ASK Internal Technical Report

Danny Swan

May 31, 2018

Introduction

The goal of this technical report is to document my workflow in examining the characteristics of the ASK dataset as precisely as possible, in order to aid any future analysts who are interested in extending the work performed here. This document is strictly for internal consumption. Upon completion of this technical report, a shorter, revised version of the technical report will be produced for external consumption so that groups outside of MCPER can make use of the ASK in their own research. This document assumes that anyone who wants to follow up will have some working knowledge of R and MPLUS, and may need some slight alteration to account for the location of their data files.

This document is separate into several parts

- The main body of the text contains the primary IRT and DIF analyses.
- Appendix 1 Contains R code for the full analysis
- Appendix 2 contains limited MPLUS code for the 2pl model

IRT models for the tests

Inspection of the code books for Y2 and Y3 indicate that the items kept from Y2 in Y3 were identical in terms of what they asked and their distractors, making it reasonable to combine the two years.

Multiple Choice

For the initial investigation into the multiple choice knowledge items, I combined Y2 and Y3 data, focusing only on those items that Y3 students had answered. My understanding of what took place is as follows: the Y2 students were given a 46-item multiple choice knowledge test. After the completion of Y2, the items were assessed (I do not know exactly how) and 13 of the items deemed less useful were removed from the 46 item test, and 9 new items were added into the test for Y3, and the items were re-ordered. This yielded a 42 item test. In the following code, I read in data from Y2 and Y3, and create a combined dataset consisting of only those items from Y3. Students in Y2 are "missing" on the 9 new items. Because the ordering of questions changed from Y2 to Y3, the "numbering" in the reduced dataset is a little strange. I have set the items to reflect the ordering in Y3. After running an initial 2PL model, I removed item 39, because its discrimination parameter was negative, indicating that it higher-ability individuals were less likely to answer it correctly.

Table 1 contains the item parameter estimates.

Table 1: Item parameters for the Multiple Choice Test

Question	Discrimination	Difficulty
Q1	1.3644854	-1.7346388
Q2	1.4636784	-0.9822731
Q3	1.6564425	-0.4775322
Q4	0.5645806	-0.5508988
Q5	1.4767236	-0.7312600

Question	Discrimination	Difficulty
Q6	0.8590497	0.0681771
Q7	0.9433795	0.2653540
Q8	0.6946118	-0.3257144
Q9	1.2923126	-0.2368859
Q10	1.4159160	-0.7956488
Q11	0.9055303	-0.0512658
Q12	1.3775947	-0.6586147
Q13	1.4285307	-1.0143051
Q14	1.3359215	-1.4267217
Q15	2.8374411	-1.3418742
Q16	2.4193215	-1.1221477
Q17	1.8487770	-1.3441510
Q18	1.0662774	-0.5444963
Q19	1.9141691	-0.7802741
Q20	0.8583605	-0.3150340
Q21	0.4307190	1.6779977
Q22	1.1349031	-0.0656703
Q23	1.1637962	-0.0482497
Q24	2.3453199	-1.0793028
Q25	0.9178983	-0.7261372
Q26	1.2643859	-0.3073418
Q27	2.0913948	-1.1643755
Q28	2.4029780	-1.1762801
Q29	1.7397153	-0.9997344
Q30	1.3395579	-0.8697401
Q31	1.2709101	-0.1097361
Q32	1.9689396	-0.6448712
Q33	0.8411368	-0.3121080
Q34	2.0505707	-0.6540548
Q35	1.9804315	-0.5790669
Q36	0.9864397	0.3422242
Q37	1.2469835	-0.7283026
Q38	0.9830122	-1.0404786
Q40	1.5526907	-0.7153855
Q41	0.7387770	-0.7865267
Q42	2.1271418	-0.9716817

Item difficulty estimates range from -1.7346388 to 1.6779977. However, if we consider items with negative scores to be generally ‘easy’ and positive scores to be generally ‘hard’, there are more ‘easy’ items than hard items, with a mean difficulty of -0.6111958. This is reflected by the fact that higher ability scores are related to larger standard errors, plotted in Figure 1. Individuals who had larger standard errors than the apparent lines are individuals with incomplete tests. This is consistent with the item characteristic curves (Figure 2) the individual item information function curves (Figure 3) and the test information function curve (Figure 4).

