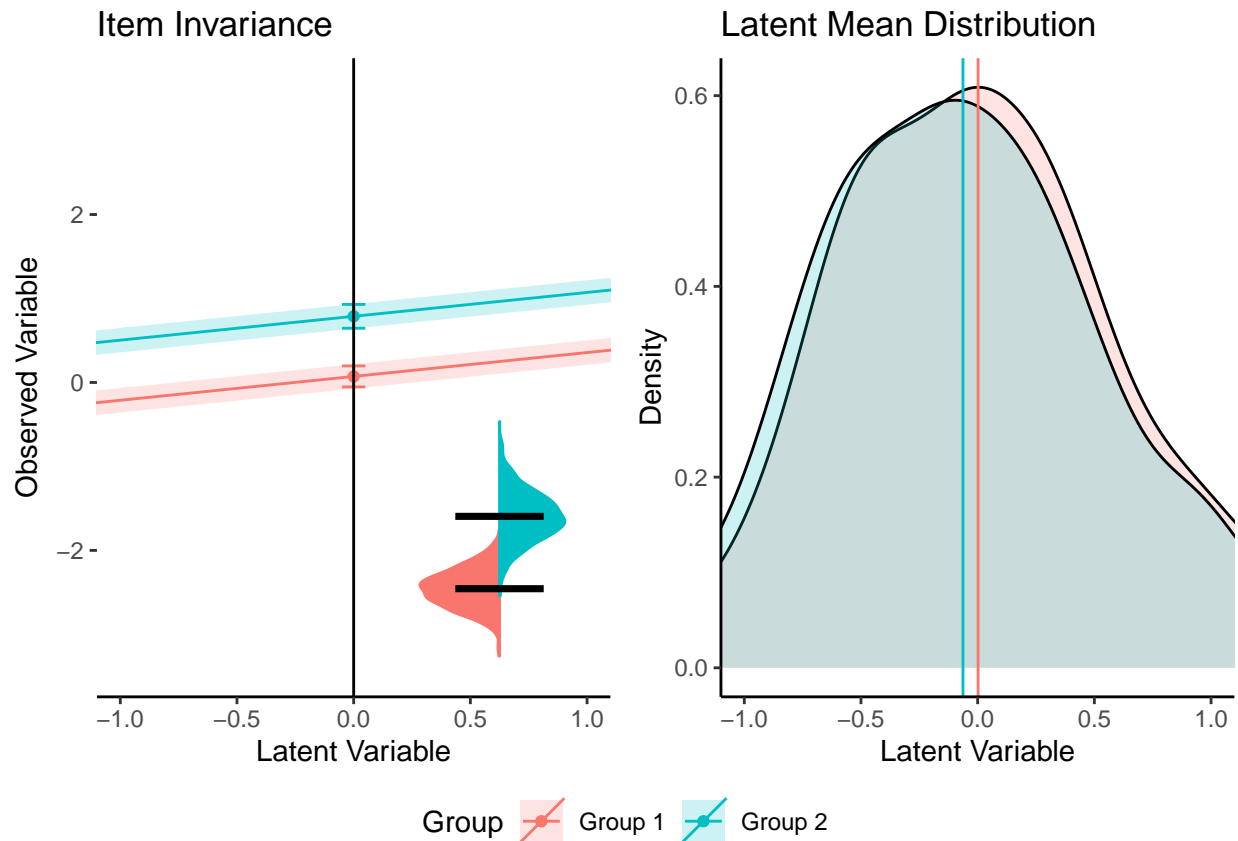
Small Intercepts Model Visualization

Next, Figure B10 demonstrates the patterns one might find for small, medium, and large effects at each type of invariance when data is simulated with *one* difference. For loadings, the pattern shows a larger effect for loadings with zero or negative effect sizes for other effect sizes. The intercept simulations show non-zero effect sizes in the loadings and intercepts, likely for the same reasons d_{MACS} is interpreted as a combined effect size. When intercepts are changed, loadings may naturally shift with those means. Last, the residual results present an unexpected pattern, wherein the effect is primarily seen in the loadings, rather than the residuals step. However, when distributions of error variance are different, one may expect that those effects are pushed toward the loadings as well (as values can vary more, thus potentially weakening the relationship between observed and latent variable).

**Figure B5**

Medium Intercepts Model Visualization

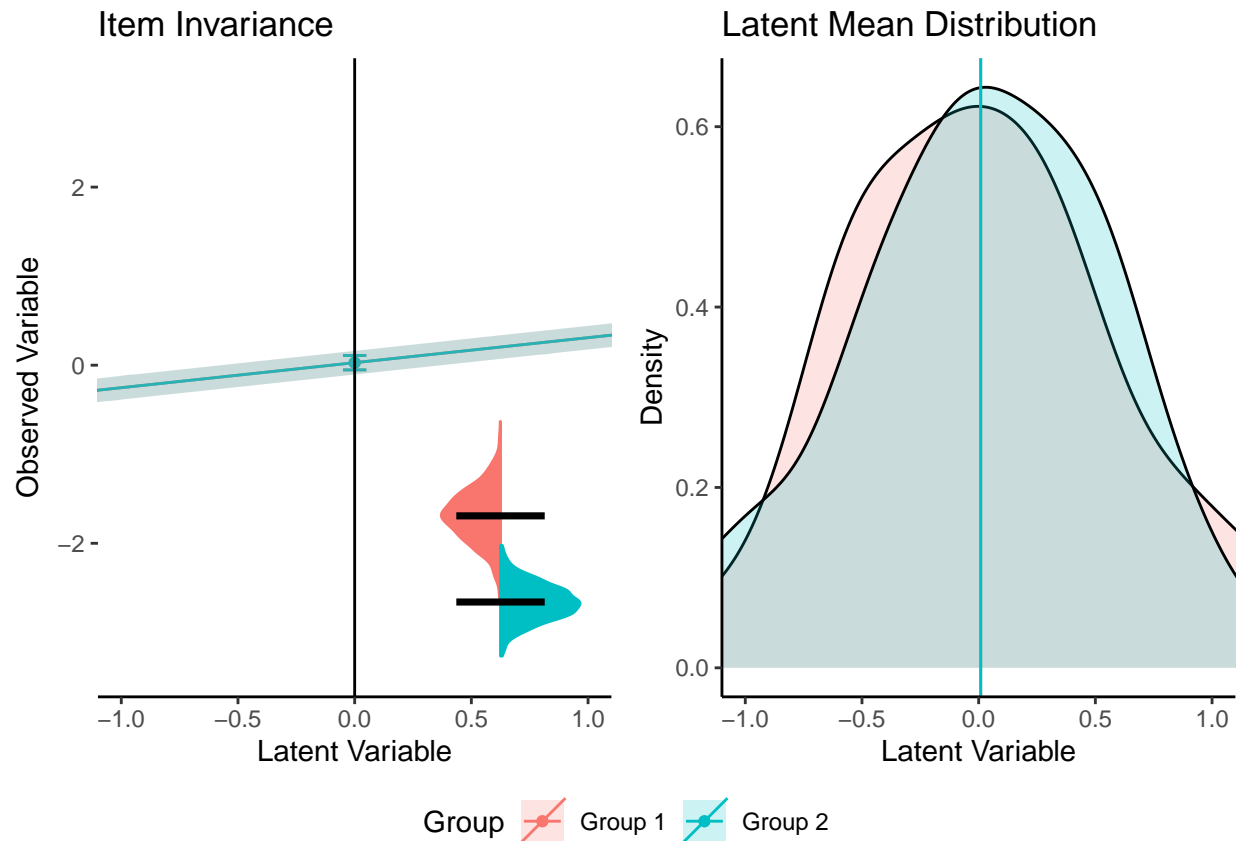
An example of interpretation on real data is given in a later section. From a research study, only one effect size for each equality constraint would be calculated. The interpretation will often be up to the researcher's smallest effect of interest, and this simulation gives some guidance that the values should not be interpreted with traditional rules of thumb. The pattern of effects is potentially the most useful information: 1) positive effects on the loadings with negative or very close to zero effects on the other parameters may indicate a non-replication in loadings, 2) equal effects on loadings and intercepts with smaller or negative effects may indicate intercepts may be an issue, and 3) residuals may be determined by the same pattern as loadings but with a smaller ratio of loadings to residuals effect (i.e., loadings h_{nmi} / residuals h_{nmi}). The "size" could be determined by the ratio of effect sizes for each constraint. Of course, this represents one simulation study, and results

**Figure B6***Large Intercepts Model Visualization*

1154 from many studies in a meta-analysis would be fruitful for future work.

1155 **Parameter Replication and Effect Size**

1156 Figure B11 shows the effect size differences within large loadings simulations. The
 1157 results demonstrate that most of the loadings were considered non-invariant in the
 1158 bootstrapped models (while holdings all others equal). This result is partially due to
 1159 simulating very good data, so small changes in loadings results in a drop in fit for our chosen
 1160 invariance index. However, we can use this graph to show that question four shows a
 1161 possible effect size ranging from -0.07 to 0.13. The h_{nmip} value for question four was 0.27,
 1162 representing about a quarter of a possible total effect. Last, the density plot in Figure B12
 1163 shows the separation of the two different groups loadings in item four, thus, illustrating

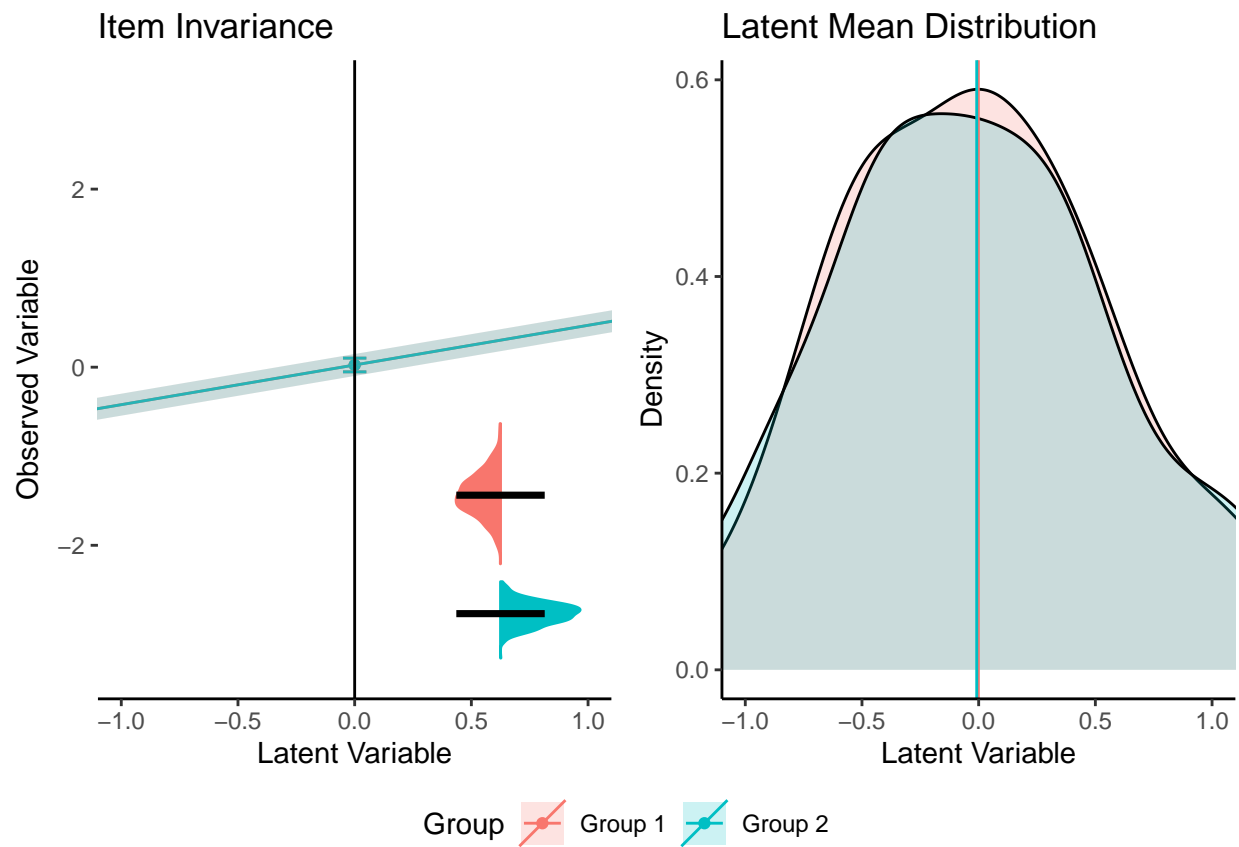
**Figure B7**

Small Residuals Model Visualization

group differences in the findings for their loadings. Each of the other combination of plots can be found in the supplemental materials.

MGCFA Model Fit Statistics

Model fit statistics are provided for each of the ten model combinations (invariant, three sizes for each loadings, intercepts, and residuals). These tables could be used to examine the traditional change in fit statistics cutoff rules of thumb (Cheung & Rensvold, 2002), such as Δ CFI or Δ RMSEA, to the visualizations presented in the manuscript.

**Figure B8**

Medium Residuals Model Visualization

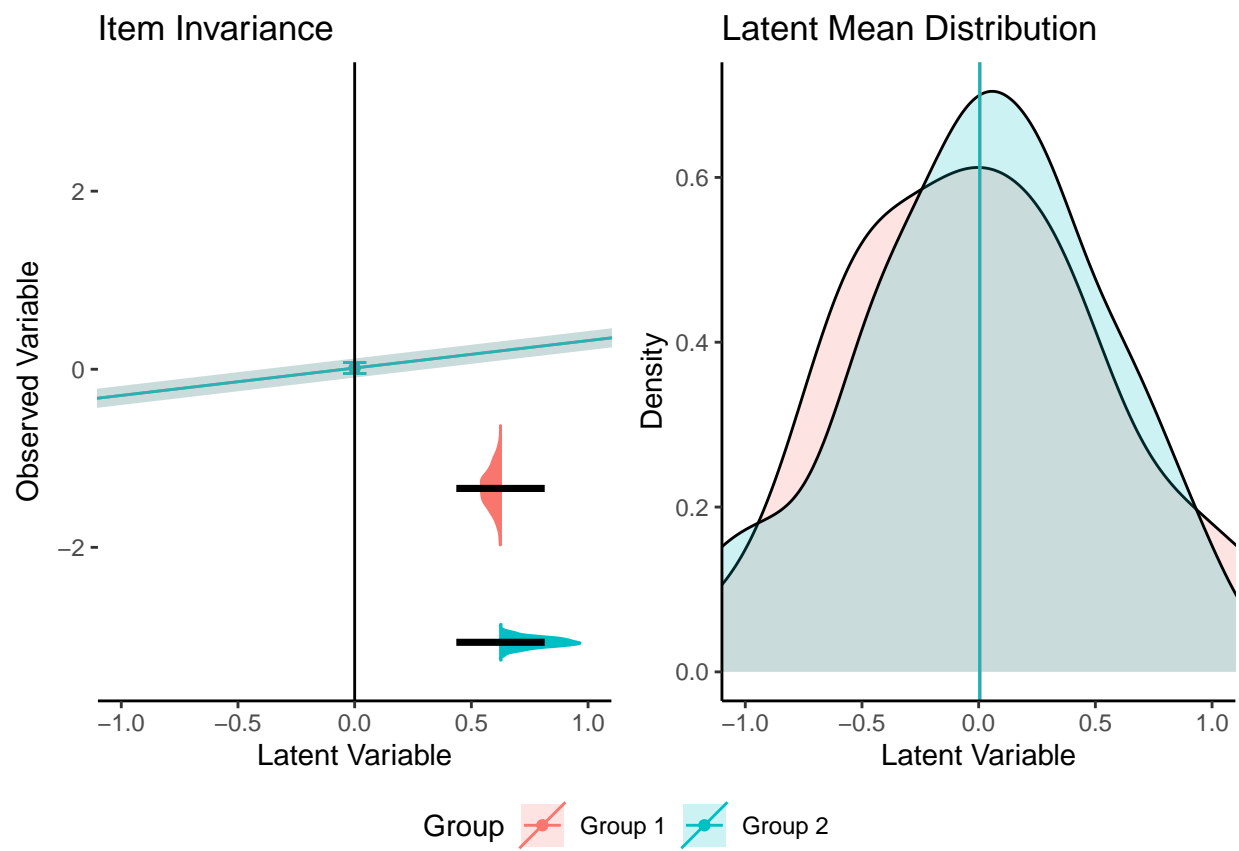
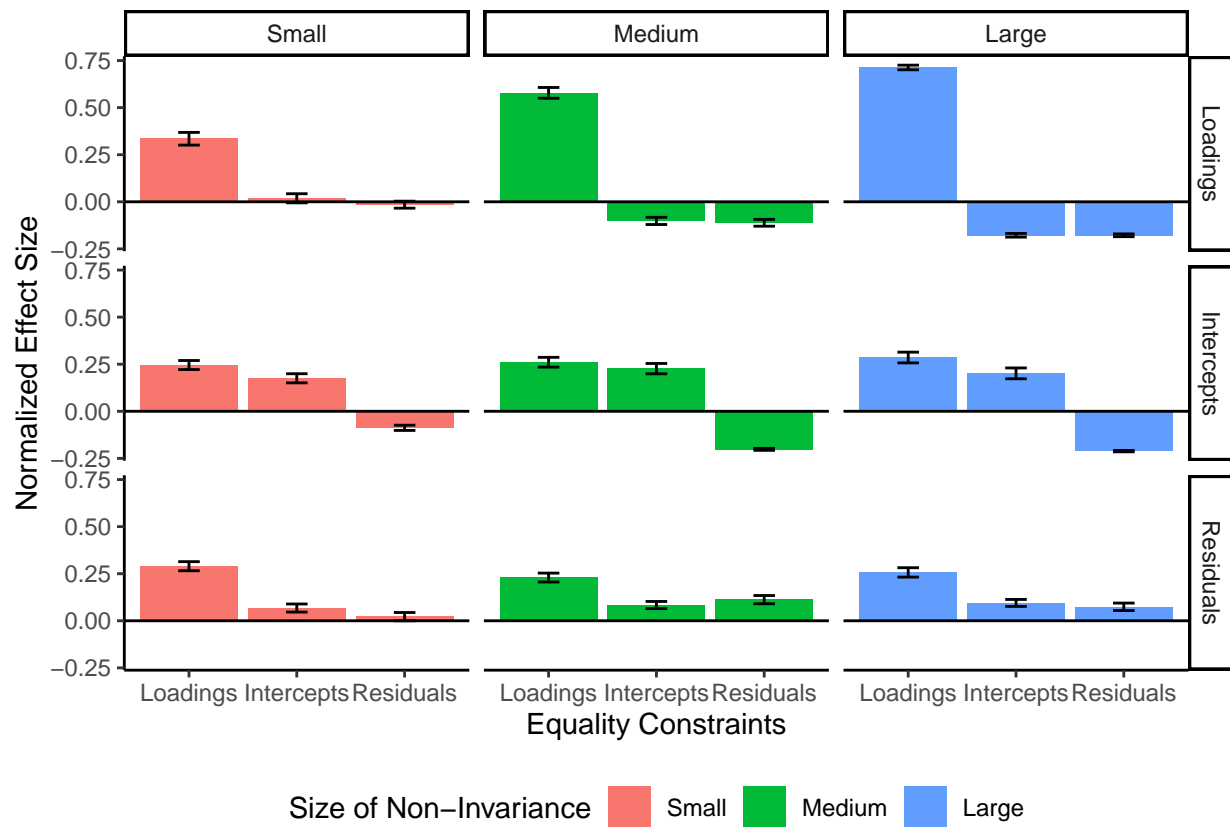
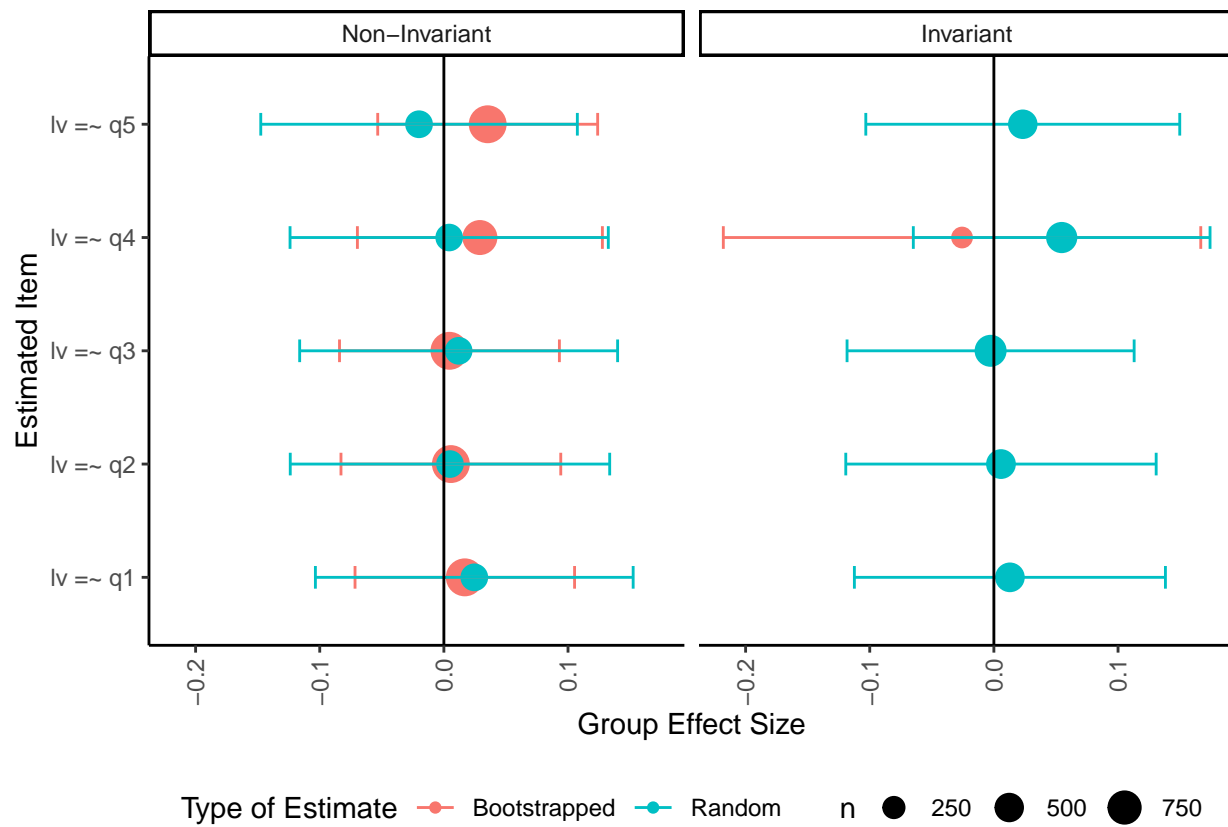


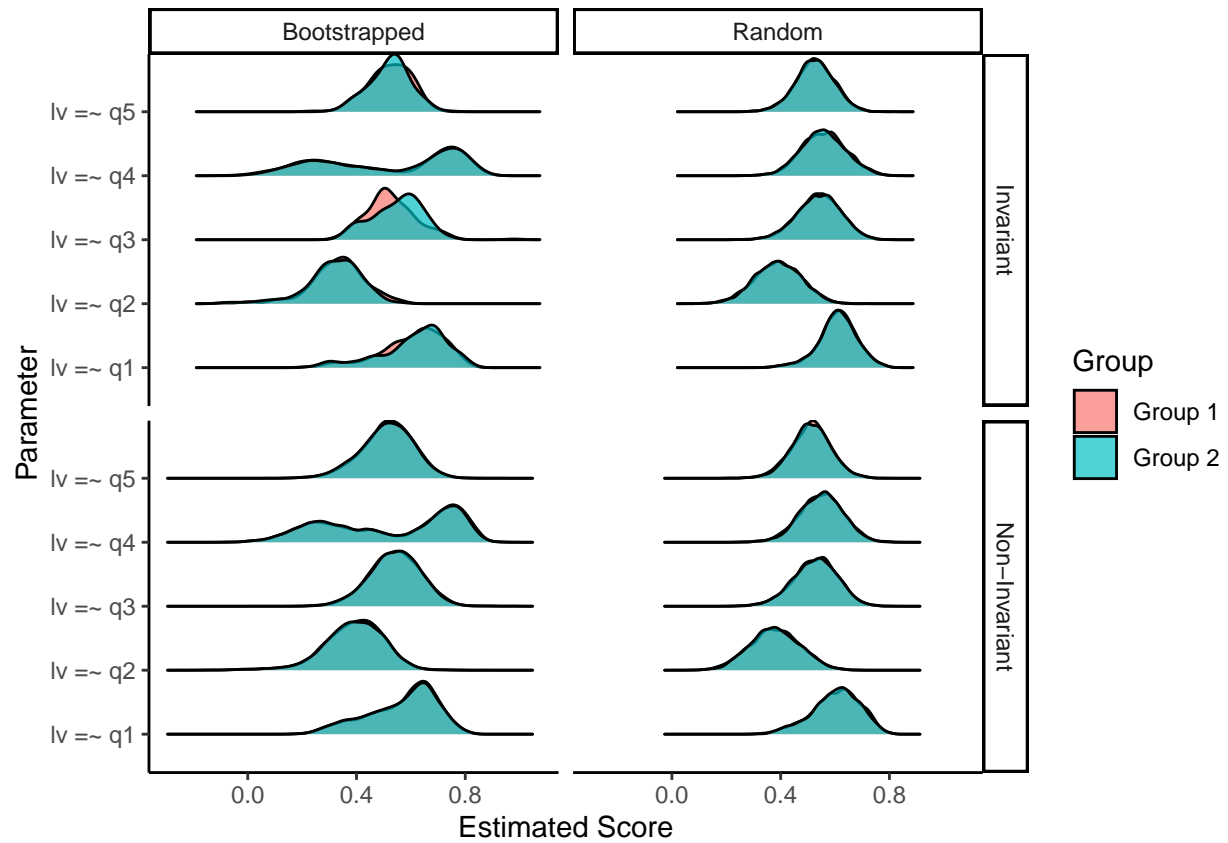
Figure B9
Large Residuals Model Visualization

**Figure B10**

Visualization of the effect size of bootstrapped replication proportions on simulated data. Each panel indicates the simulated data type, colors represent the differences in the strength of the non-invariance, and the bars on the x-axis represent the effect size for the equality constraint.

**Figure B11**

Bootstrapped and Random Group effect size differences in loadings for the Large Loading difference simulation. The size of the point represents the number of data points included in that calculation.

**Figure B12**

Bootstrapped and Random density plots for invariant and non-invariant bootstrapped partial effects examining only large loadings.

[tbp]

Table B1*Model Fit for Invariant Model*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,516.454	7,579.673	1.000	1.036	0.000	0.006
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,767.599	3,820.421	1.000	1.008	0.000	0.021
Configural	7,533.348	7,659.786	0.991	0.982	0.030	0.026
Loadings	7,528.476	7,638.056	0.994	0.992	0.020	0.033
Intercepts	7,522.397	7,615.118	1.000	1.003	0.000	0.035
Residuals	7,520.435	7,592.083	0.991	0.992	0.020	0.046

[tbp]

Table B2*Model Fit for Small Differences in Loadings*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,530.321	7,593.540	0.977	0.955	0.049	0.025
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,785.242	3,838.064	0.979	0.958	0.050	0.029
Configural	7,550.991	7,677.430	0.978	0.956	0.048	0.030
Loadings	7,550.133	7,659.713	0.966	0.952	0.051	0.047
Intercepts	7,542.675	7,635.397	0.979	0.977	0.035	0.047
Residuals	7,534.091	7,605.739	0.993	0.994	0.019	0.054

[tbp]

Table B3*Model Fit for Medium Differences in Loadings*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,595.822	7,659.041	1.000	1.017	0.000	0.012
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,815.314	3,868.136	0.996	0.991	0.025	0.023
Configural	7,581.063	7,707.501	0.988	0.976	0.038	0.027
Loadings	7,609.348	7,718.928	0.878	0.826	0.101	0.079
Intercepts	7,601.550	7,694.271	0.891	0.879	0.084	0.079
Residuals	7,596.811	7,668.459	0.890	0.905	0.075	0.096

[tbp]

Table B4*Model Fit for Large Differences in Loadings*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,665.760	7,728.979	1.000	1.022	0.000	0.010
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,855.953	3,908.775	0.997	0.995	0.020	0.023
Configural	7,621.702	7,748.140	0.989	0.978	0.036	0.027
Loadings	7,659.690	7,769.270	0.852	0.789	0.114	0.088
Intercepts	7,652.360	7,745.081	0.863	0.848	0.097	0.088
Residuals	7,664.853	7,736.502	0.806	0.832	0.102	0.135

[tbp]

Table B5*Model Fit for Small Differences in Intercepts*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,520.471	7,583.690	1.000	1.035	0.000	0.007
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,767.599	3,820.421	1.000	1.008	0.000	0.021
Configural	7,533.348	7,659.786	0.991	0.982	0.030	0.026
Loadings	7,528.476	7,638.056	0.994	0.992	0.020	0.033
Intercepts	7,526.312	7,619.034	0.987	0.986	0.027	0.040
Residuals	7,524.356	7,596.005	0.975	0.978	0.033	0.050

[tbp]

Table B6*Model Fit for Medium Differences in Intercepts*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,538.375	7,601.594	1.000	1.033	0.000	0.008
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,767.599	3,820.421	1.000	1.008	0.000	0.021
Configural	7,533.348	7,659.786	0.991	0.982	0.030	0.026
Loadings	7,528.476	7,638.056	0.994	0.992	0.020	0.033
Intercepts	7,544.002	7,636.724	0.917	0.907	0.068	0.059
Residuals	7,542.064	7,613.712	0.905	0.917	0.065	0.067

[tbp]

Table B7*Model Fit for Large Differences in Intercepts*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,568.748	7,631.967	1.000	1.032	0.000	0.008
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,767.599	3,820.421	1.000	1.008	0.000	0.021
Configural	7,533.348	7,659.786	0.991	0.982	0.030	0.026
Loadings	7,528.476	7,638.056	0.994	0.992	0.020	0.033
Intercepts	7,574.054	7,666.776	0.797	0.775	0.106	0.084
Residuals	7,572.174	7,643.823	0.785	0.813	0.097	0.090

[tbp]

Table B8*Model Fit for Small Differences in Residuals*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,462.007	7,525.226	1.000	1.020	0.000	0.013
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,703.797	3,756.619	0.962	0.924	0.061	0.037
Configural	7,469.546	7,595.984	0.969	0.938	0.054	0.034
Loadings	7,471.637	7,581.217	0.944	0.920	0.062	0.049
Intercepts	7,465.722	7,558.443	0.952	0.946	0.051	0.051
Residuals	7,465.986	7,537.635	0.930	0.939	0.054	0.065

[tbp]

Table B9*Model Fit for Medium Differences in Residuals*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,382.876	7,446.095	0.992	0.985	0.028	0.020
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,601.920	3,654.742	1.000	1.023	0.000	0.018
Configural	7,367.669	7,494.108	0.996	0.992	0.020	0.025
Loadings	7,371.027	7,480.607	0.969	0.956	0.049	0.046
Intercepts	7,364.724	7,457.446	0.977	0.975	0.037	0.047
Residuals	7,386.854	7,458.502	0.877	0.893	0.076	0.076

[tbp]

Table B10*Model Fit for Large Differences in Residuals*

Model	AIC	BIC	CFI	TLI	RMSEA	SRMR
Overall	7,301.897	7,365.116	0.993	0.986	0.026	0.019
Group Group 1	3,765.749	3,818.571	0.976	0.953	0.047	0.031
Group Group 2	3,454.309	3,507.131	0.945	0.889	0.075	0.036
Configural	7,220.058	7,346.496	0.960	0.920	0.063	0.034
Loadings	7,218.878	7,328.458	0.948	0.926	0.060	0.046
Intercepts	7,213.011	7,305.732	0.956	0.951	0.049	0.048
Residuals	7,305.660	7,377.309	0.559	0.617	0.137	0.189

Appendix C

Simulated Partial Invariance Results

[tbp]

Table C1

Fit Estimates for Partial Invariance

Residuals on Invariant Data

Estimated Parameter	CFI	RSMEA
q1 \sim q1	0.990	0.021
q2 \sim q2	0.987	0.024
q3 \sim q3	0.996	0.014
q4 \sim q4	1.000	0.000
q5 \sim q5	0.987	0.025
lv \sim lv	0.991	0.020

[tbp]

Table C2

Fit Estimates for Partial Invariance

Loadings for Small Loading Data

Estimated Parameter	CFI	RSMEA
lv \sim q1	0.993	0.019
lv \sim q2	0.989	0.023
lv \sim q3	0.989	0.023
lv \sim q4	1.000	0.000
lv \sim q5	0.994	0.017

[tbp]

Table C3*Fit Estimates for Partial Invariance**Loadings for Medium Loading Data*

Estimated Parameter	CFI	RSMEA
lv =~ q1	0.890	0.075
lv =~ q2	0.904	0.072
lv =~ q3	0.887	0.078
lv =~ q4	1.000	0.000
lv =~ q5	0.914	0.068

[tbp]

Table C4*Fit Estimates for Partial Invariance**Loadings for Large Loading Data*

Estimated Parameter	CFI	RSMEA
lv =~ q1	0.806	0.102
lv =~ q2	0.812	0.102
lv =~ q3	0.813	0.102
lv =~ q4	1.000	0.000
lv =~ q5	0.861	0.088

[tbp]

Table C5*Fit Estimates for Partial Invariance**Loadings for Small Intercept Data*

Estimated Parameter	CFI	RSMEA
q1 ~1	0.975	0.033
lv ~1	0.975	0.033
q2 ~1	0.972	0.035
q3 ~1	0.972	0.036
q4 ~1	0.988	0.023
q5 ~1	0.971	0.036

[tbp]

Table C6*Fit Estimates for Partial Invariance**Loadings for Medium Intercept Data*

Estimated Parameter	CFI	RSMEA
q1 ~1	0.905	0.065
lv ~1	0.905	0.065
q2 ~1	0.901	0.067
q3 ~1	0.901	0.067
q4 ~1	0.988	0.023
q5 ~1	0.902	0.067

[tbp]

Table C7
Fit Estimates for Partial Invariance
Loadings for Large Intercept Data

Estimated Parameter	CFI	RSMEA
q1 ~1	0.785	0.097
lv ~1	0.785	0.097
q2 ~1	0.781	0.100
q3 ~1	0.781	0.100
q4 ~1	0.988	0.023
q5 ~1	0.784	0.099

[tbp]

Table C8
Fit Estimates for Partial Invariance
Loadings for Small Residual Data

Estimated Parameter	CFI	RSMEA
q1 ~~ q1	0.928	0.056
q2 ~~ q2	0.936	0.053
q3 ~~ q3	0.926	0.057
q4 ~~ q4	0.955	0.044
q5 ~~ q5	0.926	0.057
lv ~~ lv	0.930	0.054

[tbp]

Table C9*Fit Estimates for Partial Invariance**Loadings for Medium Residual Data*

Estimated Parameter	CFI	RSMEA
q1 \sim q1	0.879	0.077
q2 \sim q2	0.873	0.079
q3 \sim q3	0.878	0.077
q4 \sim q4	0.980	0.031
q5 \sim q5	0.873	0.079
lv \sim lv	0.877	0.076

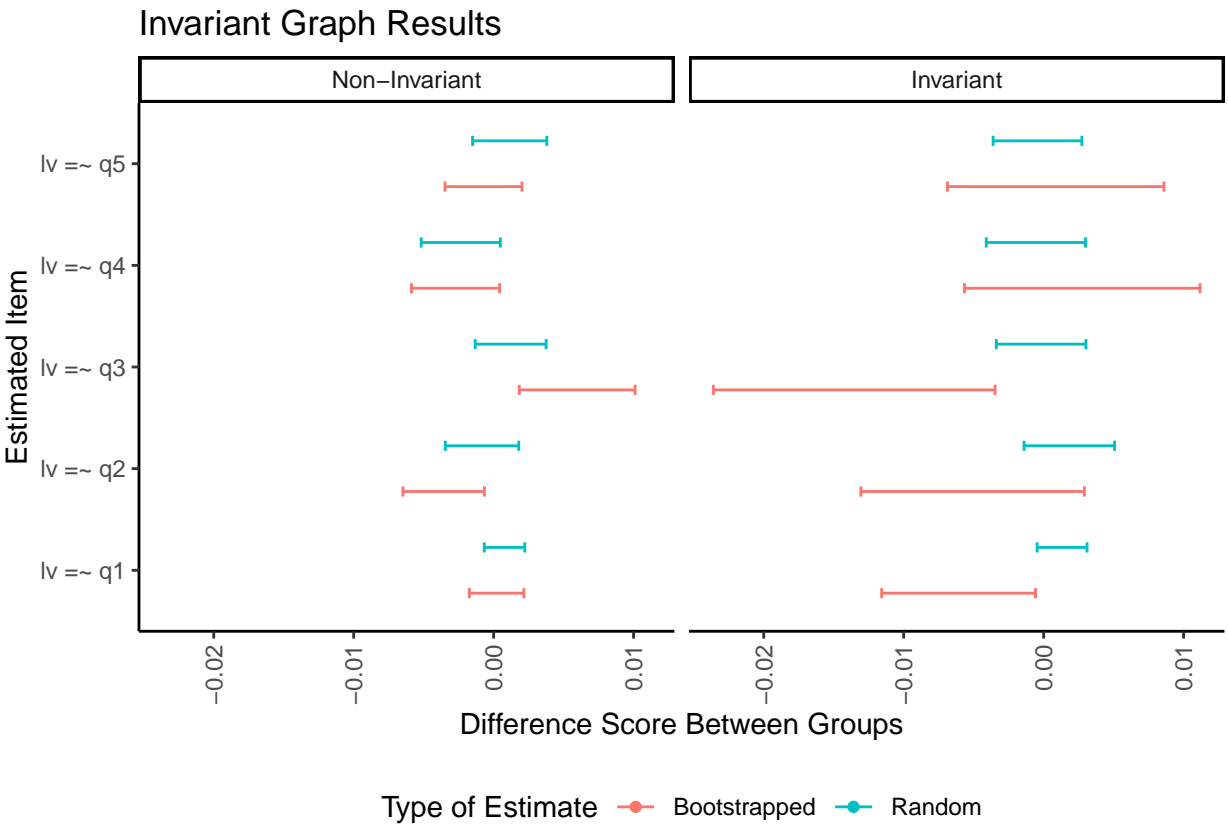
[tbp]

Table C10*Fit Estimates for Partial Invariance**Loadings for Large Residual Data*

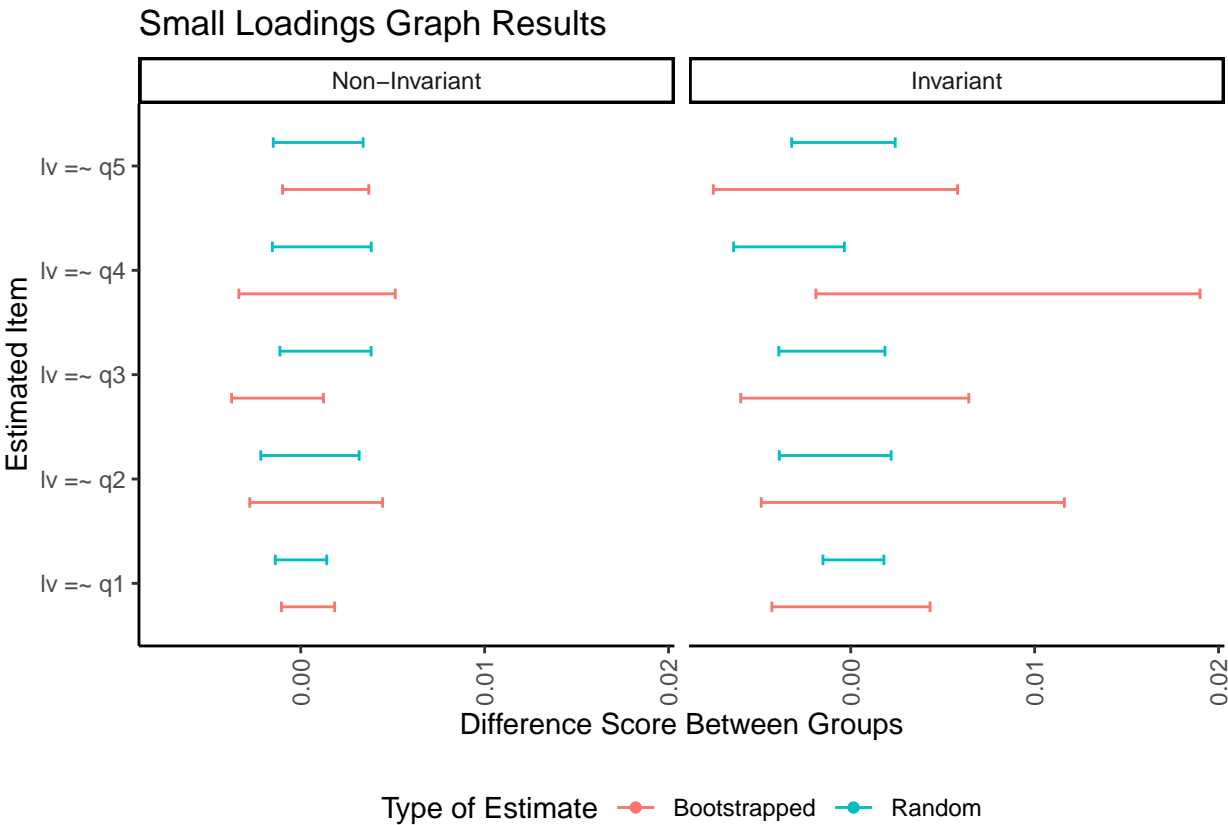
Estimated Parameter	CFI	RSMEA
q1 \sim q1	0.555	0.141
q2 \sim q2	0.556	0.141
q3 \sim q3	0.556	0.141
q4 \sim q4	0.967	0.039
q5 \sim q5	0.557	0.141
lv \sim lv	0.559	0.137

Appendix D

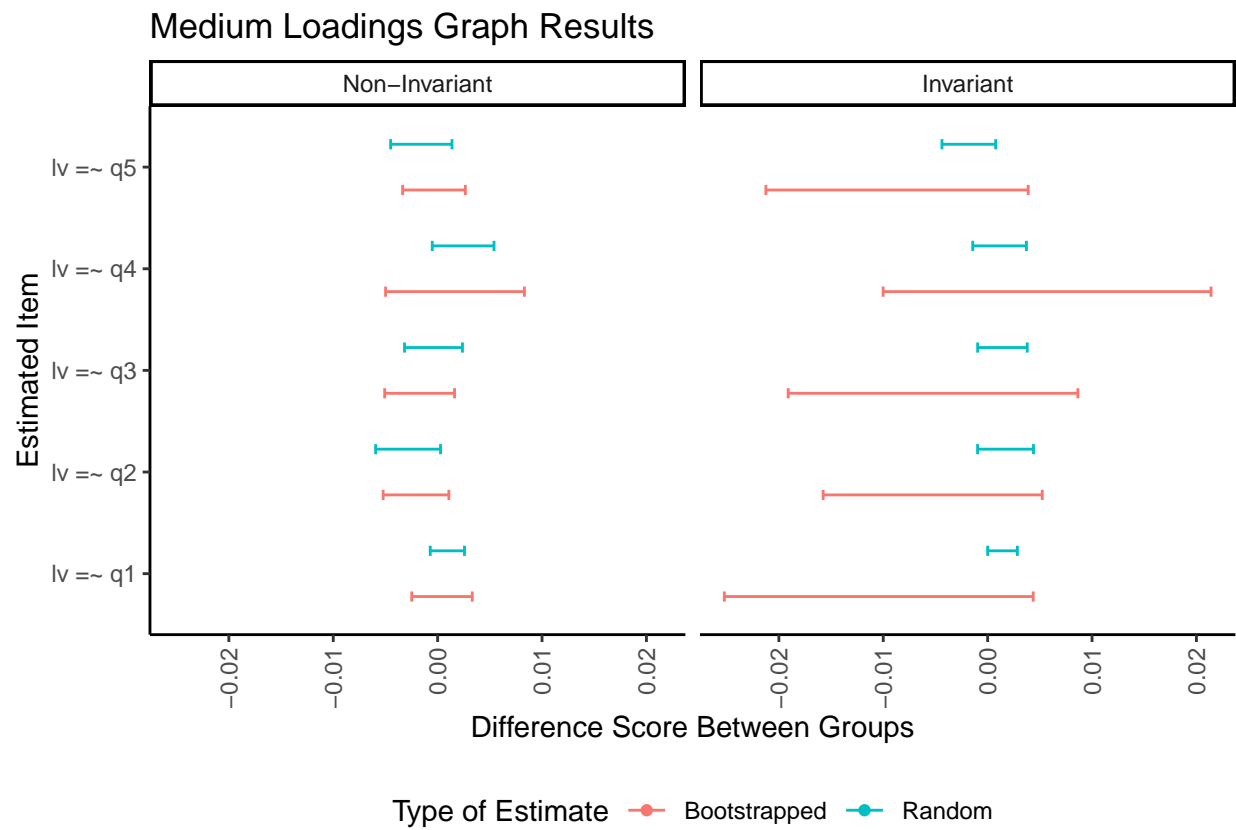
Invariance Plots Difference Scores by Condition