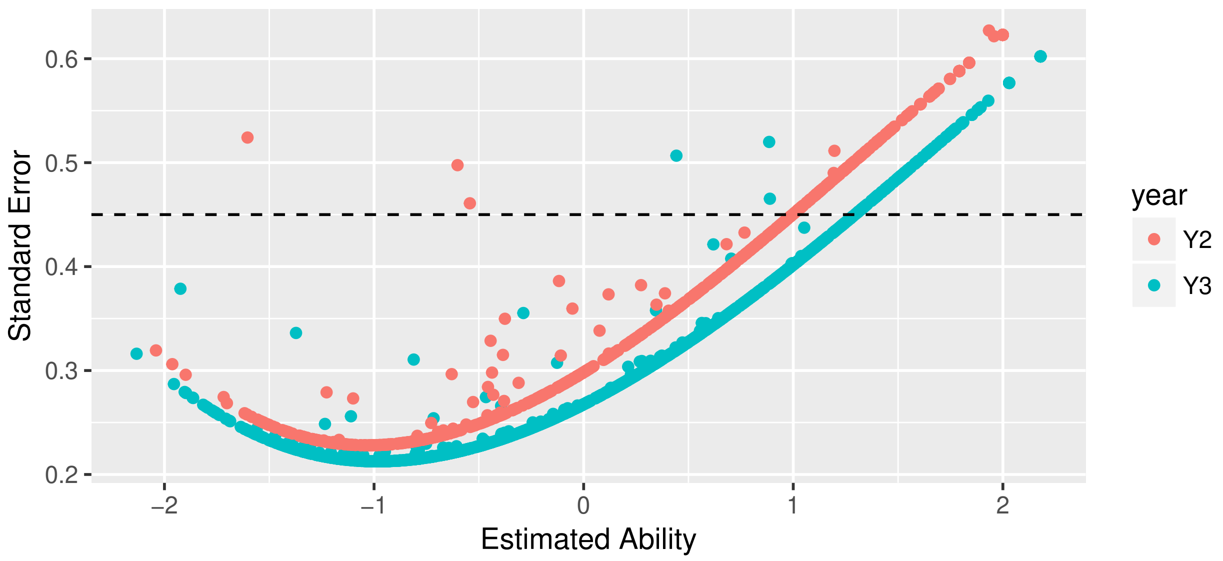

Figure 1: Estimated Standard Errors by Ability Scores

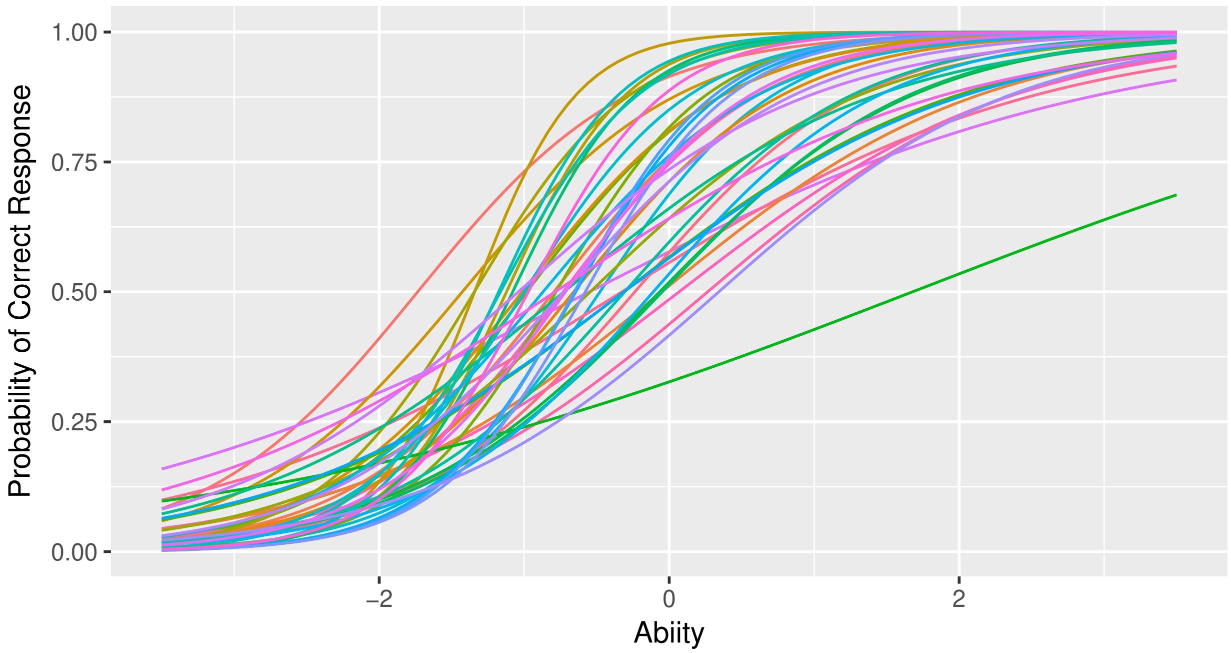

Figure 2: Item Characteristic Curves

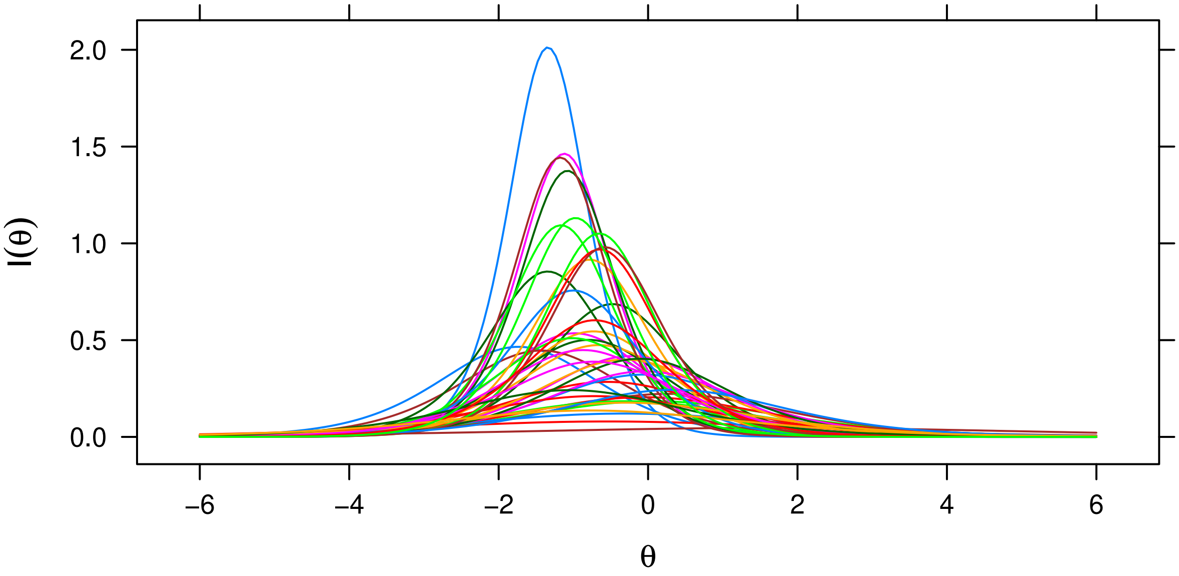

Figure 3: Item Information Curves

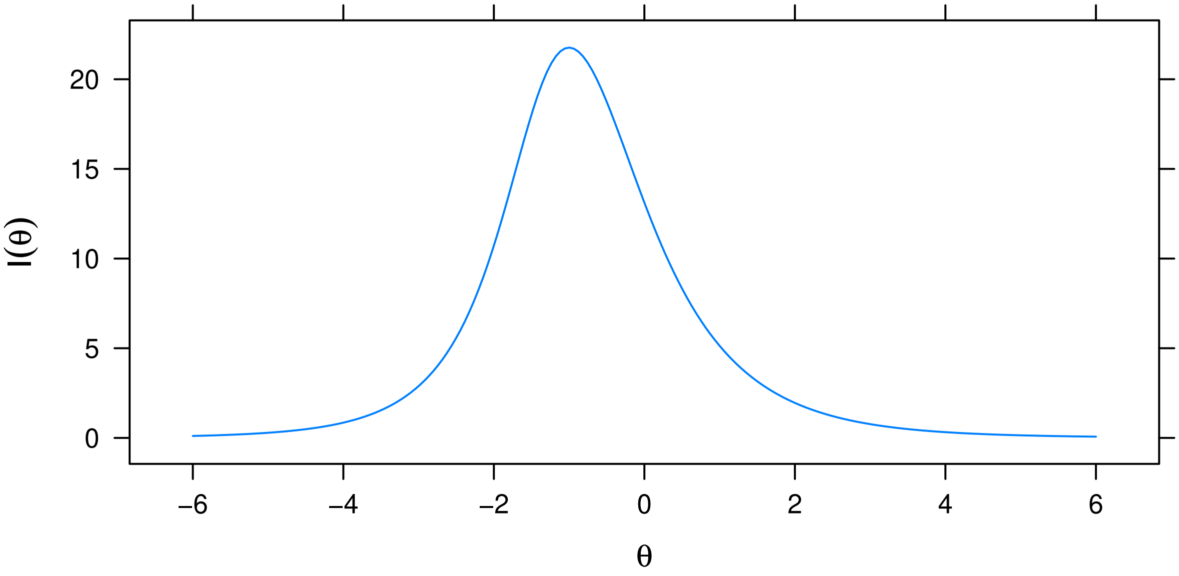

Figure 4: Test Information Curve

Reading Comprehension

The reading test was initially modeled using a unidimensional 2PL IRT model. However, these models assume local independence i.e. that the probability of answering any question on the test correctly is independent of the probability of answering any other question, controlling for the ability estimate. The actual test was composed of three reading passages, each with 7 items related to the content of these passages. It seems like a strong assumption to make that there is no dependence among items, so I decided to investigate this. I approached this first in MPLUS, however, given some difficulties with estimating appropriate multiple factor models in MPLUS, I switched to using the `mirt` package in R. The `mirt` package can estimate unidimensional models in addition to a variety of multidimensional models, including the bifactor model. First I ran a unidimensional 2PL model.