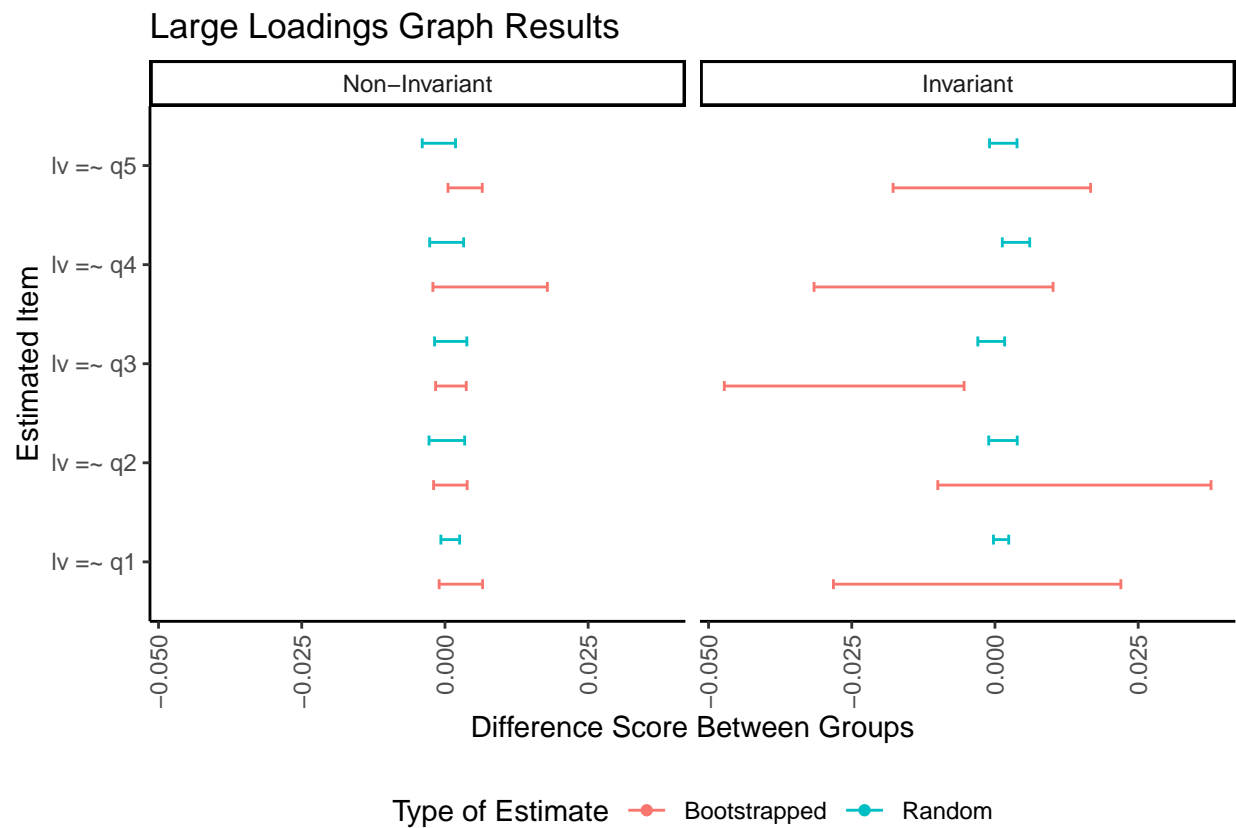


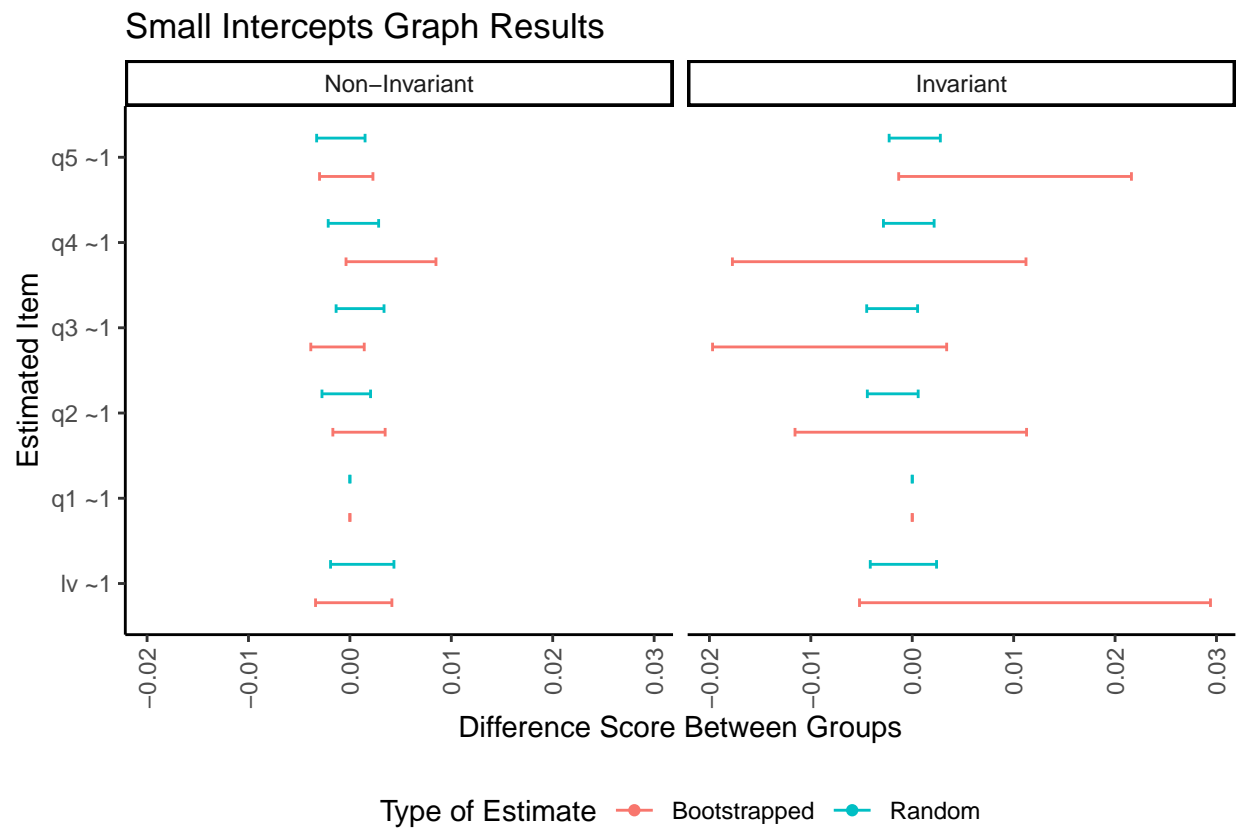
1171

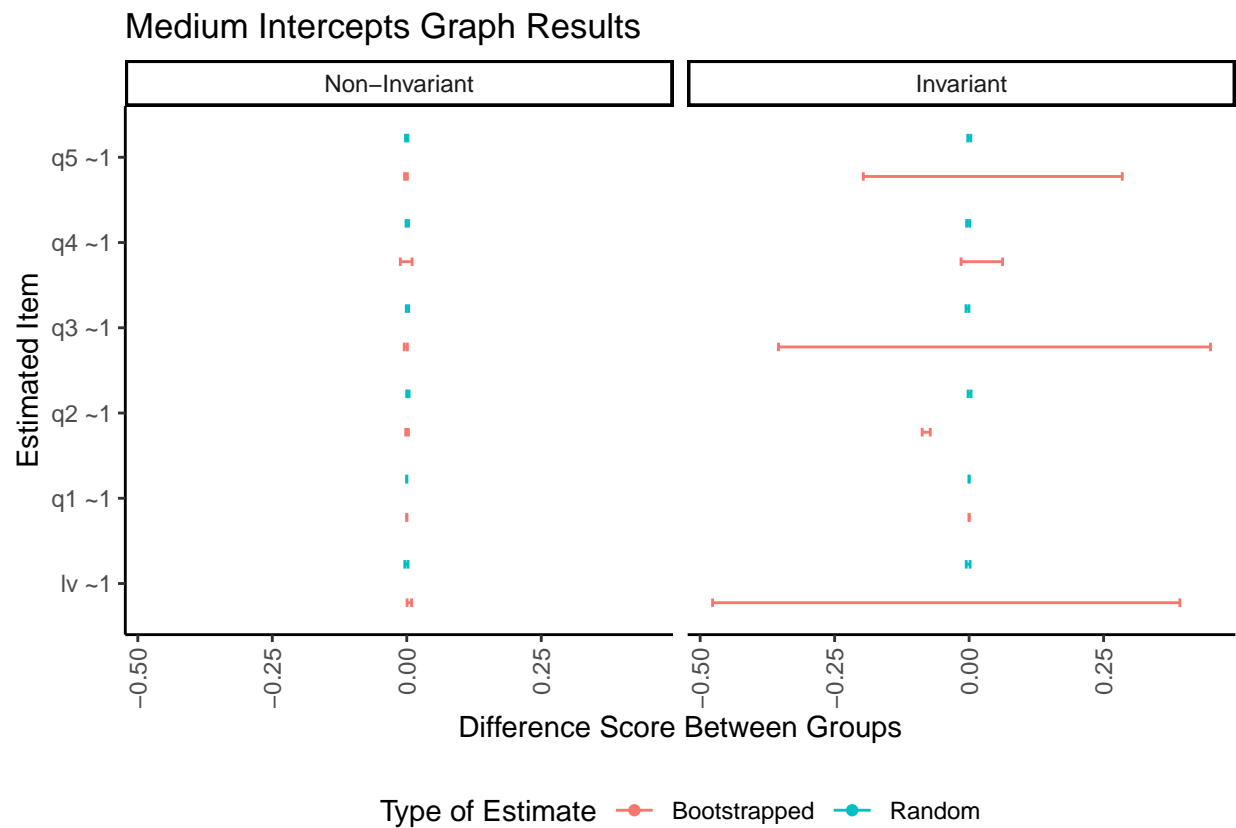


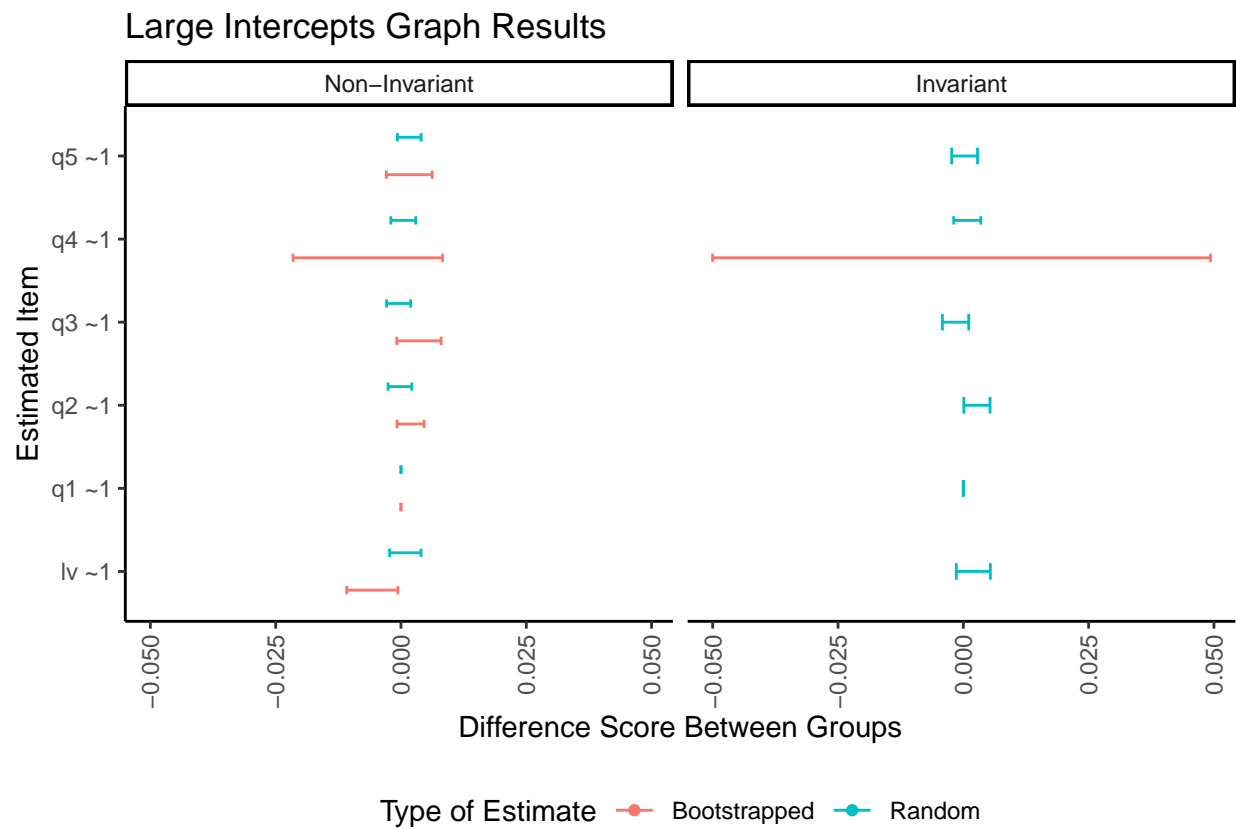
1172

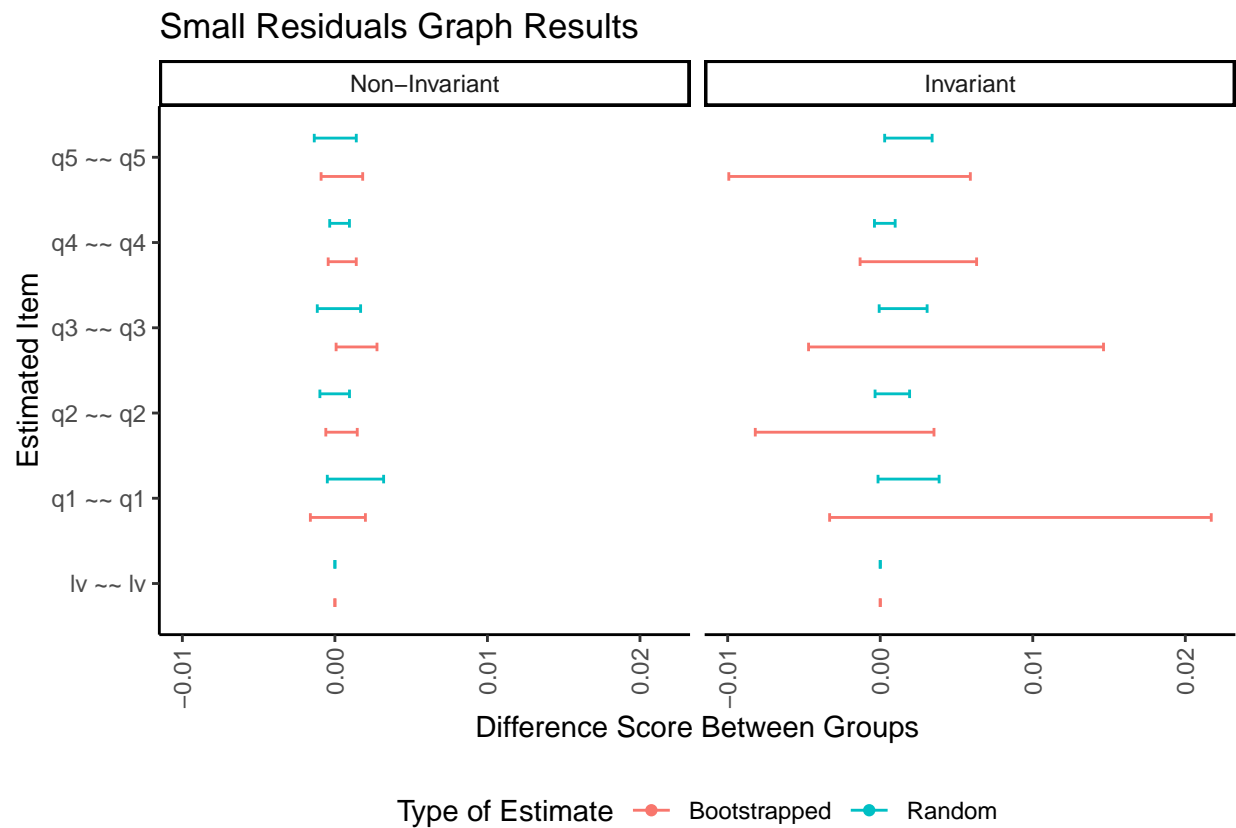


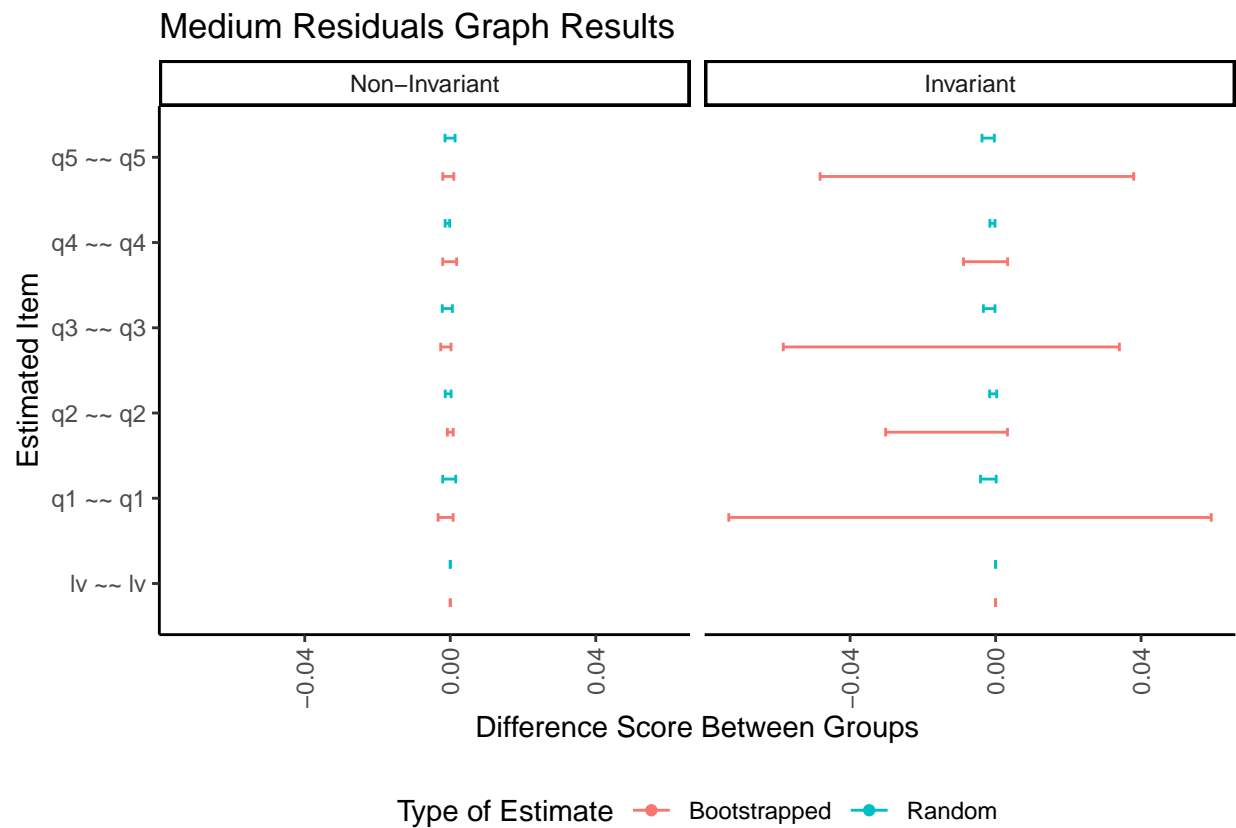


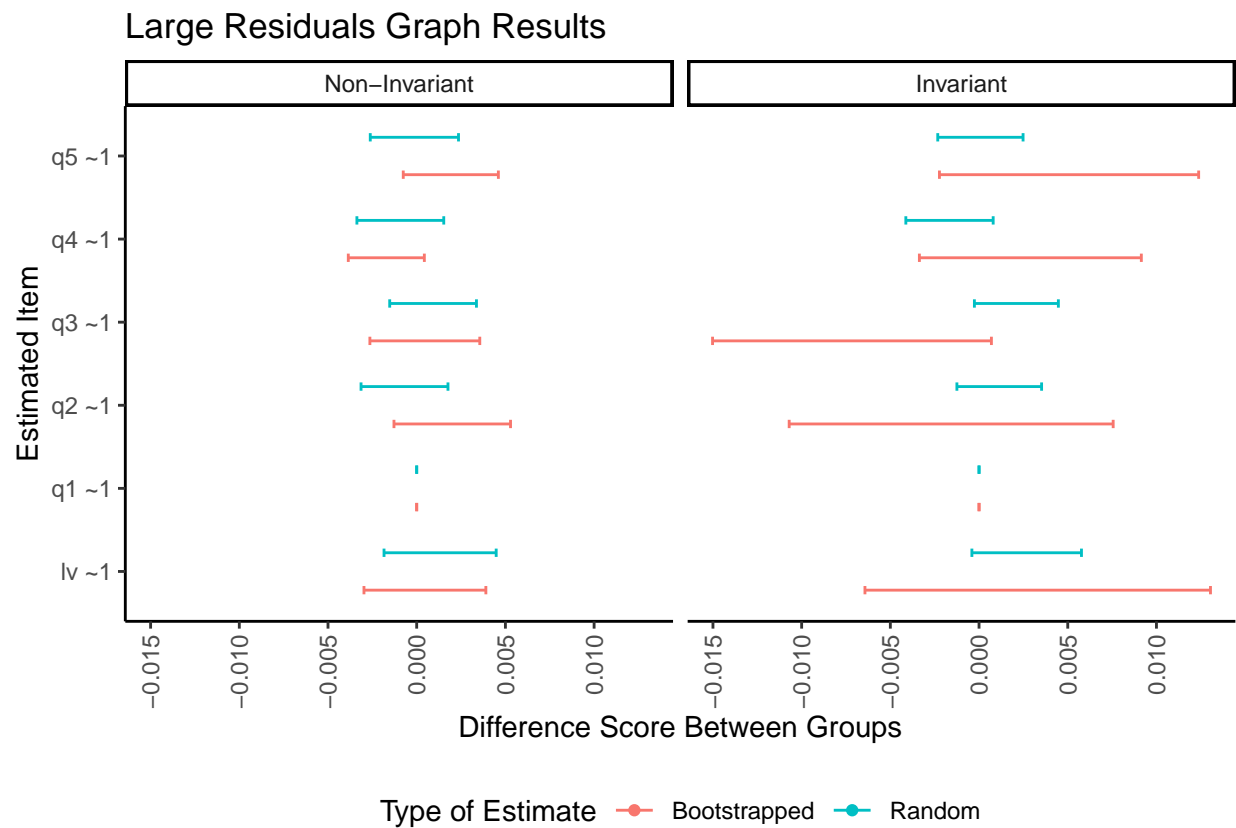






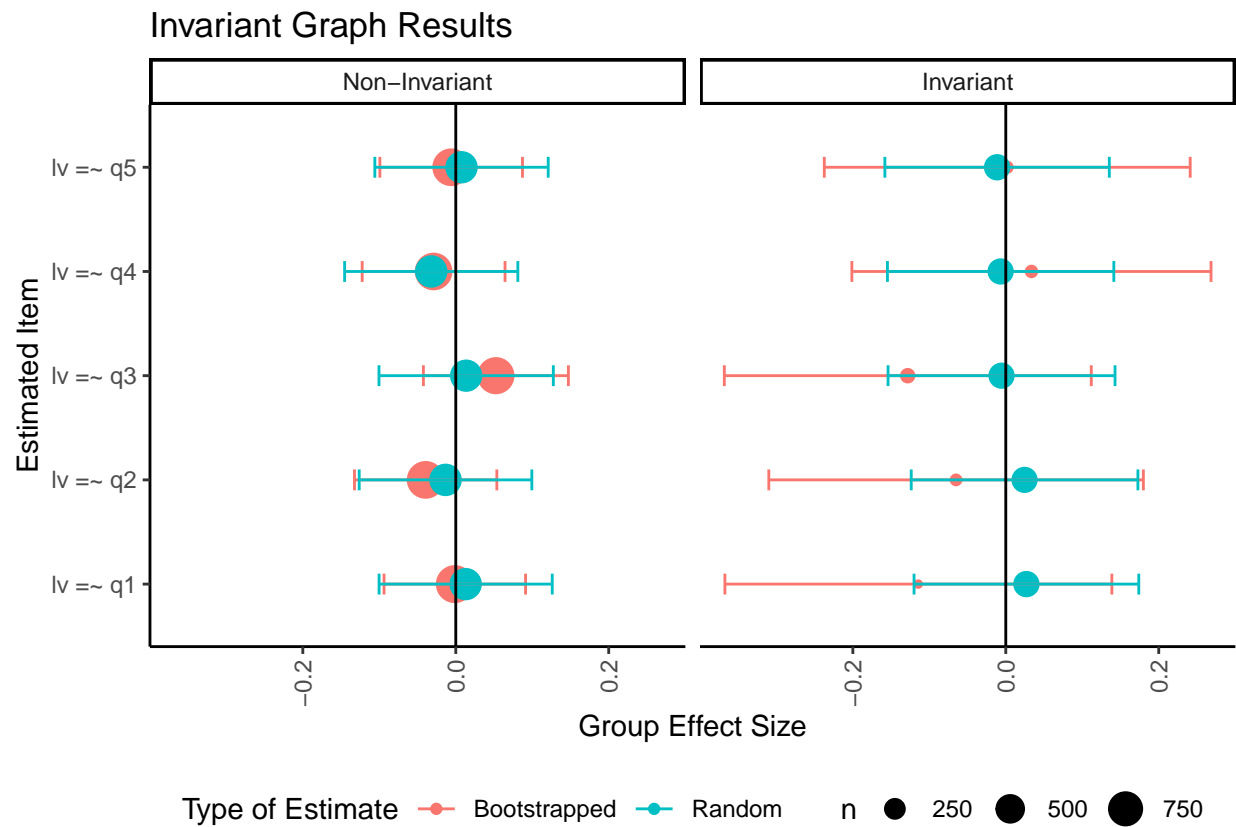






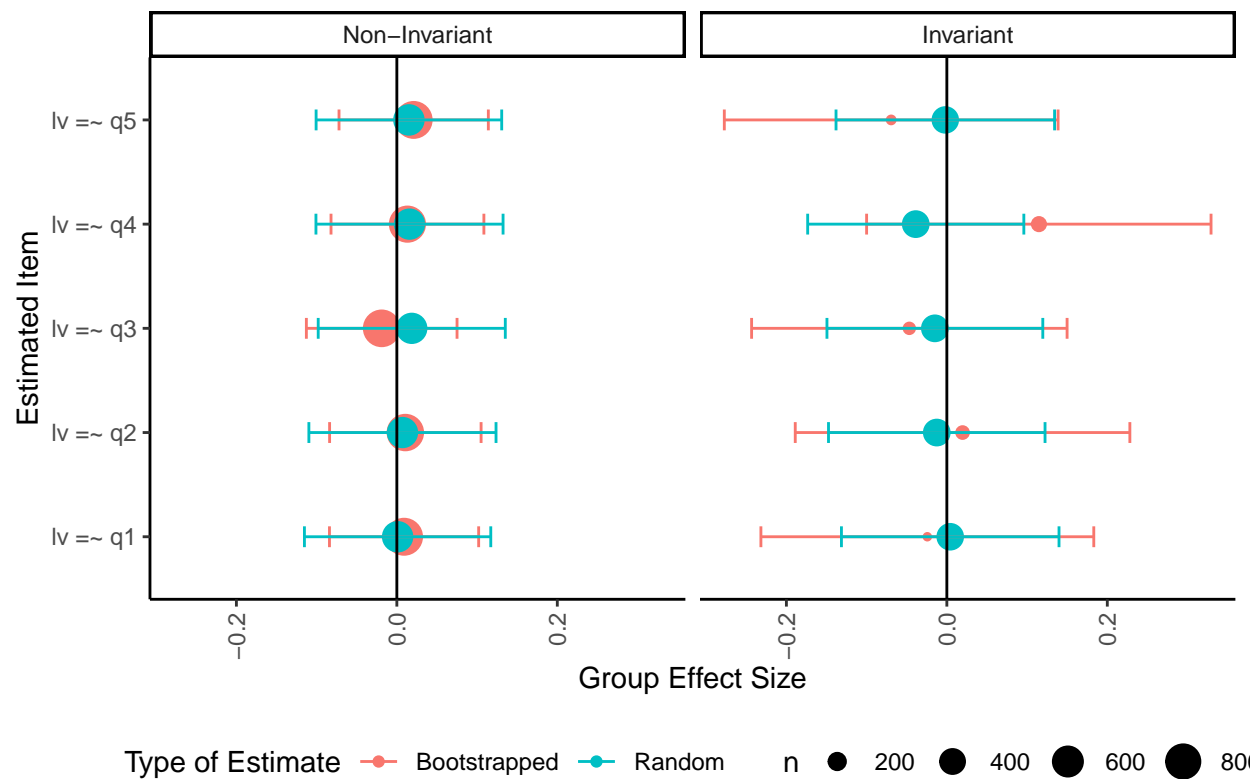
Appendix E

Invariance Plots Effect Sizes by Condition

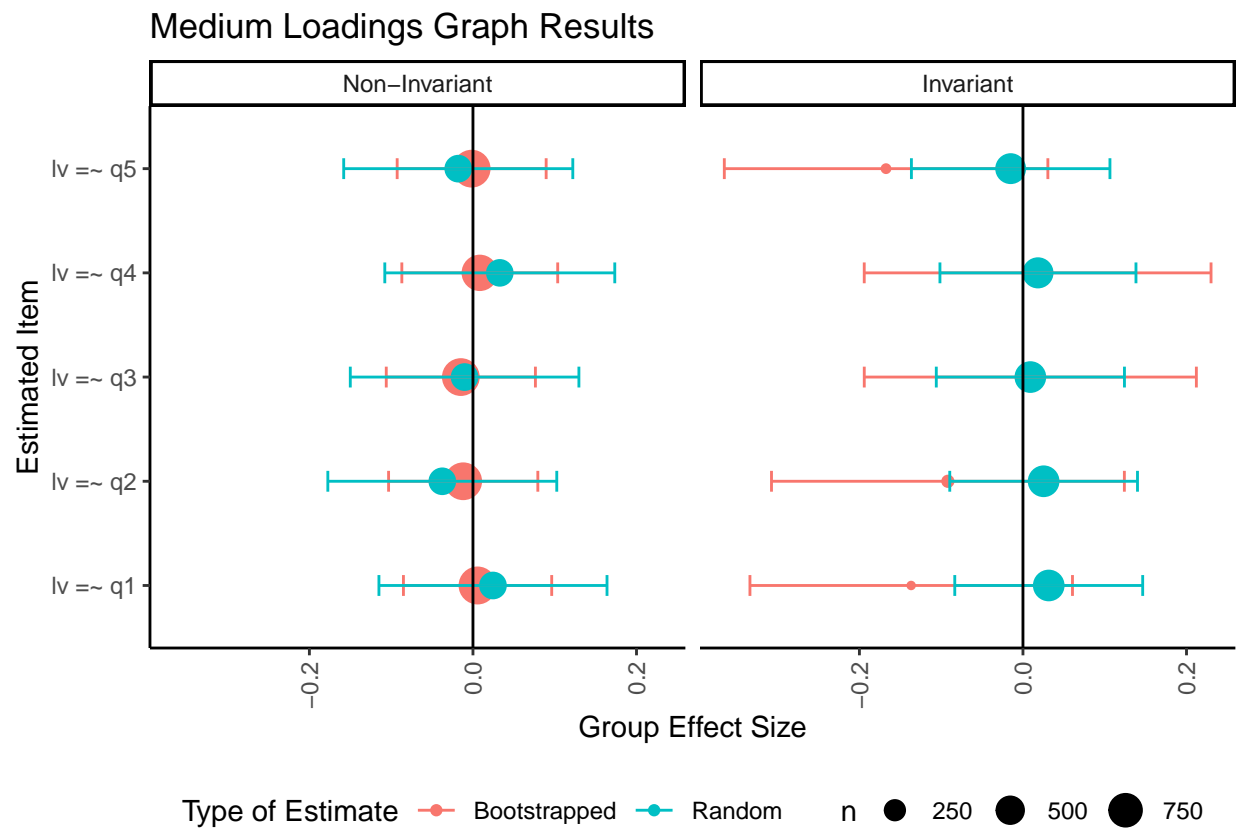


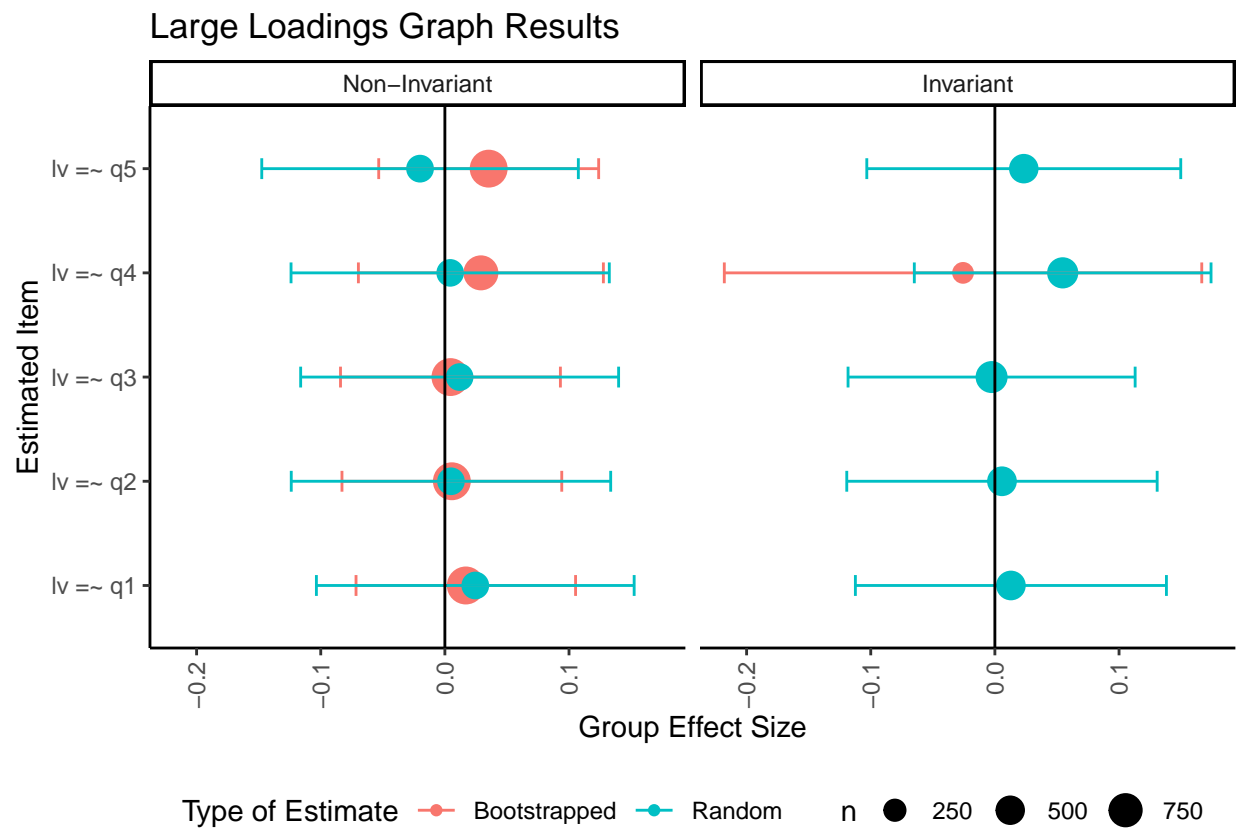
1181

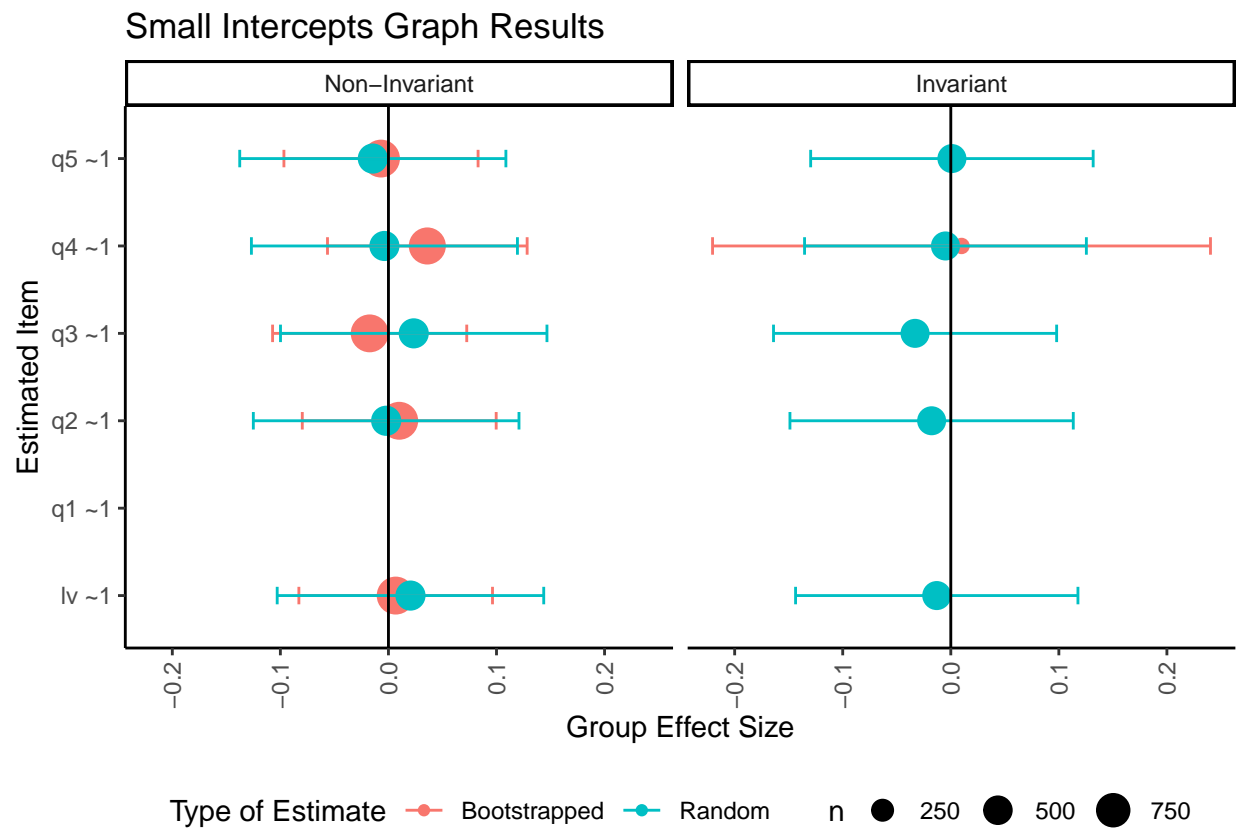
Small Loadings Graph Results

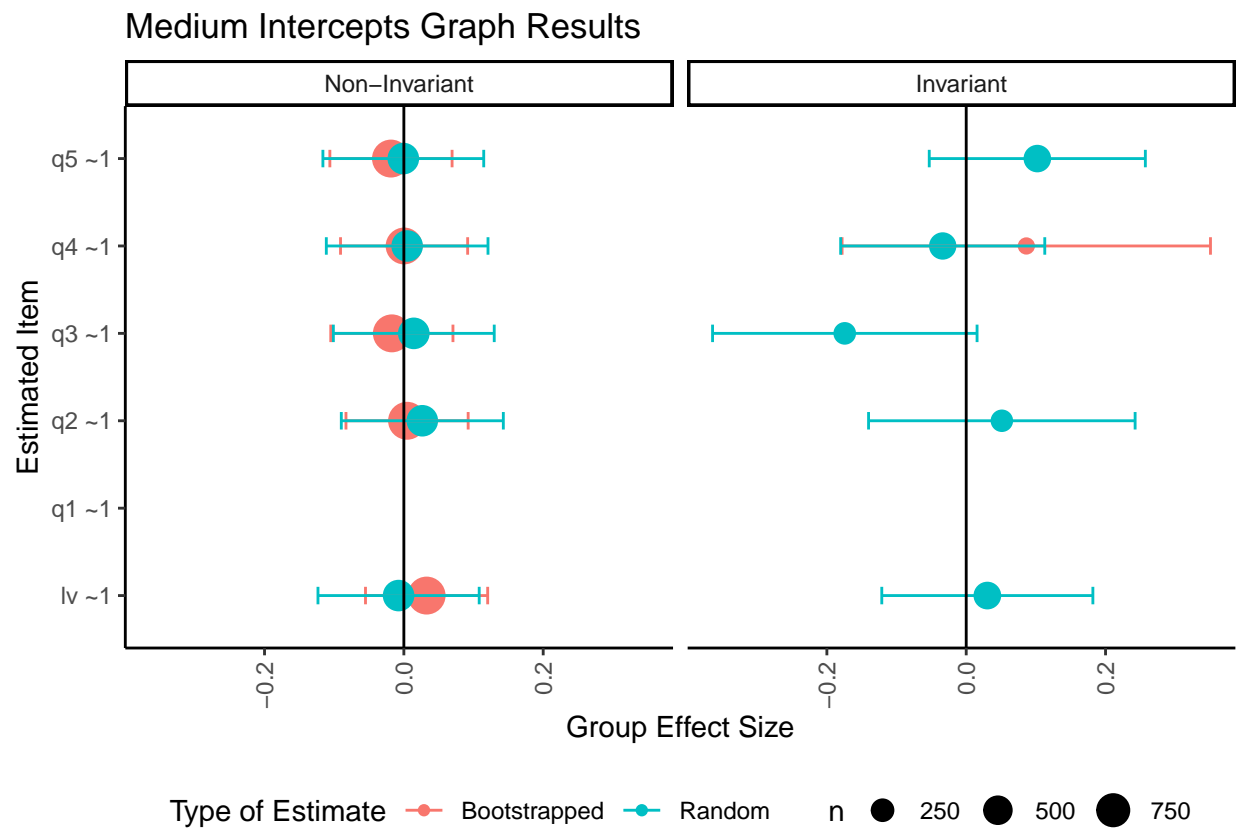


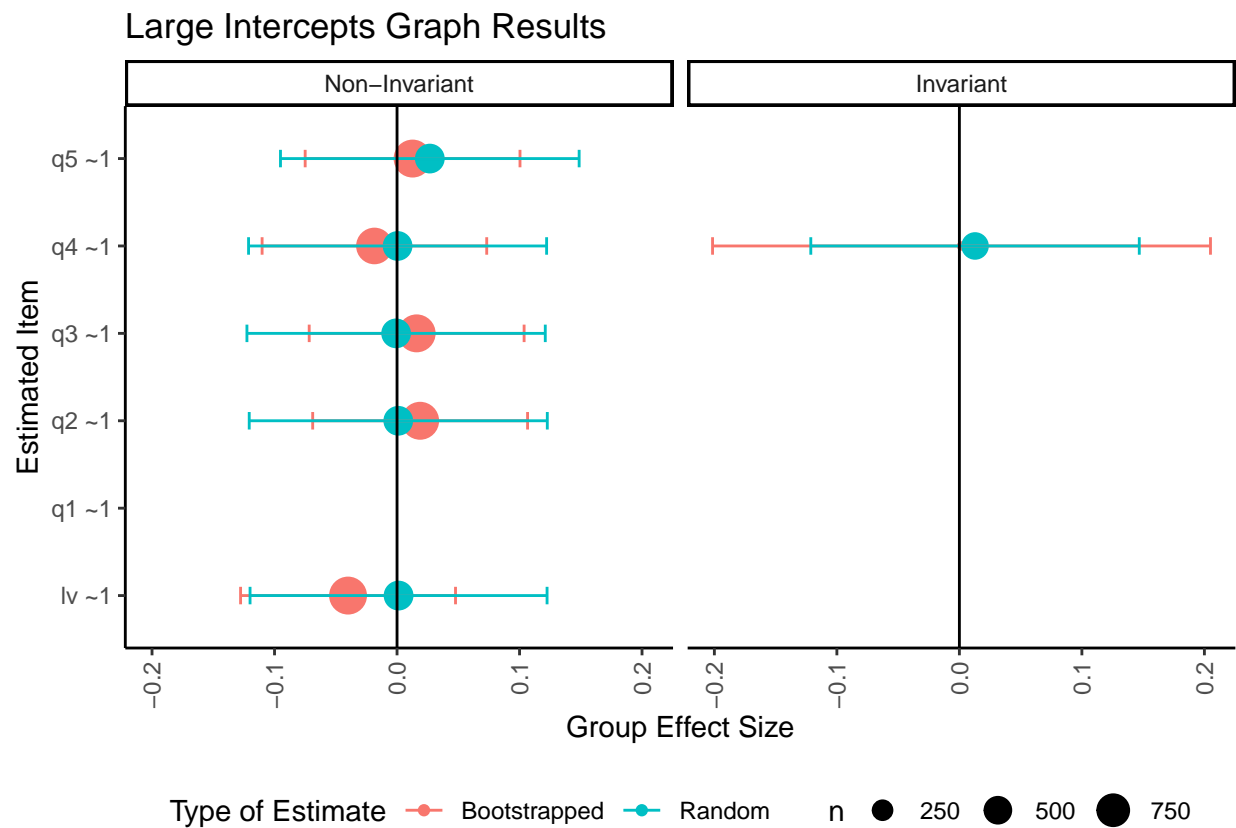
1182

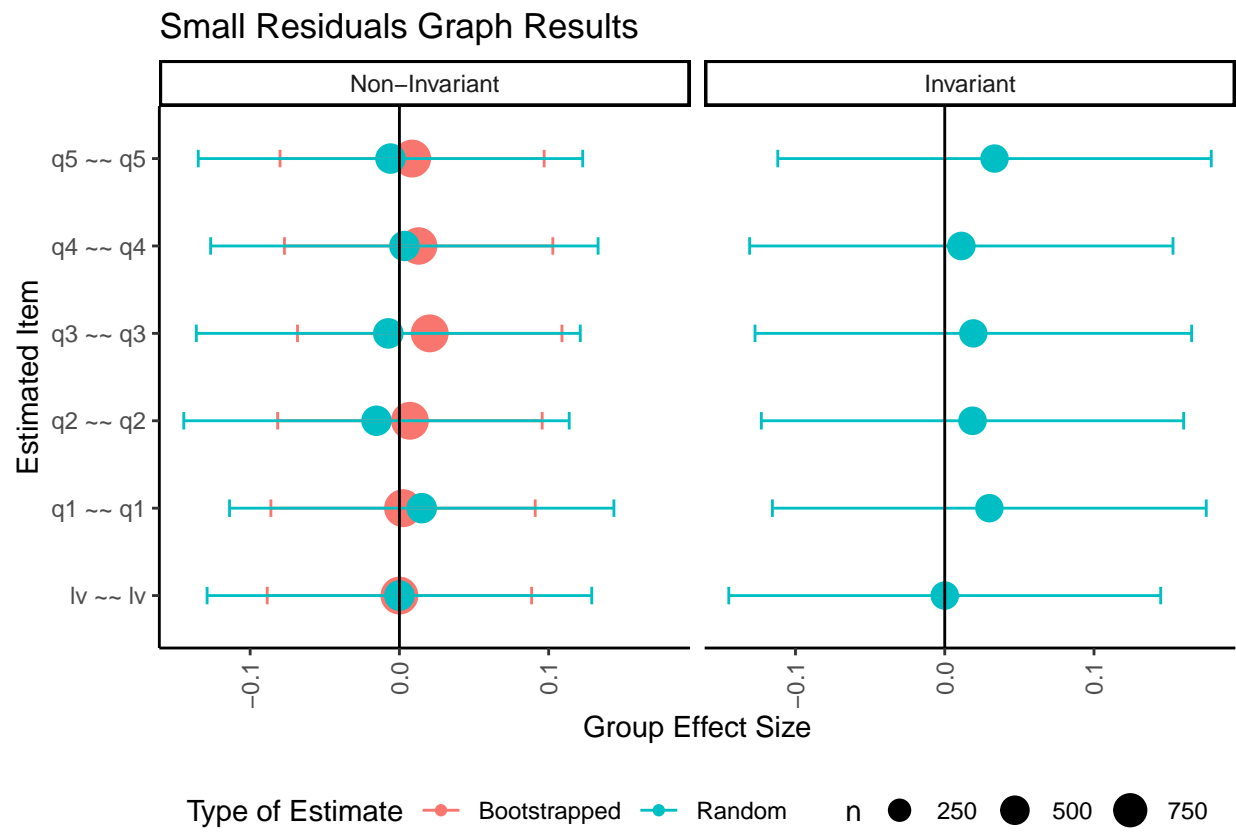


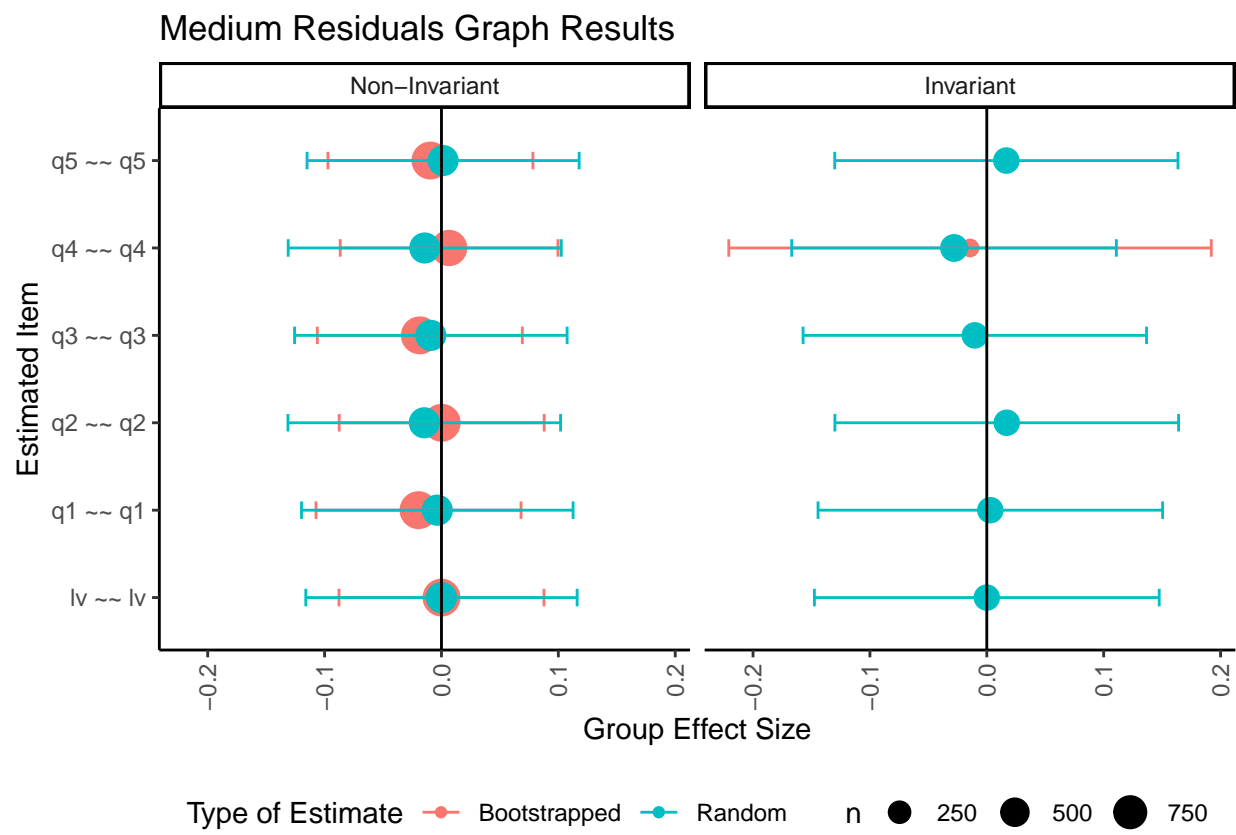


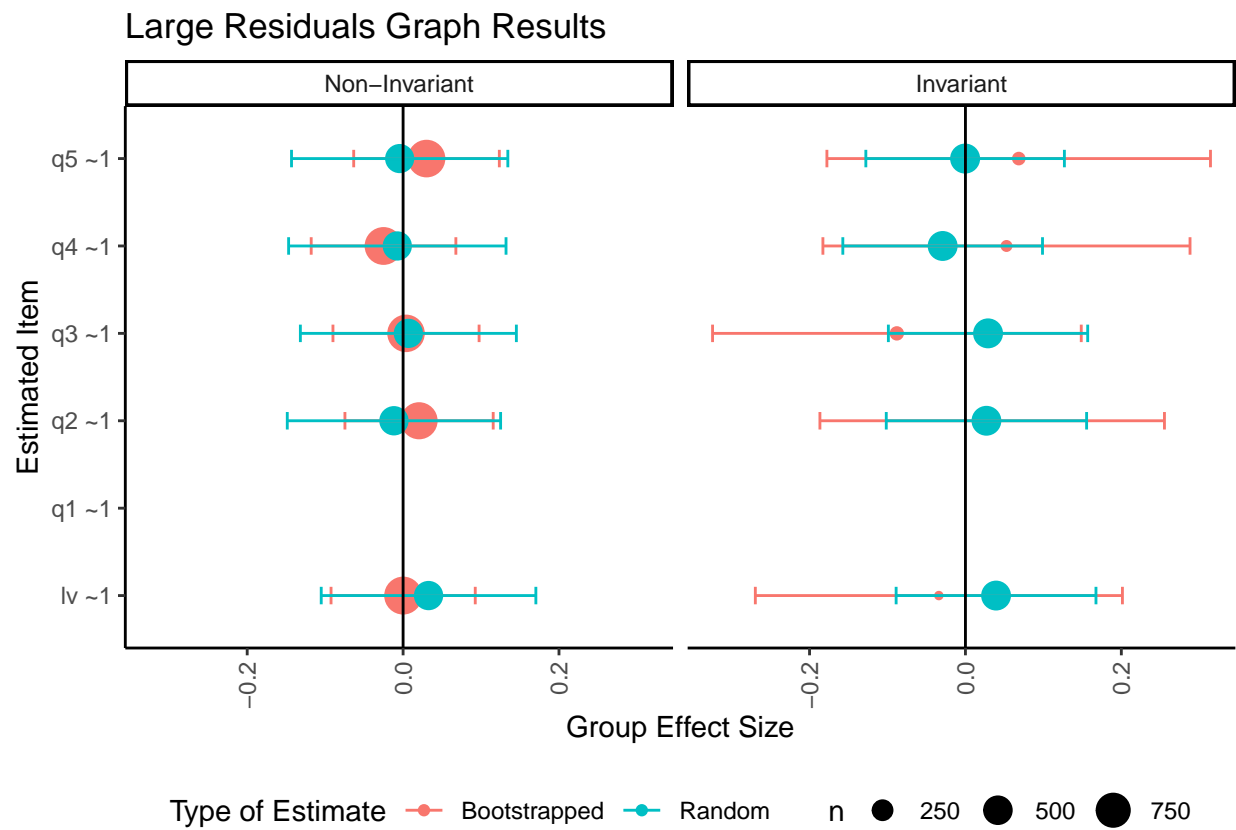






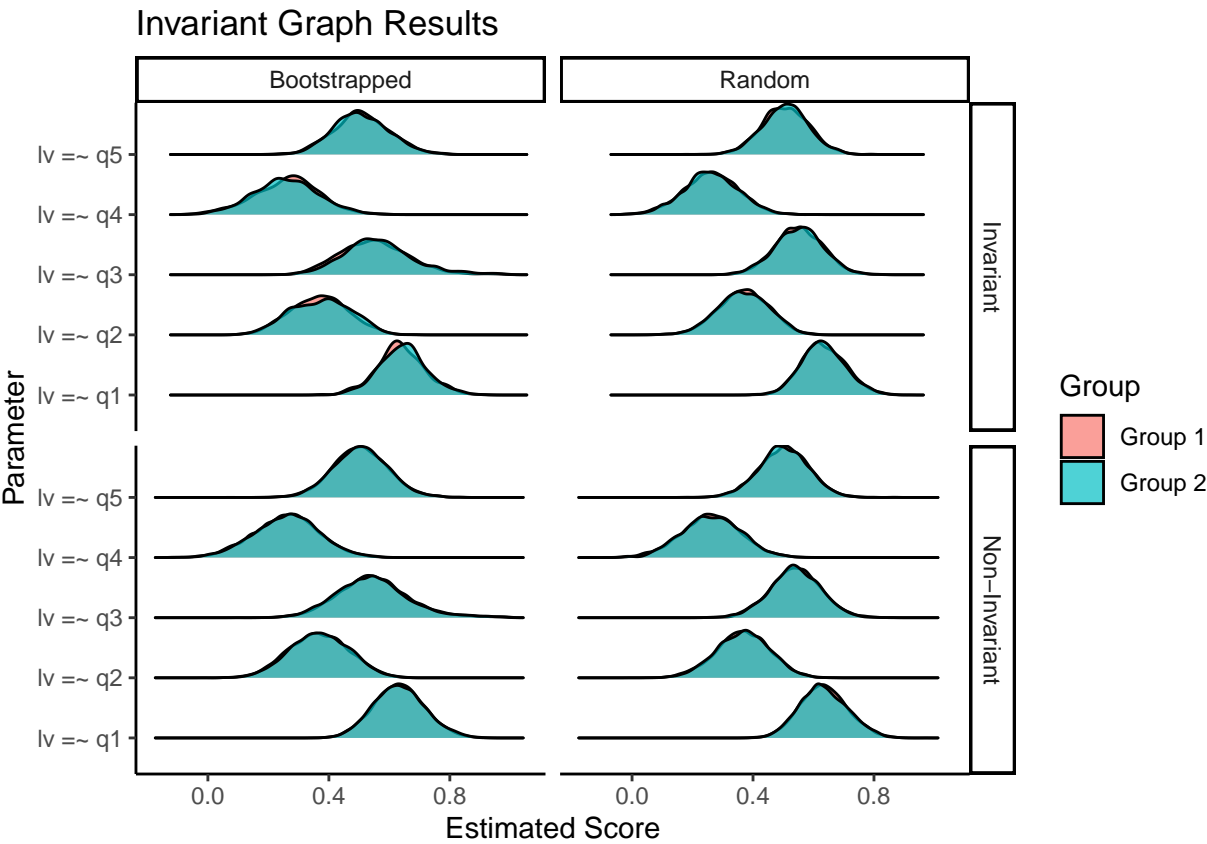






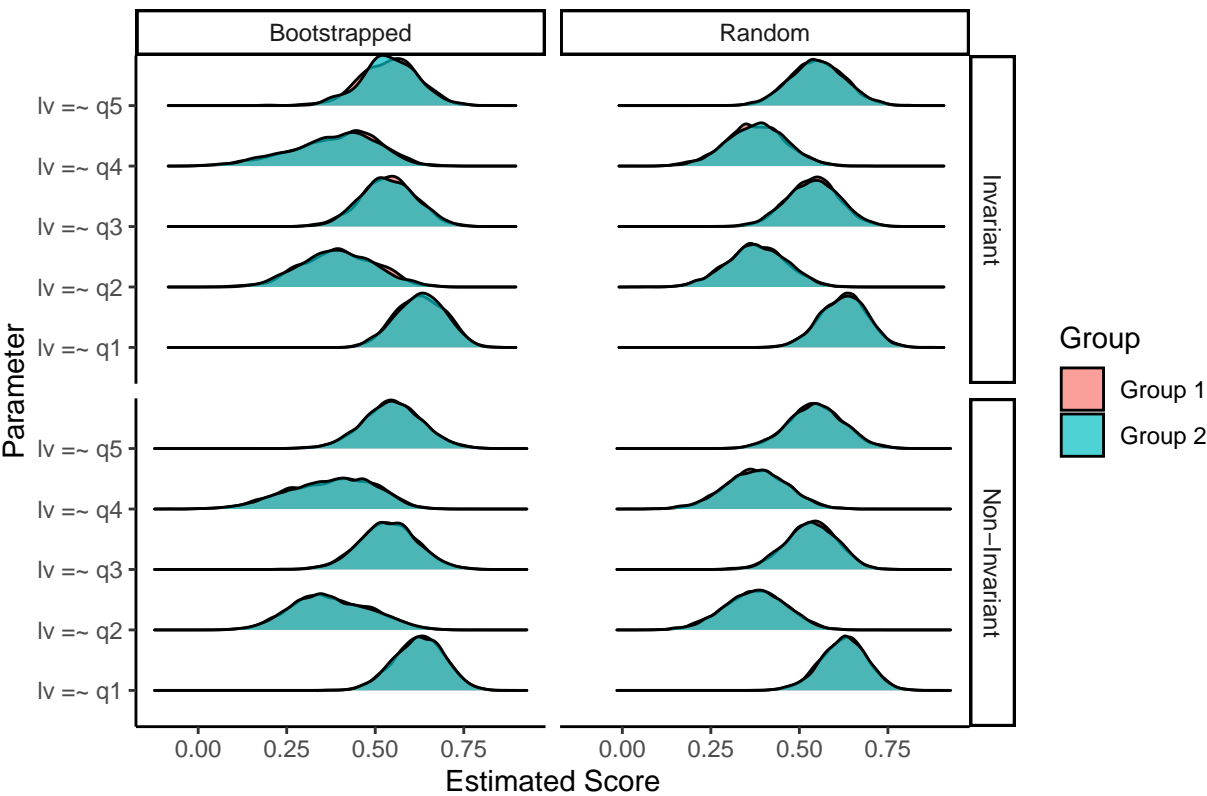
Appendix F

Density Plots by Condition

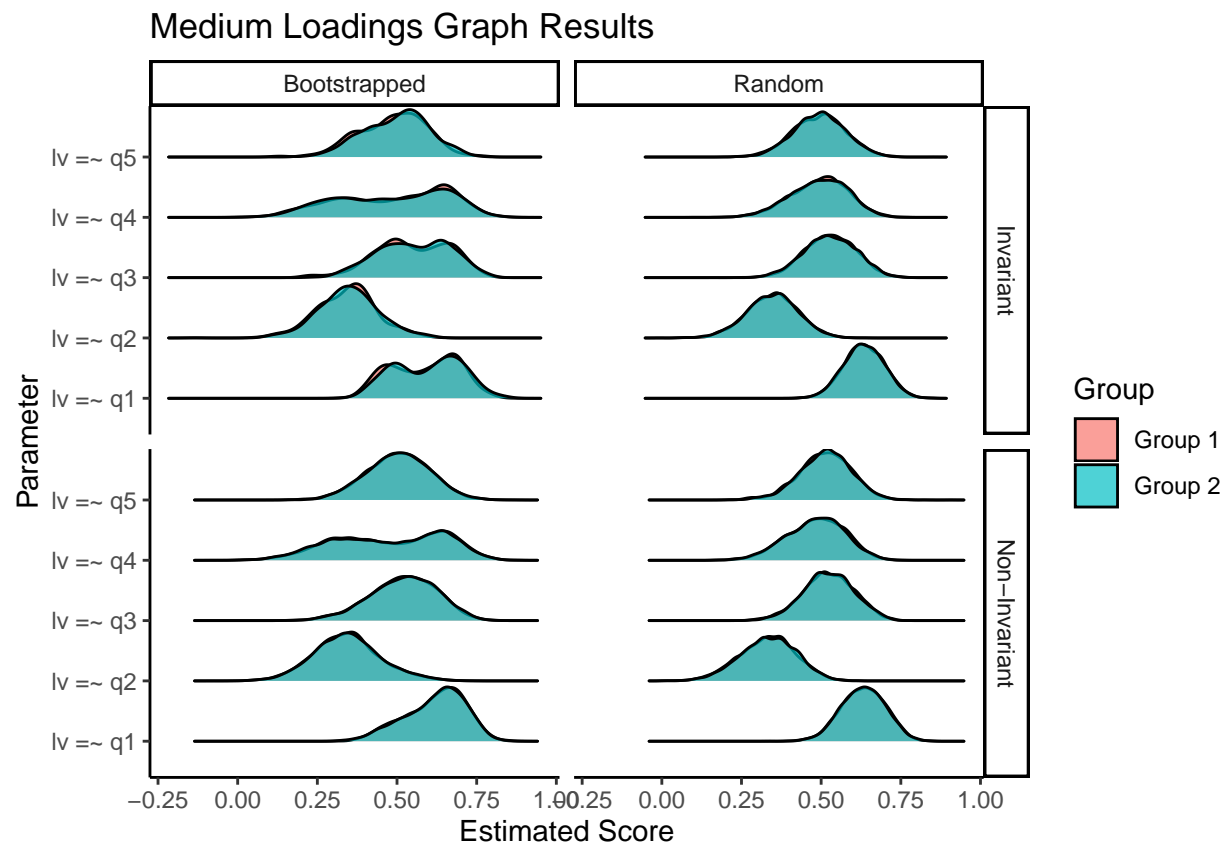


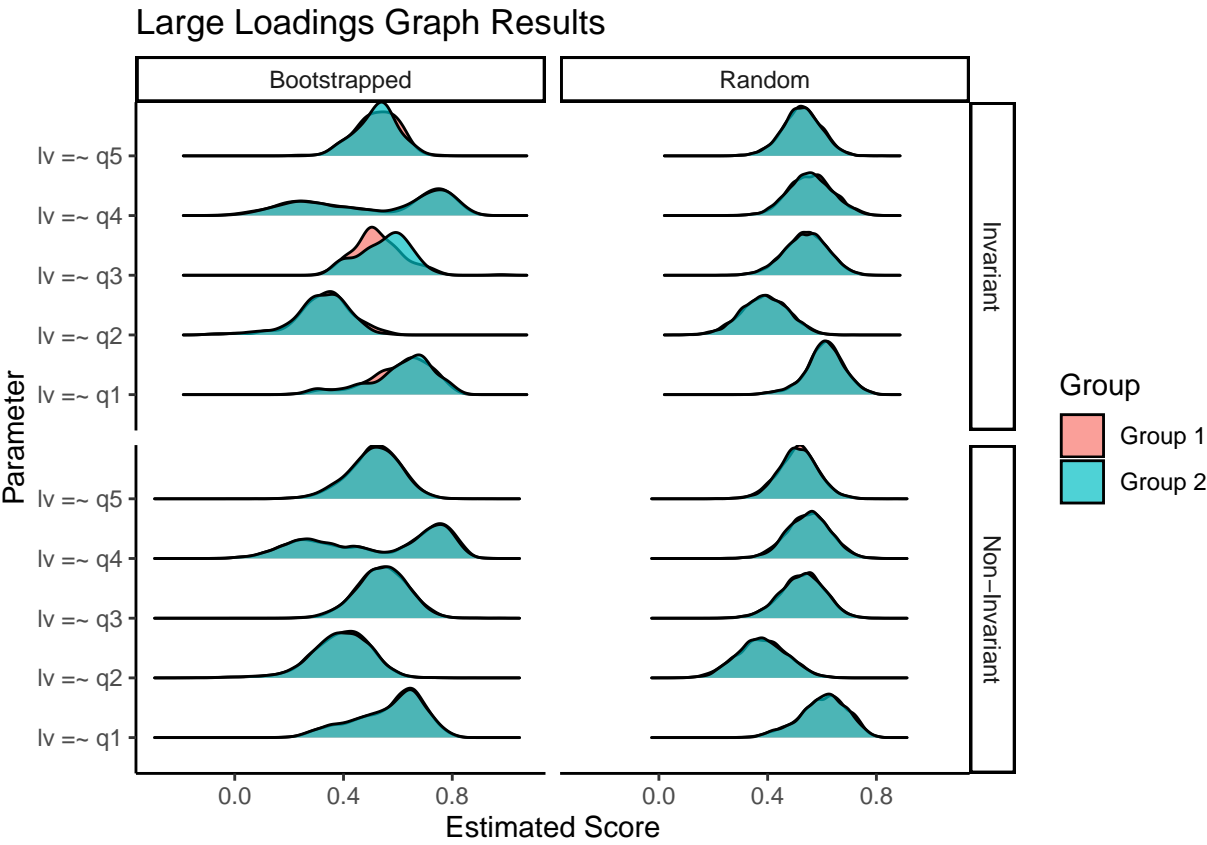
1191

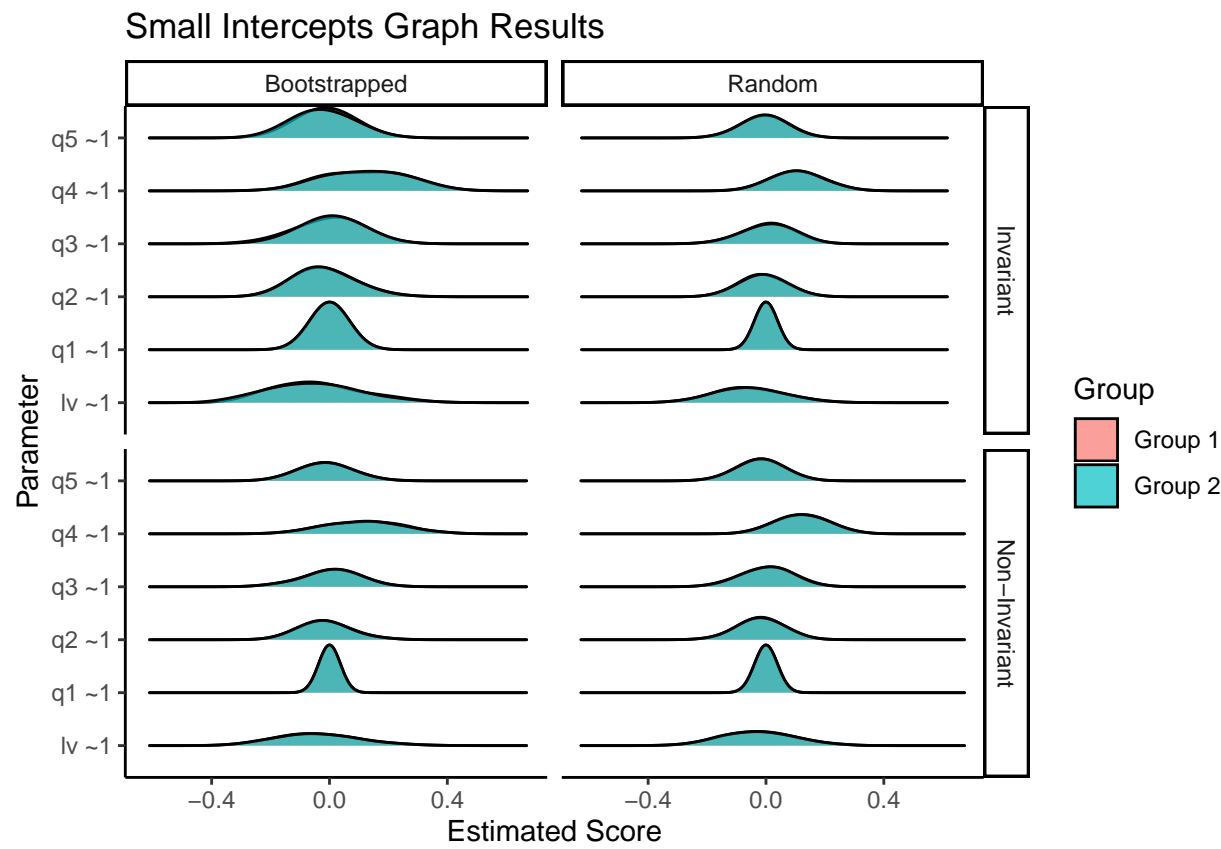
Small Loadings Graph Results

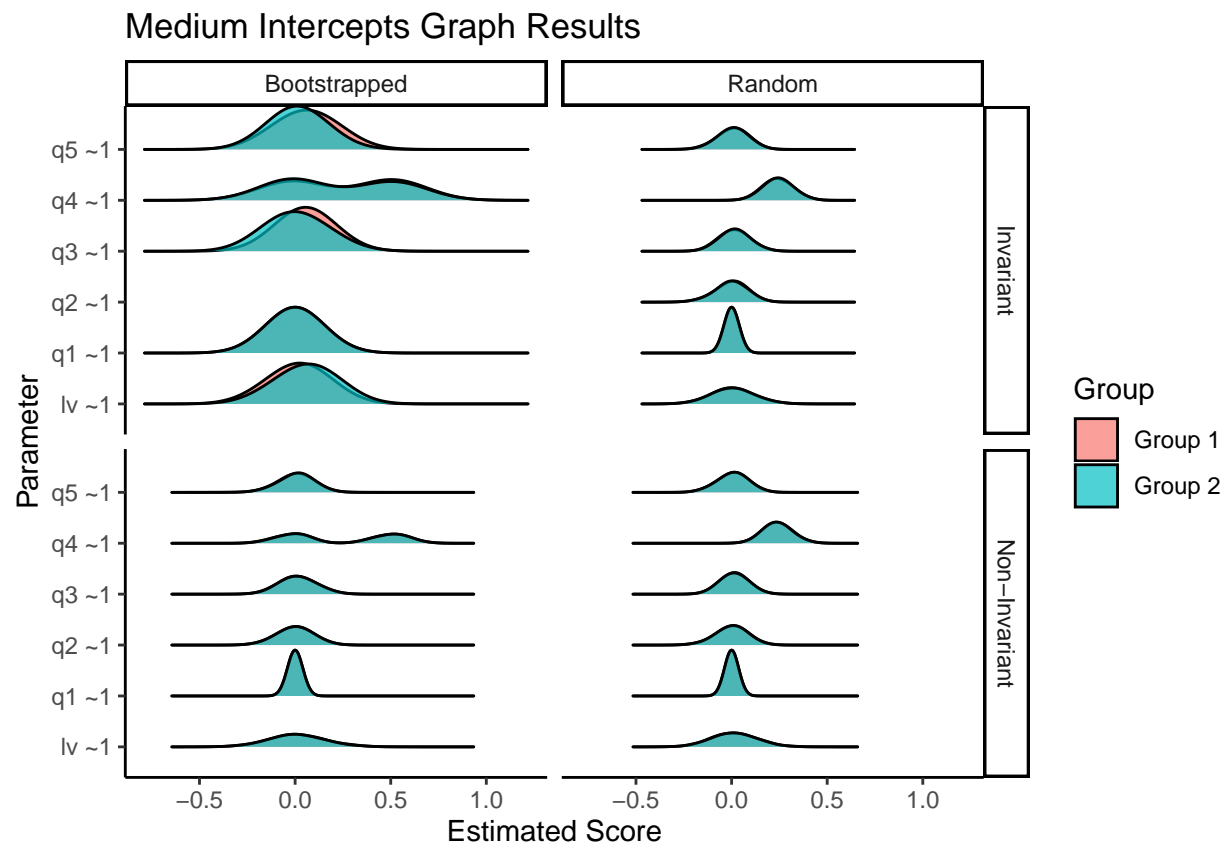


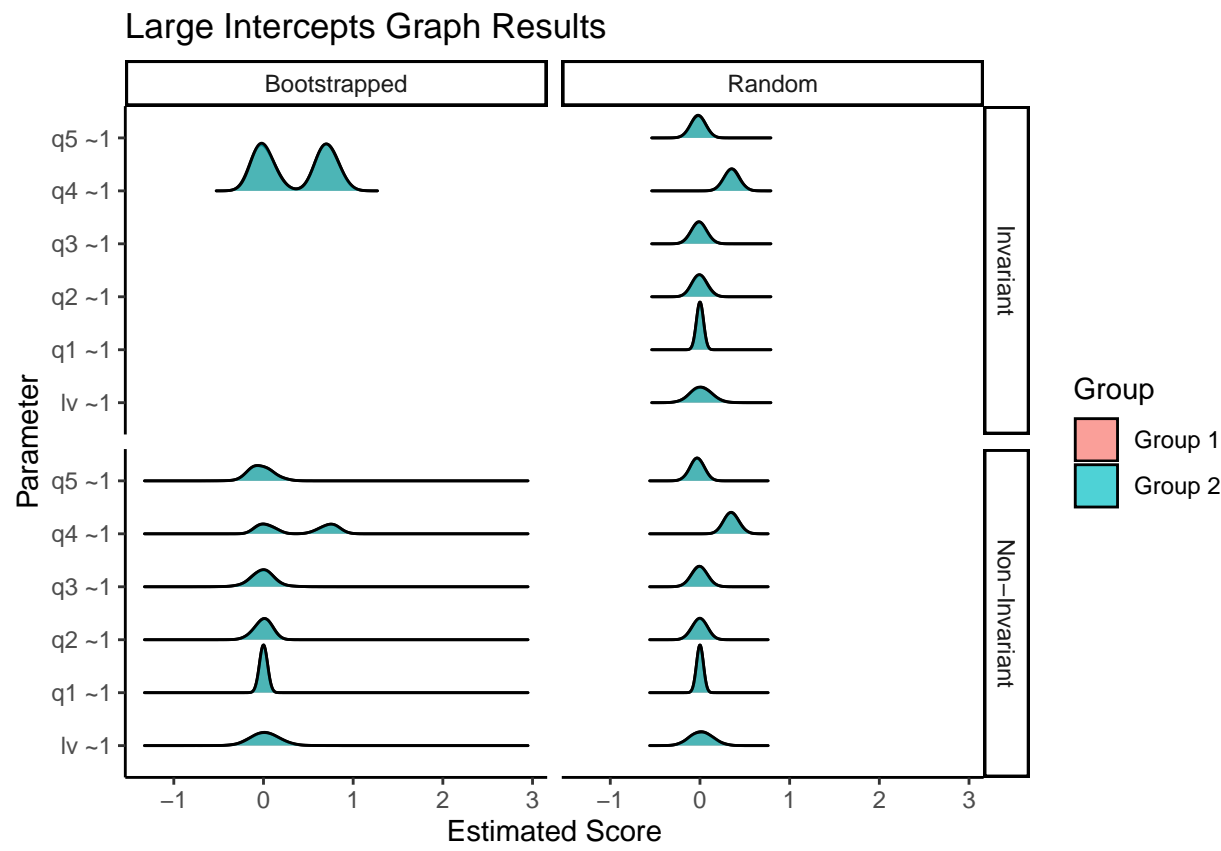