For the 2PL model, I have included the loadings/slops (a1), the intercept (d), and the difficulty (b). The factor loadings/slopes (a1) are essentially the same thing as the discrimination parameters, and the intercept (d) is negatively related to the difficulty of the item, because difficulty is calculated as $-d/a$. It is not possible to directly compare the difficulty of 2PL items to the difficulty of a bifactor model due to the compensatory nature of the bifactor model. I have provided the difficulty to here to facilitate easier comparison to classical IRT and provided the intercepts to facilitate comparison between the 2PL and bifactor models. In general terms, the larger the value of d, the easier the item is. Item parameters for the 2PL model for the reading exam are found in Table 2. Of particular note is item 60, which is far more difficult than any other item.

Table 2: Item Parameters for the 2PL model

Q	a1	d	b
Q43	1.8106	2.7940	-1.5432
Q44	1.7340	2.1723	-1.2527
Q45	1.5306	0.8837	-0.5774
Q46	0.9373	-0.4704	0.5018
Q47	1.0367	0.7126	-0.6874
Q48	0.6862	0.7546	-1.0998
Q49	1.6481	2.4963	-1.5147
Q50	1.2820	0.8865	-0.6915
Q51	1.1211	0.5342	-0.4765
Q52	1.7199	0.7884	-0.4584
Q53	1.1861	-0.6248	0.5268
Q54	0.7916	0.3474	-0.4389
Q55	1.6412	0.3111	-0.1895
Q56	1.7568	1.0734	-0.6110
Q57	1.3732	0.9037	-0.6581
Q58	0.9649	0.4005	-0.4151
Q59	1.5019	-0.0654	0.0435
Q60	0.2082	-0.9048	4.3453
Q61	0.9234	-0.4754	0.5148
Q62	1.4440	0.1813	-0.1256
Q63	1.2026	0.6239	-0.5188

I also calculated fit statistics, which uses a fit statistic called the M_2 which is similar to the χ^2 but operates on the more limited information generally available in large uni-dimensional and almost all multi-dimensional IRT models (Maydeu-Olivares & Joe, 2006). Because there was missing data in the original model, I had to use imputed datasets to calculate the fit statistics (automatically generated by the mirt package). Fit statistics for the 2PL model are contained in Table 3. The model appears to have generally good fit.

Table 3: Fit statistics for the 2PL model

	M2	df	p	RMSEA	RMSEA_5	RMSEA_95	SRMSR	TLI	CFI
stats	506.0569	189	0	0.0315	0.0282	0.0349	0.0343	0.9762	0.9786
SD_stats	7.9145	0	0	0.0004	0.0004	0.0004	0.0003	0.0006	0.0005

Next, I ran the bifactor model where each item loads on a general factor as well as a specific factor. The model fitting is not executed within this document because it takes considerable time, instead I ran it outside of this document, and the results are saved and then loaded in a future code block.

Table 4 contains the parameters for the bifactor model. Unlike unidimensional IRT, this model does not have a traditional "difficulty" and "discrimination". Instead, it has a factor loadings for each factor the item loads on as well as an intercept. As noted the `d` parameter is negatively related to difficulty, so the larger the value the easier the item.

Table 4: Item parameters from a bifactor model

	a1	a2	a3	a4	d
Q43	1.7733485	1.4903610	0.0000000	0.0000000	3.2649413
Q44	1.6441772	0.8434973	0.0000000	0.0000000	2.2663523
Q45	1.5389566	0.2531660	0.0000000	0.0000000	0.8918714
Q46	0.9801071	-0.0496262	0.0000000	0.0000000	-0.4686964
Q47	1.0259593	0.6133822	0.0000000	0.0000000	0.7533180
Q48	0.6545944	0.7817783	0.0000000	0.0000000	0.8356215
Q49	1.6428218	1.2477913	0.0000000	0.0000000	2.8366286
Q50	1.3002704	0.0000000	0.1580554	0.0000000	0.9268074
Q51	1.1494396	0.0000000	0.2609556	0.0000000	0.5625525
Q52	1.8372625	0.0000000	0.3642124	0.0000000	0.8431653
Q53	1.3862145	0.0000000	1.1509404	0.0000000	-0.7396061
Q54	0.7855982	0.0000000	0.0860954	0.0000000	0.3547445
Q55	1.6989325	0.0000000	0.3568363	0.0000000	0.3492907
Q56	1.8853601	0.0000000	-0.0997592	0.0000000	1.1296617
Q57	1.4325599	0.0000000	0.0000000	0.3263401	0.9423512
Q58	0.9617889	0.0000000	0.0000000	0.0906393	0.4095952
Q59	2.7844212	0.0000000	0.0000000	2.7341028	-0.0718057
Q60	0.1593573	0.0000000	0.0000000	0.1527658	-0.9100719
Q61	0.9490576	0.0000000	0.0000000	0.1350286	-0.4831056
Q62	1.4497871	0.0000000	0.0000000	0.2684613	0.1976931
Q63	1.2063730	0.0000000	0.0000000	0.1690297	0.6449553