1192

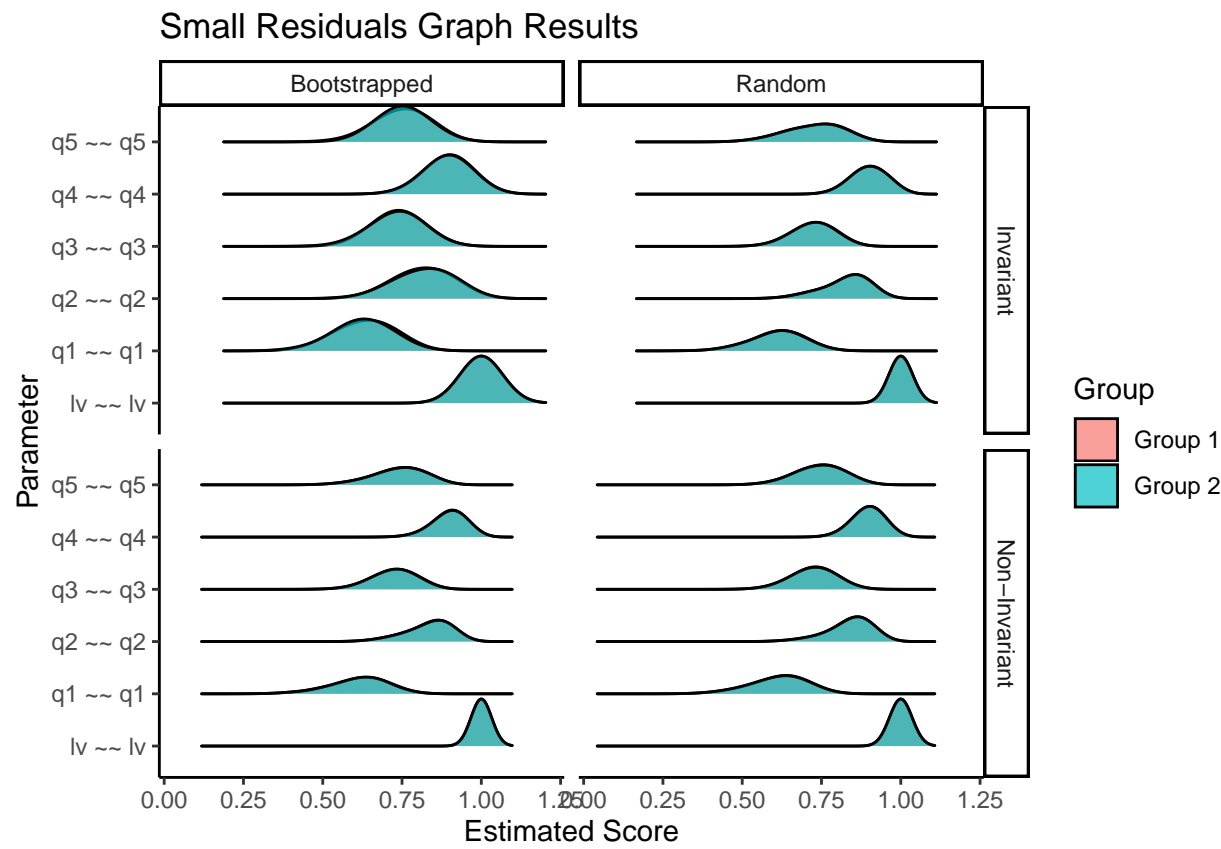


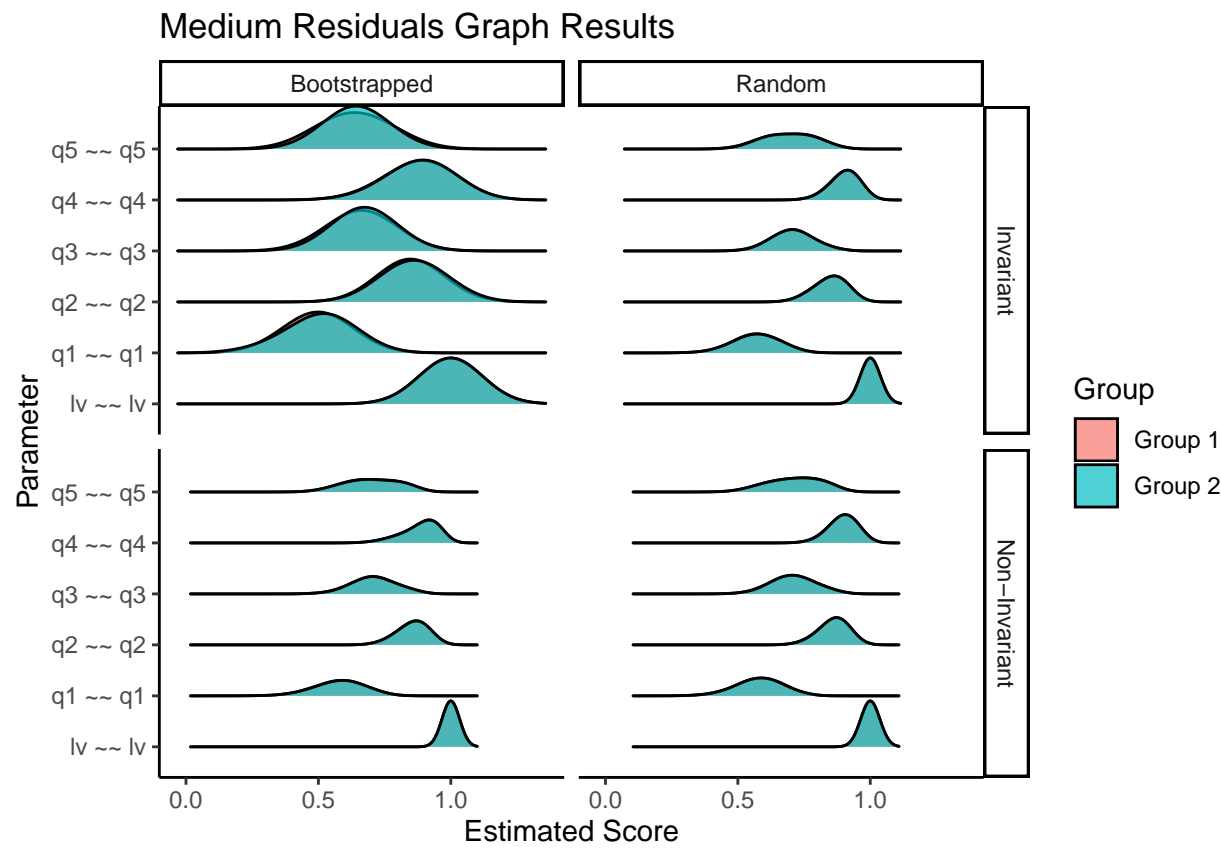


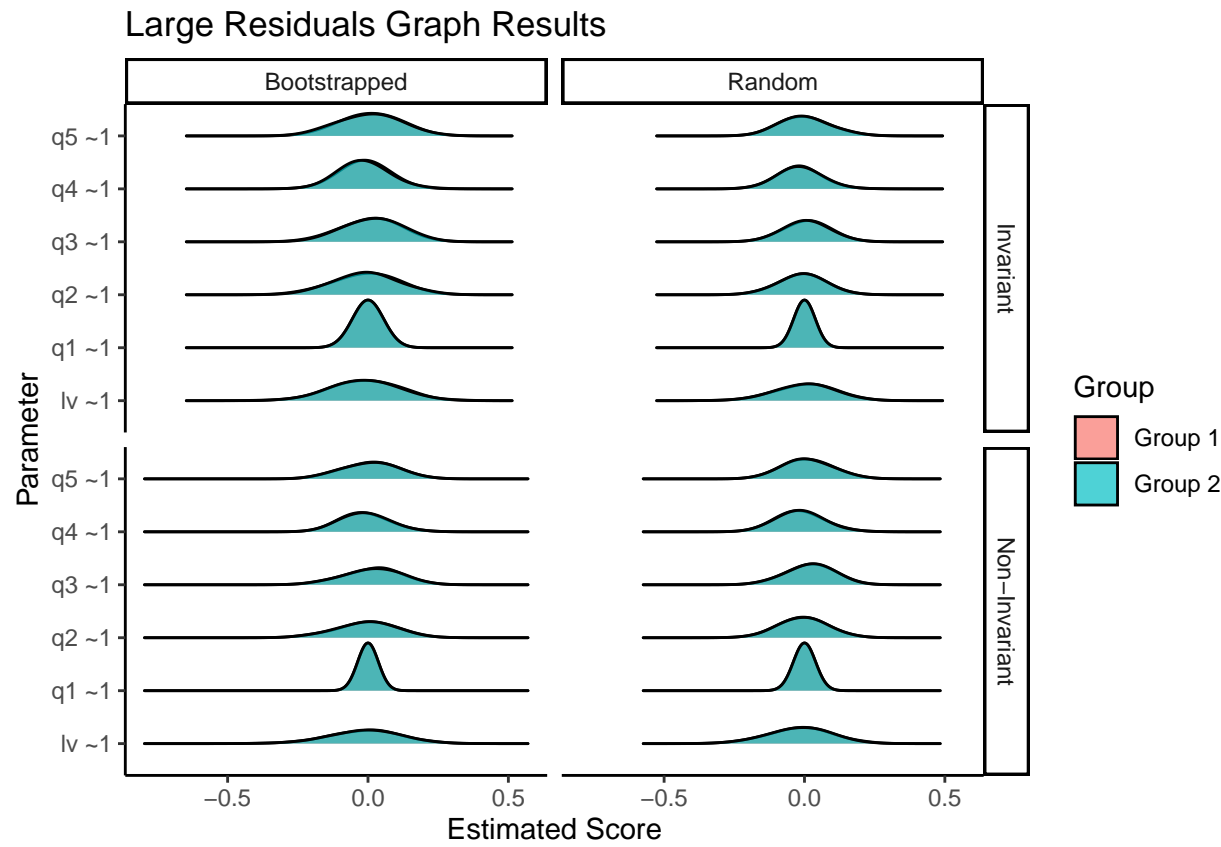












Replication Test

Data

```
## # A tibble: 2 x 2
##   group sample
##   <chr>   <int>
## 1 Aiena   1765
## 2 Chen    1010
```

MGCFA

```
## # A tibble: 7 x 7
##   model          AIC      BIC   cfi   tli  rmsea  srmr
##   <chr>          <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Overall      113442. 113608. 0.939 0.928 0.0890 0.0344
## 2 Group Schulenberg 69101.  69254. 0.928 0.915 0.108  0.0353
```

```

1214 ## 3 Group Chen          42603.  42741.  0.929 0.916 0.0766 0.0399
1215 ## 4 Configural          111760. 112258. 0.929 0.916 0.0975 0.0347
1216 ## 5 loadings            111789. 112210. 0.927 0.921 0.0946 0.0427
1217 ## 6 intercepts          112785. 113129. 0.892 0.891 0.111  0.0599
1218 ## 7 residuals            113318. 113579. 0.873 0.880 0.116  0.0661

```

```

1219      Overall, the one-factor model fits the data well. Each group also shows adequate
1220 model fit. If we use  $\Delta CFI \leq .01$ , we find that the loadings would be considered invariant
1221 across the English and Chinese samples. The intercepts were not invariant.