Table 5 contains the fit statistics for the model, which indicate good fit that appears to be better than the 2PL model.

Table 5: Fit statistics for the bifactor model

	M2	df	p	RMSEA	RMSEA_5	RMSEA_95	SRMSR	TLI	CFI
stats	268.7083	168	0	0.0191	0.0147	0.0233	0.0263	0.9913	0.9931
SD_stats	6.3004	0	0	0.0006	0.0007	0.0006	0.0003	0.0005	0.0004

According to Cai & Hansen (2013), the 2PL model is nested in the bifactor model, and so it is possible to perform a χ^2 difference test using the M_2. The critical value for a χ^2 with df = 21 is 32.6705733, which is quite a bit smaller than the calculated difference of 235.4276, which suggests the bifactor model is a better fit for the data.

Table 6: Comparison of the 2PL and bifactor reading models.

Question	a_2pl	a_bif	d_2pl	d_bif
Q43	1.8105731	1.7733485	2.7940208	3.2649413
Q44	1.7340263	1.6441772	2.1722963	2.2663523
Q45	1.5305963	1.5389566	0.8837453	0.8918714
Q46	0.9372999	0.9801071	-0.4703797	-0.4686964
Q47	1.0366794	1.0259593	0.7126387	0.7533180
Q48	0.6861545	0.6545944	0.7546056	0.8356215
Q49	1.6480687	1.6428218	2.4963296	2.8366286
Q50	1.2820060	1.3002704	0.8865381	0.9268074
Q51	1.1211175	1.1494396	0.5342441	0.5625525
Q52	1.7198977	1.8372625	0.7883813	0.8431653
Q53	1.1860532	1.3862145	-0.6248427	-0.7396061
Q54	0.7916092	0.7855982	0.3474220	0.3547445
Q55	1.6411970	1.6989325	0.3110887	0.3492907
Q56	1.7567807	1.8853601	1.0733717	1.1296617
Q57	1.3732250	1.4325599	0.9037434	0.9423512
Q58	0.9649026	0.9617889	0.4005316	0.4095952
Q59	1.5018526	2.7844212	-0.0653651	-0.0718057
Q60	0.2082206	0.1593573	-0.9047914	-0.9100719
Q61	0.9234360	0.9490576	-0.4753945	-0.4831056
Q62	1.4439993	1.4497871	0.1813411	0.1976931
Q63	1.2025764	1.2063730	0.6239250	0.6449553

Table 6 contains the discrimination and the intercepts of the 2pl model and the specific factor discrimination and intercept of the bifactor model. Generally speaking, the consequence of modeling the test using a bifactor model is that the items are less difficult, although the differences are not large. The violation of local independence may not be of much concern here. Both models fit quite well, and exploration of differential item functioning is difficult in the context of a bifactor model, so examination of DIF will focus on the 2PL version of the model.

DIF

Differential item function (DIF) analysis can be useful for assessing the fairness of an exam. It can help tell if some items perform differently based on important demographic differences in the sample (e.g. SES, race, or primary language spoken). The focus of this analysis was for items whose *difficulty* was different, not those items where discrimination was different. Ultimately, while discrimination differences would effect the uncertainty around an ability estimate, differences in discrimination will not systematically bias estimates of ability.

One of the crucial aspects for assessing DIF is the anchor selection strategy. Because θ (ability) is latent, you have to have a set of low-DIF anchor items to set the scale. Imagine you are assessing DIF for low-income students first middle- and high-income students. If you use a set of items that are harder for low-income students of the same ability as middle- and high-income students, you will incorrectly estimate ability levels for the low-income students that are lower than their true ability. One strategy for setting the anchors involves making use of expert knowledge to identify which items are believed to be low in DIF, and use those anchor items to estimate DIF.

There are also a set of what I'll call "low information" methods for selecting anchors. These strategies employ no prior knowledge about which items have DIF, but instead make use of the information in the sample to select anchor items. Kopf, Zeileis & Strobl (2014) examined a number of existing anchor selection strategies in addition to several new anchor selection strategies. They divided the two kinds of anchor strategies into *constant anchor* strategies and *iterative forward class* strategies. In the constant anchor strategy a set of number of anchors, in this case four, are selected and then DIF is assessed on the remaining items in the exam. In the iterative forward class strategy, you begin with the anchor that exhibits the lowest DIF based on your selection strategy, and then assess DIF on all items. If there are more items that do not exhibit DIF than are currently in your anchor pool, you add the next-lowest DIF item based on your selection strategy to the anchor pool, and assess DIF again. You continue this iterative process until the number of items outside your anchor pool that do not exhibit DIF is equal to or less than the number of items within your anchor pool.

The two strategies that performed the best were the Mean Test Statistic Threshold Selection (MTT) selection method combined with the iterative forward class strategy, and the Mean p-value Threshold (MPT) selection method combined with the 4-anchor strategy. To understand the MTT method, imagine that there are j items in your pool. The first step in this method involves performing j DIF analyses, each with a single item as the anchor. This will provide $j-1$ DIF test statistics for each item (because each item has served as an anchor one time). For each item, you calculate a mean test statistic from the collection of $j-1$ statistics, and then sort them in order. You take the 0.5 * j ranked mean, rounding up if there are an odd number of items, and that mean serves as a threshold. For each item, you then calculate the number of $j-1$ test statistics that exceed the threshold. Rank each item by the number of $j-1$ test statistics that exceed the threshold in ascending order to determine which have the lowest to highest DIF. You can then proceed to iteratively add items until you have the the full pool.

The MPT method works in essentially the same way, except instead of test statistics you collect a set of $j-1$ p-values. The 0.5 * j ranked mean p-value serves as the threshold, and in this case you rank each item by the number of values that are lower than the threshold in ascending order. Select the four items with the fewest number of $j-1$ p-values that are lower then the threshold to serve as your anchors.

As a practical aside, the there were sometimes ties. When there were ties, I broke them by the mean p-value or test statistic as appropriate.

Multiple Choice

In the case of the multiple-choice Social Studies exam, this means there are 40 test statistics and p-values to collect for each type of low-information DIF testing.

Free or Reduced Lunch

Free or reduced lunch status sometimes used as a proxy for family income. Students that did not receive free or reduced lunch are coded as 0 on this variable, and students who received free or reduced lunch are coded 1.

4-anchor MPT

Table 7: MPT anchors for FRL

Q	p_count
Q8	15
Q10	16
Q17	16
Q20	16

The 4 anchor items selected are questions 8, 10, 17, and 20, displayed in Table 7. I estimated a multiple group model with those four questions constrained equally across groups.

Table 8: MPT DIF results for FRL

Q	p2
Q3	0.0167346
Q13	0.0006483
Q14	0.0147340
Q25	0.0384501
Q38	0.0075131

DIF testing that uses the four items as anchors find that questions 3, 13, 14, 25, and 38 all may have DIF (the p2 values are the p-values associated with the DIF test for those items). The p-values for those questions are displayed in Table 8.

Table 9: Item parameters for FRL status using MPT anchor selection

Q	a_noFRL	a_FRL	b_noFRL	b_FRL
Q3	1.6334701	2.172717	-0.7856966	-0.844281
Q13	1.2879383	2.082733	-1.3541391	-1.255324
Q14	1.4287736	1.253445	-1.8539782	-1.570872
Q25	0.8525667	1.237058	-1.0470547	-1.007169
Q38	1.0133107	1.330950	-1.2190074	-1.311702

Table 9 lists the item parameters for the two groups. Items 3 and 38 seem to be easier for students who receive free or reduced lunch, while items 13 and 14 seem harder for students who receive free or reduced lunch. The DIF on 25, while statistically significant, doesn't seem all that meaningful.

Figure 5 displays the ICCs of the items with DIF based on FRL, using 4-anchor MPT.

One additional option in the `mirt` package is to perform a "differential test functioning" diagnostic. In a given exam, if some items favor one group and some items favor another group, it is conceivable that the performance of the test as a whole may be balanced out.