```

1222 Partial Invariance

```

1223 ## # A tibble: 15 x 2
1224 ##   free.parameter cfi
1225 ##   <chr>         <lvn.vctr>
1226 ## 1 "RS2 ~1 "      0.8970997
1227 ## 2 "RS11 ~1 "     0.8963948
1228 ## 3 "RS4 ~1 "     0.8961938
1229 ## 4 "RS12 ~1 "    0.8960095
1230 ## 5 "RS14 ~1 "    0.8954496
1231 ## 6 "RS1 ~1 "     0.8954215
1232 ## 7 "RS10 ~1 "    0.8953395
1233 ## 8 "RS3 ~1 "     0.8948455
1234 ## 9 "RS5 ~1 "     0.8945236
1235 ## 10 "RS7 ~1 "    0.8941450
1236 ## 11 "RS8 ~1 "    0.8920022
1237 ## 12 "RS13 ~1 "   0.8919491
1238 ## 13 "RS9 ~1 "    0.8919293
1239 ## 14 "RS ~1 "     0.8917757

```

```
1240 ## 15 "RS6 ~1 " 0.8917604
```

```
1241     Examining partial invariance reveals several potential candidates for partial
1242 invariance. In this next section, we relaxed group constraints until we achieved partial
1243 invariance (i.e.,  $\Delta\text{CFI} \leq .01$ ). We will need to find our CFI as at least 0.92. More than
1244 half the items are necessary to achieve “partial” invariance (which really implies no
1245 invariance is likely possible).
```

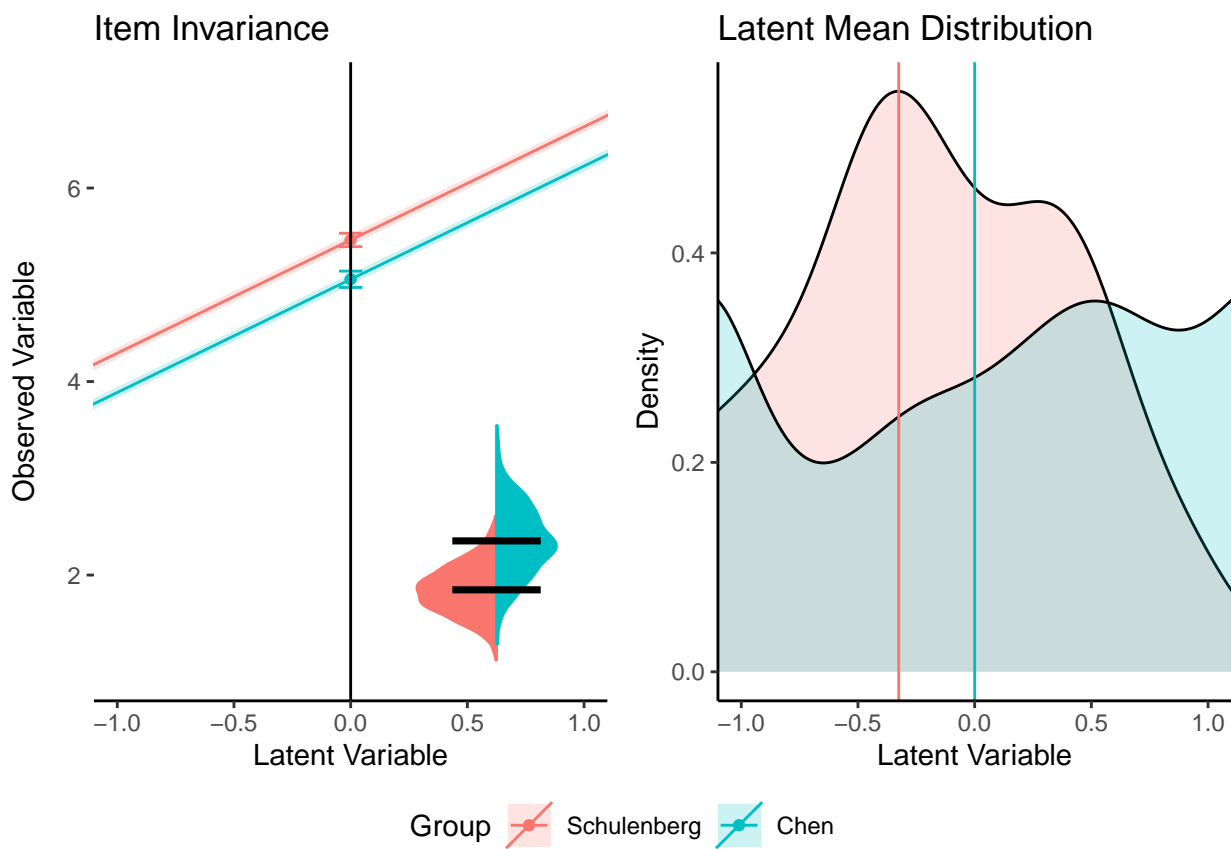
```
1246 ## # A tibble: 1 x 6
1247 ##       AIC      BIC   cfi   tli  rmsea   srmr
1248 ##   <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>
1249 ## 1 111935. 112326. 0.922 0.917 0.0966 0.0462
```

```
1250 ## # A tibble: 6 x 3
1251 ##   term      English Chinese
1252 ##   <chr>      <dbl>   <dbl>
1253 ## 1 "RS1 ~1 "    5.24    4.78
1254 ## 2 "RS3 ~1 "    5.14    5.46
1255 ## 3 "RS4 ~1 "    5.29    5.71
1256 ## 4 "RS5 ~1 "    5.13    5.13
1257 ## 5 "RS7 ~1 "    5.26    5.26
1258 ## 6 "RS12 ~1 "   5.57    5.16
```

```
1259 ## RS1 RS2 RS3 RS4 RS5 RS6 RS7 RS8 RS9 RS10 RS11 RS12 RS13 RS14
1260 ## 0.32 0.29 0.23 0.30 0.00 0.00 0.00 0.00 0.00 0.29 0.27 0.29 0.00 0.26
```

```
1261 Visualize Invariance
```

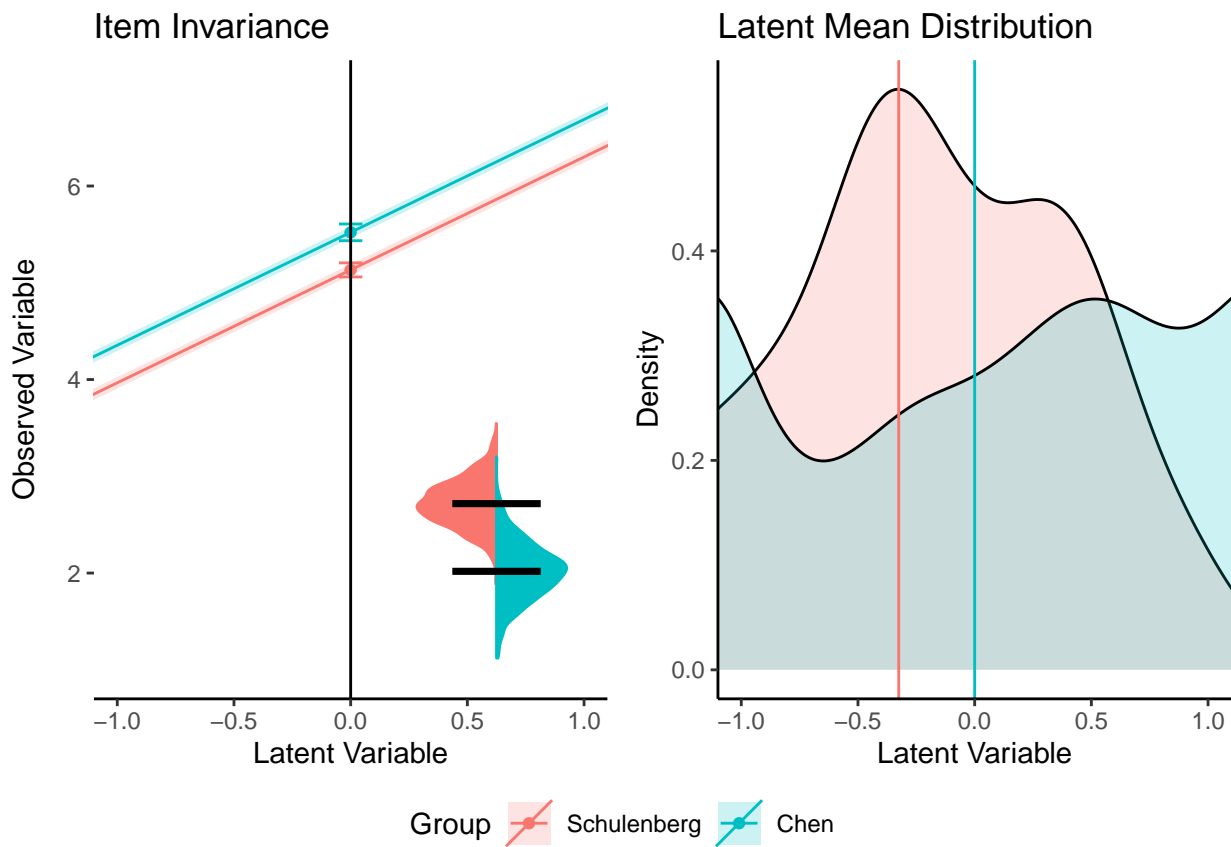
```
1262 ## Plot for RS2
```



1263

1264 ## -----

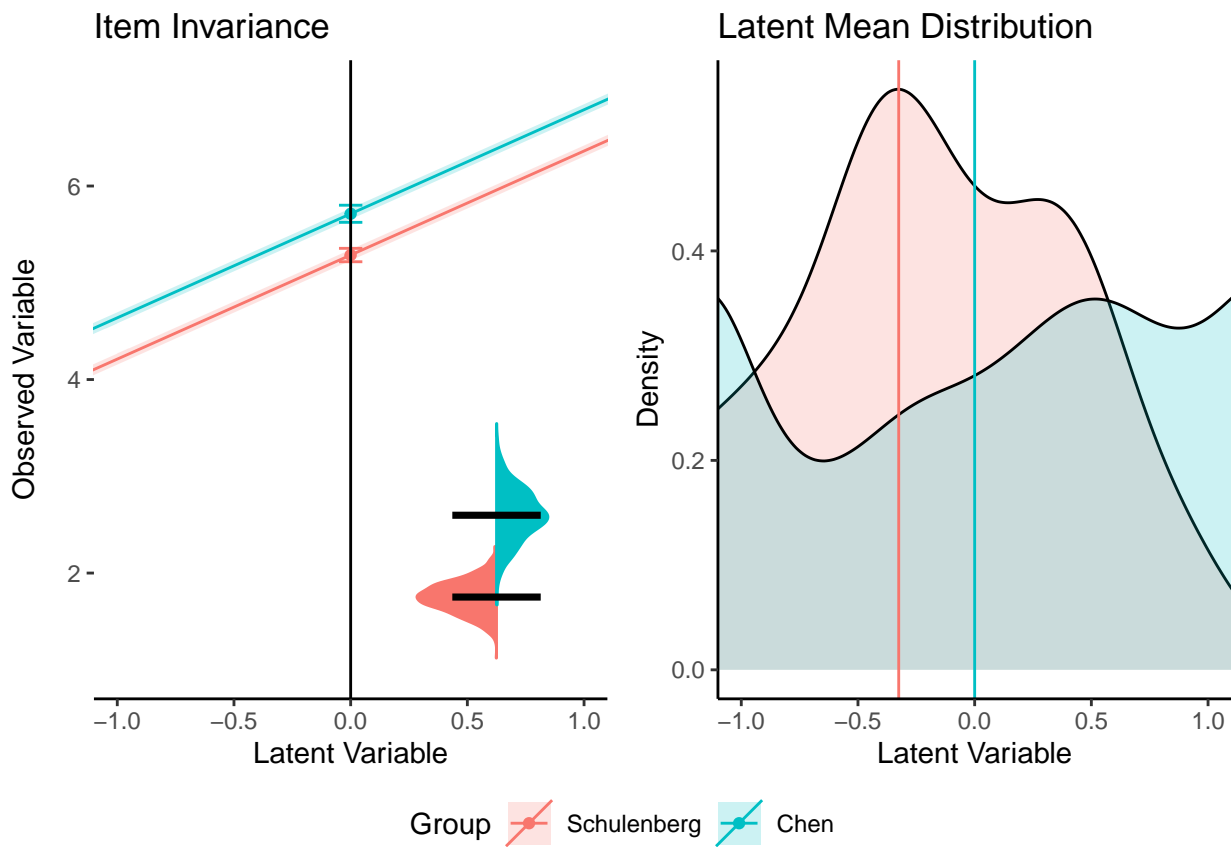
1265 ## Plot for RS11



1266

1267 ## -----

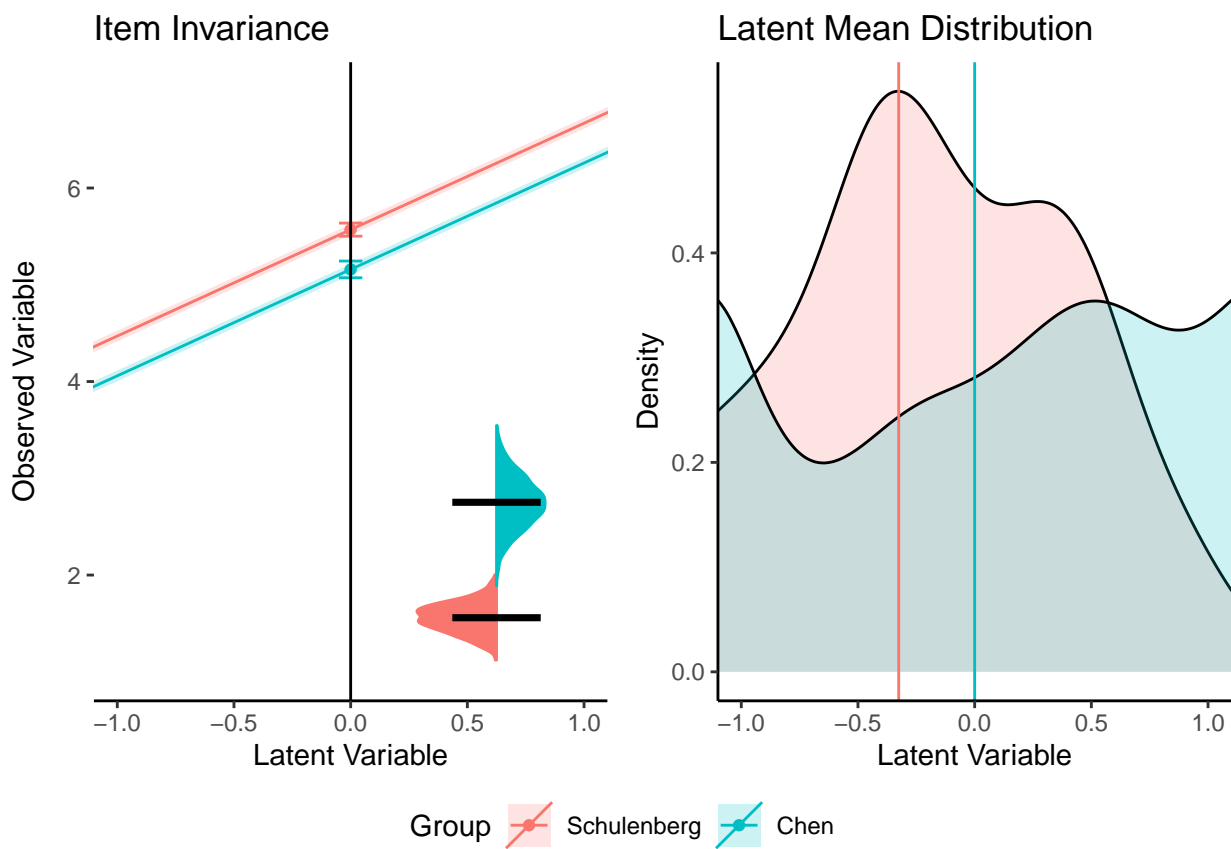
1268 ## Plot for RS4



1269

1270 ## -----

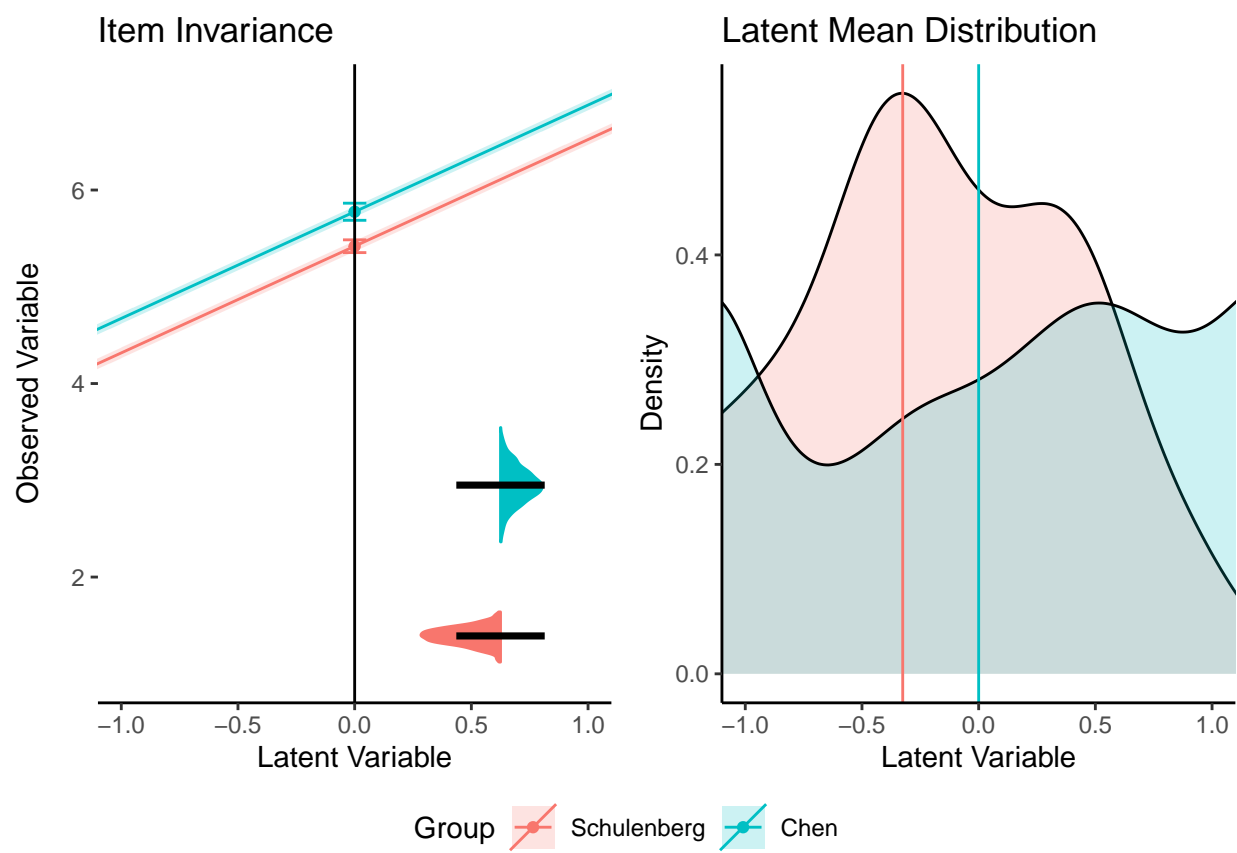
1271 ## Plot for RS12



1272

1273 ## -----

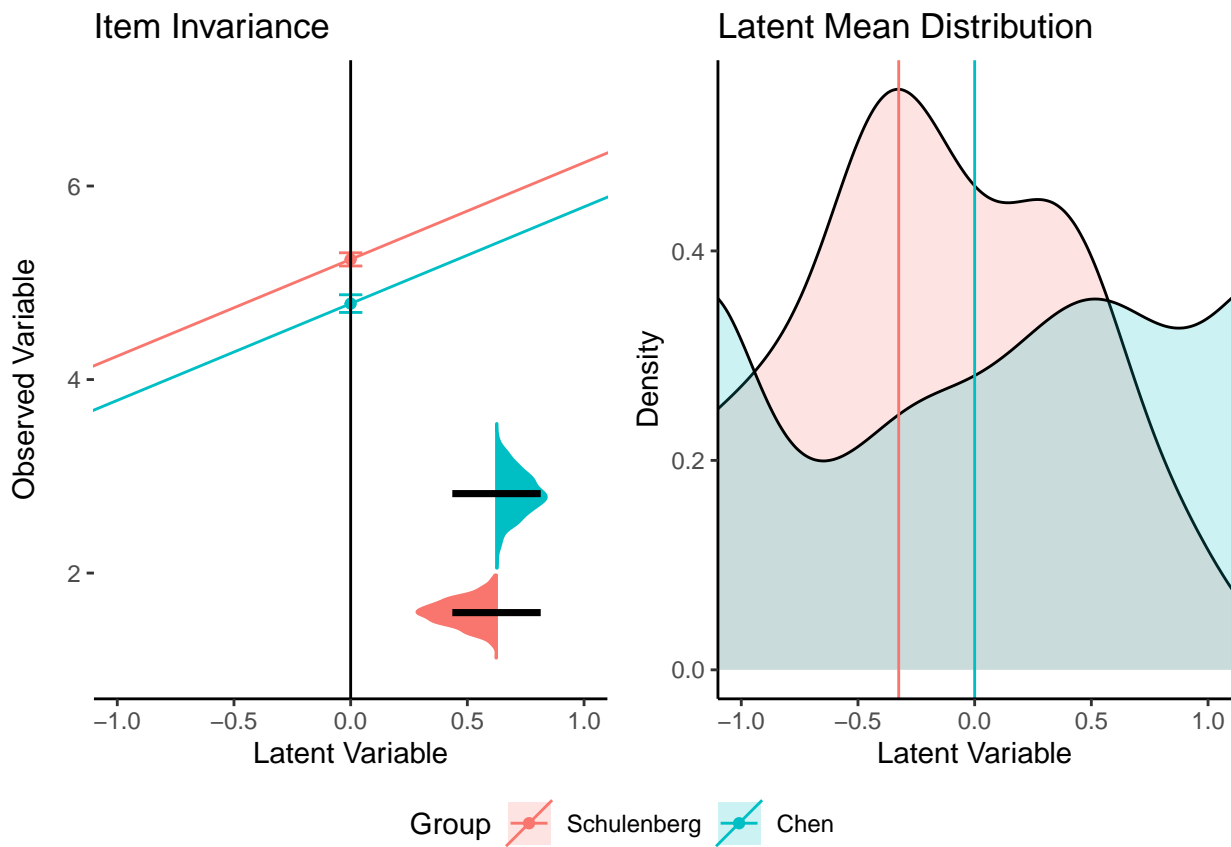
1274 ## Plot for RS14



1275

1276 ## -----

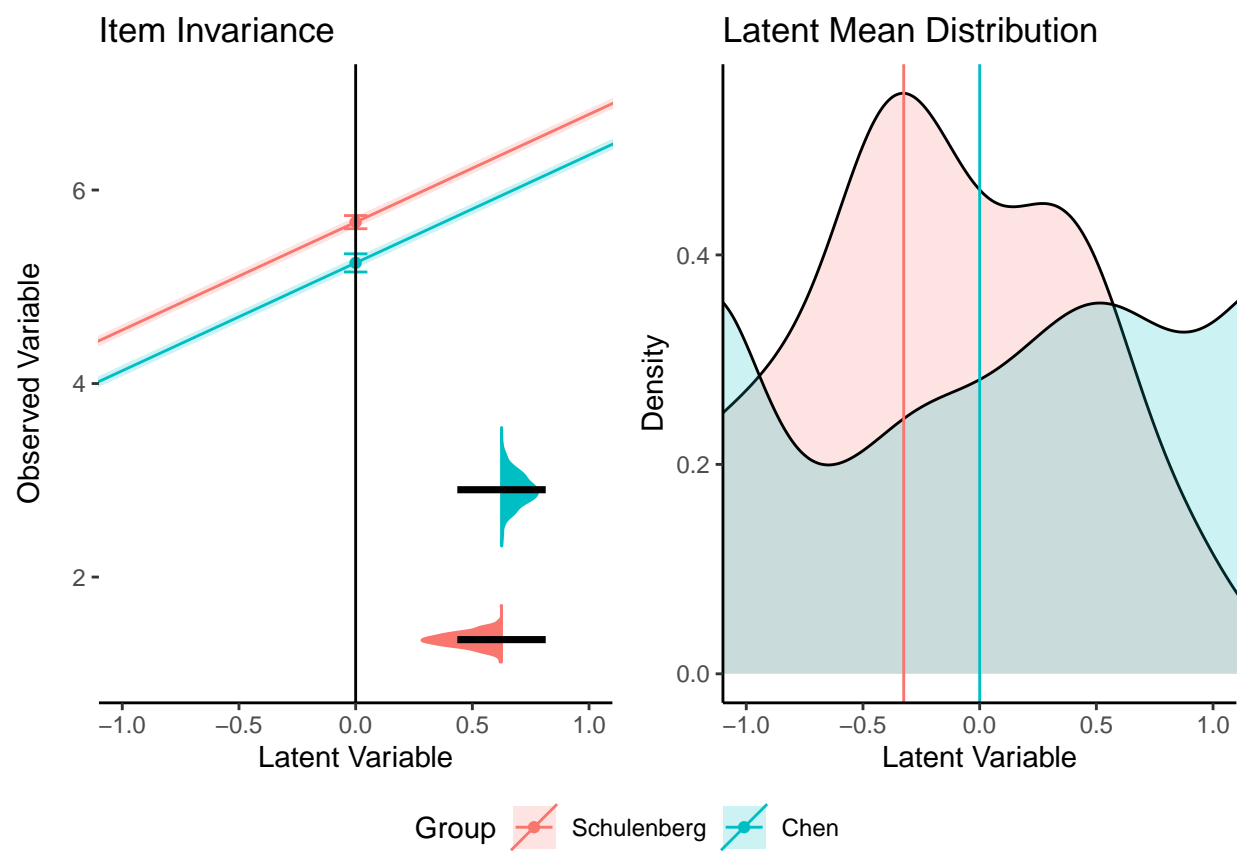
1277 ## Plot for RS1



1278

1279 ## -----

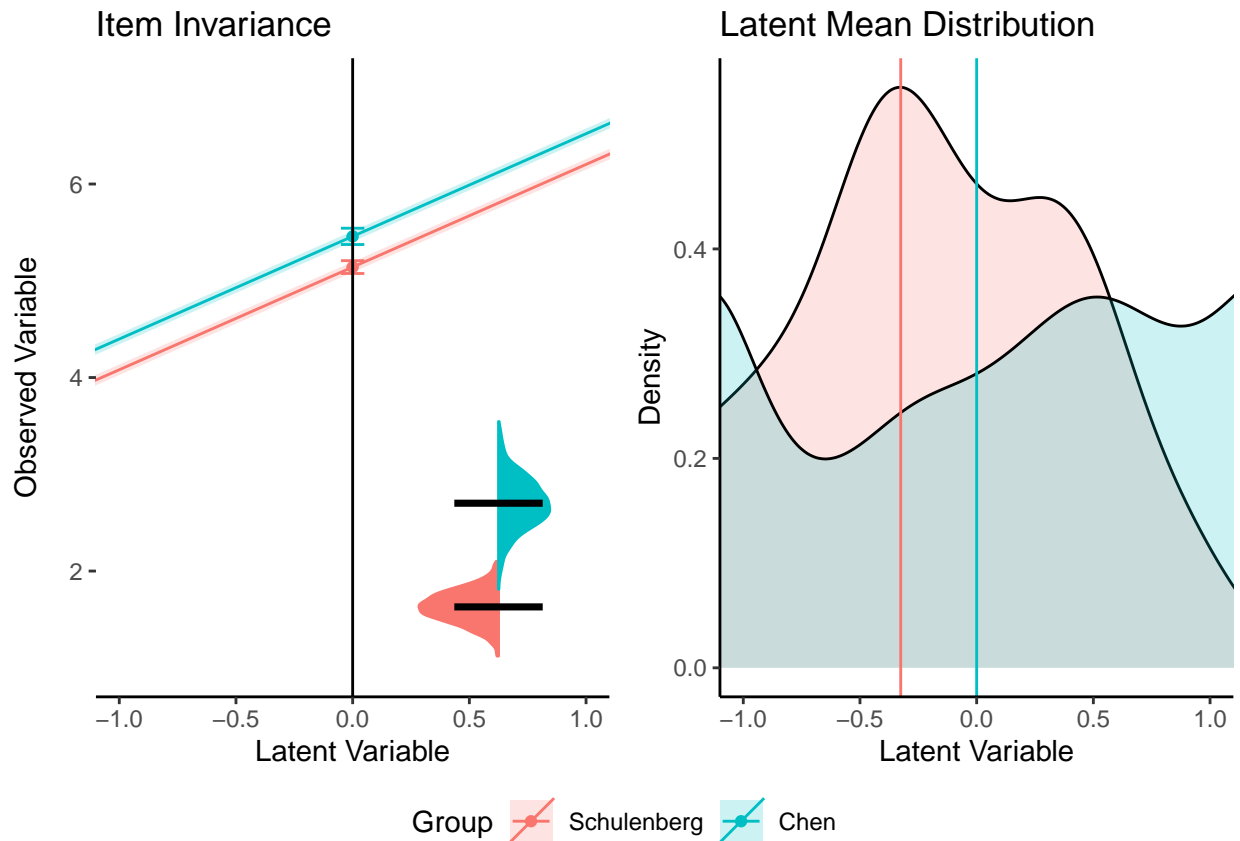
1280 ## Plot for RS10



1281

1282 ## -----

1283 ## Plot for RS3



1284

1285 ## -----

1286 Bootstrap Model

```
1287  ##          model non_invariant random_non_invariant      h_nmi h_nmi_p
```

1288	## 1 intercepts	1	0	3.141593	1
------	-----------------	---	---	----------	---

In this case, we do not see loadings print out. That implies that all models in both real data and randomized data are invariant because the function only calculates information for non-invariance. We see that the intercepts are unlikely to ever replicate across Chinese and English samples. This result is not surprising given the large number of relaxed parameters required to achieve partial invariance.

1294 Bootstrap Partial Invariance

1295 *Each Parameter on the Overall Model Invariance*

```
1296  ##          term non_invariant random_non_invariant      h_nmi  h_nmi_p
```

1297	## 1	RS ~1	1	0.002 3.052120 0.971520
1298	## 2	RS1 ~1	1	0.001 3.078337 0.979865
1299	## 3	RS10 ~1	1	0.002 3.052120 0.971520
1300	## 4	RS11 ~1	1	0.002 3.052120 0.971520
1301	## 5	RS12 ~1	1	0.002 3.052120 0.971520
1302	## 6	RS13 ~1	1	0.002 3.052120 0.971520
1303	## 7	RS14 ~1	1	0.002 3.052120 0.971520
1304	## 8	RS2 ~1	1	0.002 3.052120 0.971520
1305	## 9	RS3 ~1	1	0.002 3.052120 0.971520
1306	## 10	RS4 ~1	1	0.002 3.052120 0.971520
1307	## 11	RS5 ~1	1	0.002 3.052120 0.971520
1308	## 12	RS6 ~1	1	0.002 3.052120 0.971520
1309	## 13	RS7 ~1	1	0.001 3.078337 0.979865
1310	## 14	RS8 ~1	1	0.002 3.052120 0.971520
1311	## 15	RS9 ~1	1	0.002 3.052120 0.971520

1312 In this output, we see that all the bootstrapped runs of the real data are
 1313 non-invariant, even when each parameter is relaxed individually. A few runs of the random
 1314 data are *non*-invariant (meaning most are actually invariant when randomized). This
 1315 indicates that no one parameter is likely the reason for non-invariance, as they all show large
 1316 non-replication effects. If we use the `boot_summary`, we can see the effect size for each
 1317 parameter when the two intercepts are compared to each other (as the chart above shows the
 1318 overall model invariance effect).

1319 *Each Parameter's Standardized Difference Score*

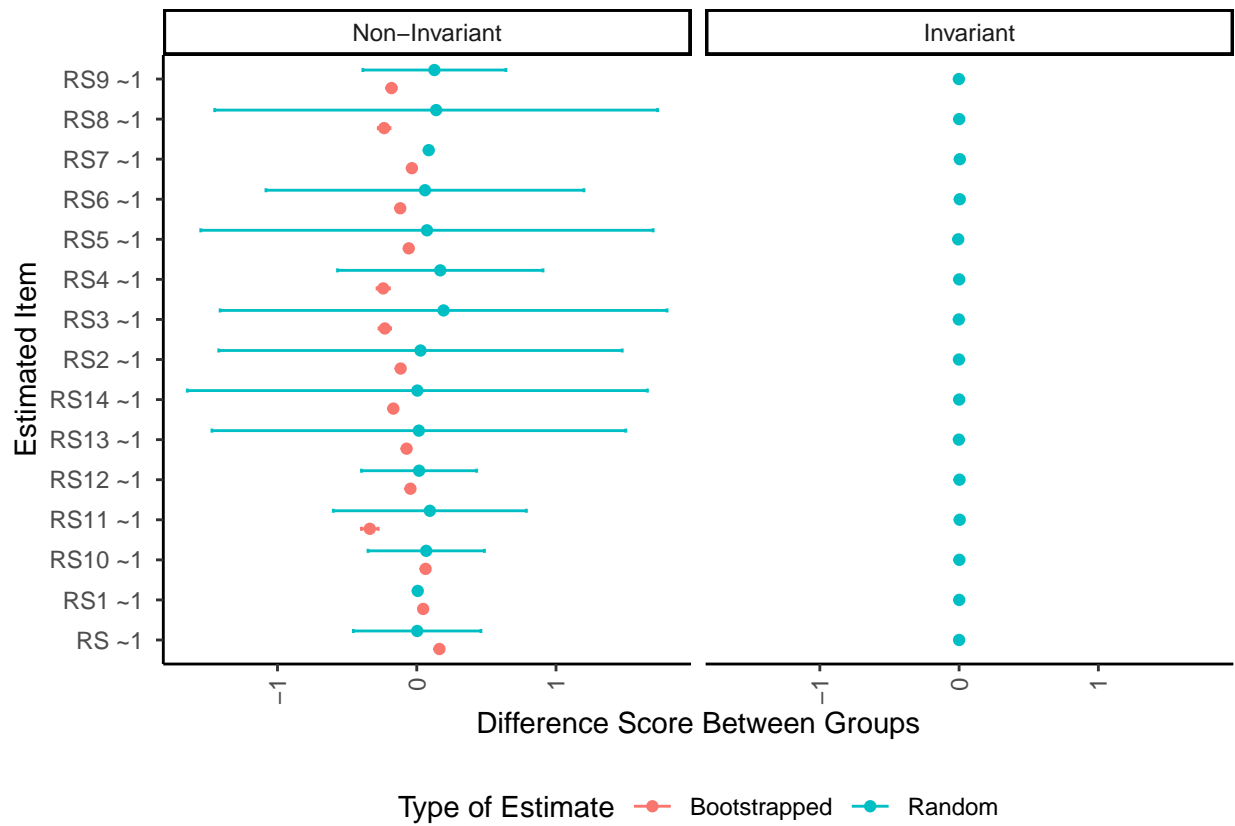
1320	##	term	invariant	n_boot	d_boot
1321	## 1	RS ~1	FALSE	1000	NA
1322	## 2	RS1 ~1	FALSE	1000	0.2913872

1323	## 3	RS10 ~1	FALSE	1000	0.3596961
1324	## 4	RS11 ~1	FALSE	1000	-0.6589317
1325	## 5	RS12 ~1	FALSE	1000	-0.3010202
1326	## 6	RS13 ~1	FALSE	1000	-0.4005088
1327	## 7	RS14 ~1	FALSE	1000	-0.5665988
1328	## 8	RS2 ~1	FALSE	1000	-0.5540349
1329	## 9	RS3 ~1	FALSE	1000	-0.6218628
1330	## 10	RS4 ~1	FALSE	1000	-0.6245830
1331	## 11	RS5 ~1	FALSE	1000	-0.3334520
1332	## 12	RS6 ~1	FALSE	1000	-0.5014247
1333	## 13	RS7 ~1	FALSE	1000	-0.2381644
1334	## 14	RS8 ~1	FALSE	1000	-0.6333750
1335	## 15	RS9 ~1	FALSE	1000	-0.5986866

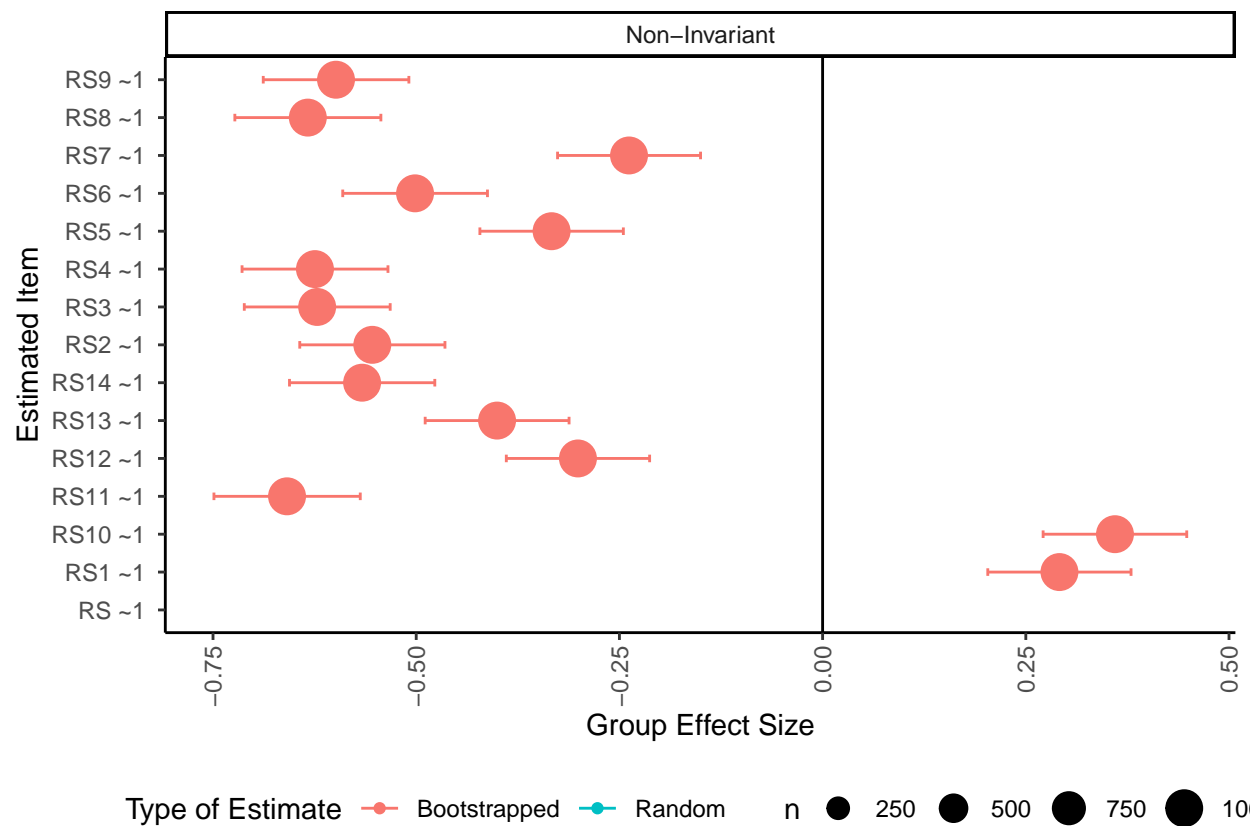
1336	##	term	invariant	n_random	d_random
1337	## 1	RS ~1	FALSE	2	NA
1338	## 2	RS1 ~1	FALSE	1	NA
1339	## 3	RS10 ~1	FALSE	2	NA
1340	## 4	RS11 ~1	FALSE	2	NA
1341	## 5	RS12 ~1	FALSE	2	NA
1342	## 6	RS13 ~1	FALSE	2	NA
1343	## 7	RS14 ~1	FALSE	2	NA
1344	## 8	RS2 ~1	FALSE	2	NA
1345	## 9	RS3 ~1	FALSE	2	NA
1346	## 10	RS4 ~1	FALSE	2	NA
1347	## 11	RS5 ~1	FALSE	2	NA
1348	## 12	RS6 ~1	FALSE	2	NA
1349	## 13	RS7 ~1	FALSE	1	NA

1363

We can view the mean difference or standardized mean difference by using:



1364



1365

1366

1367

1368

1369

When effects are non-invariant in the randomized data, the mean difference is still fairly small, but we see large mean differences in intercepts when the bootstrapped data is non-invariant. In the effect size graph, we can see that this effect is medium to large for all the parameters.