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##    0.02093832    0.05106907    0.11933185    0.29105329
##
## $CIs
##          sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5  0.06688788    0.16314118 0.18053107     0.4403197
## CI_2.5  -0.02400283   -0.05854349 0.05701308     0.1390563
##
## $tests
## P(sDTF.score = 0)
##         0.3618736
```

The DTF results suggest that there's an approximate 0.05% difference in scores by sDTF and a 0.29% difference in scores by uDTF between the two groups. They are slightly different DTF calculations, and although the values somewhat, the suggested difference between the two FRL groups is not large. Also, the overall significance tests suggests there's not significant difference in overall test performance.

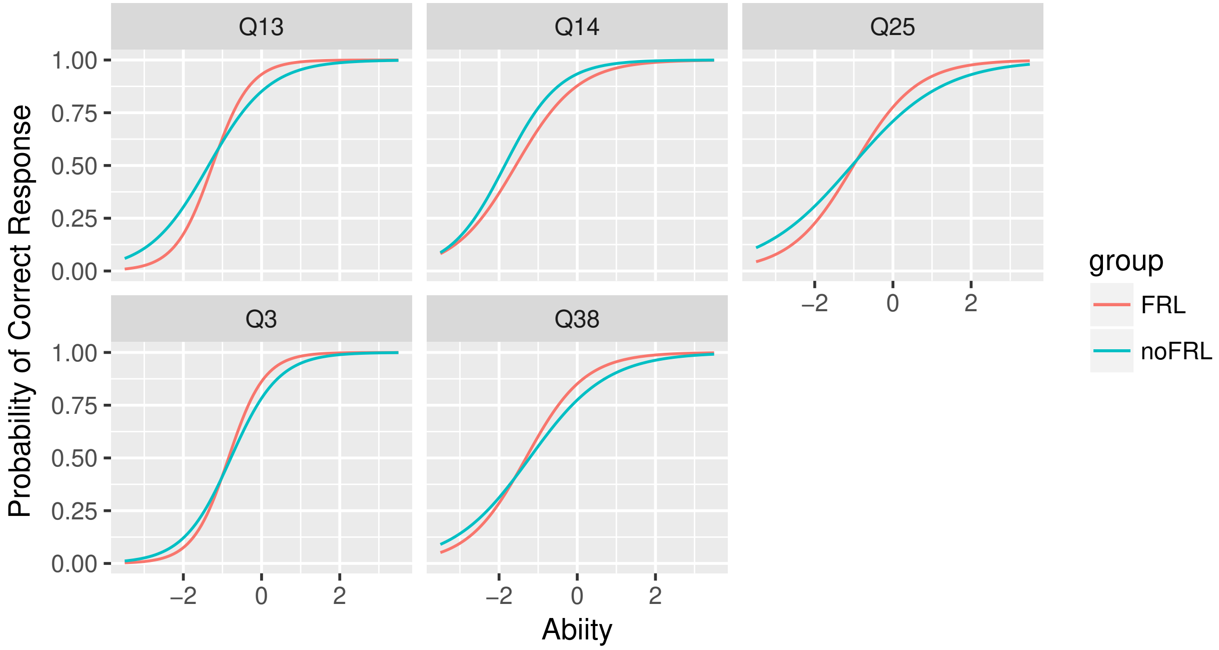

Figure 5: ICCs of items with DIF based on FRL using 4-anchor MPT

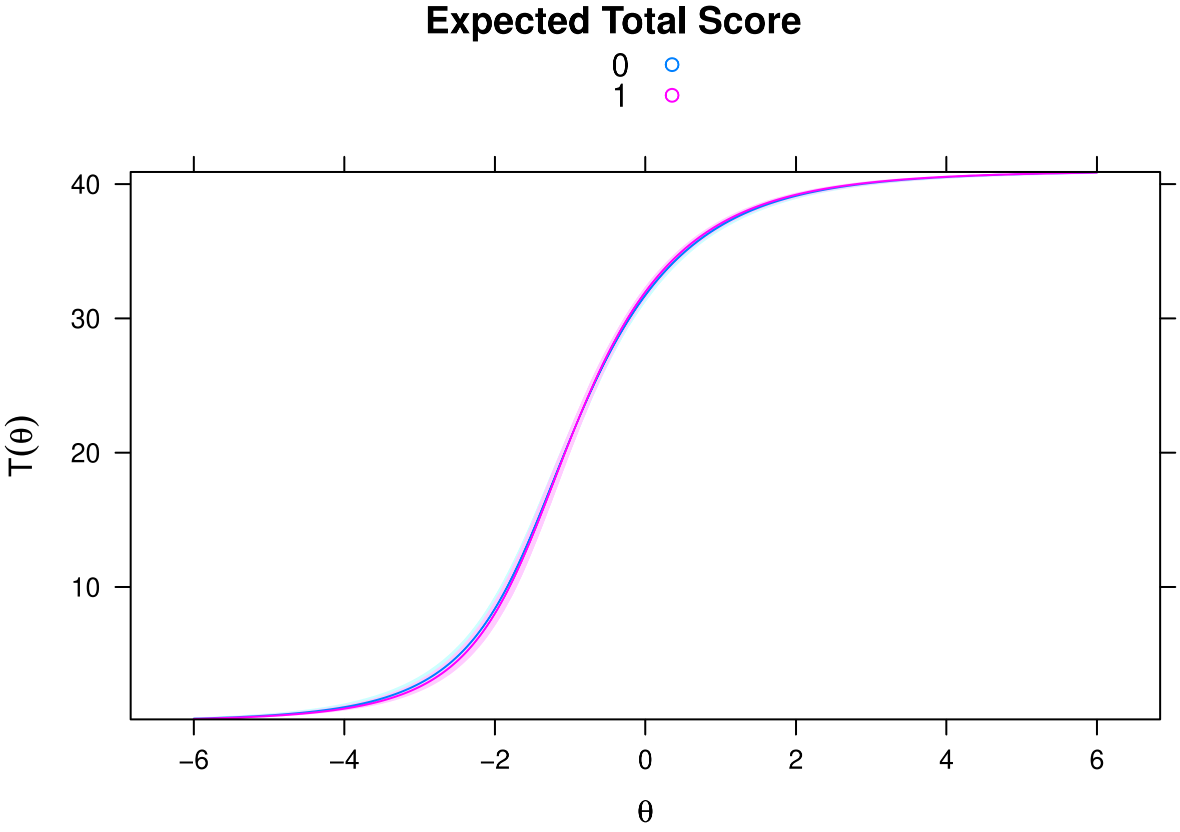

Figure 6: Expected total score for a given ability by FRL status via 4-MPT

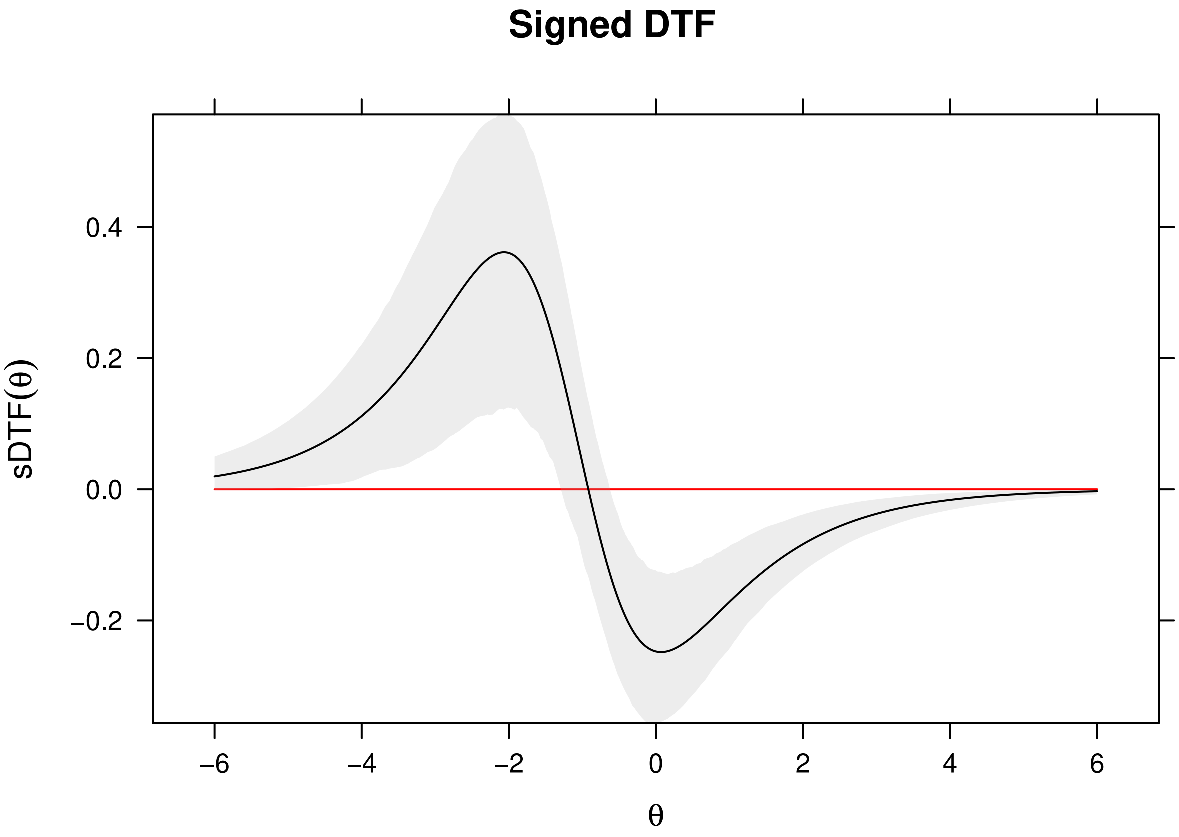

Figure 7: Differential test functioning at different levels of ability by FRL status via 4-MPT

Figure 6 displays the expected number correct for FRL = 0 (no free or reduced lunch) and for FRL = 1 (free or reduced lunch) across different levels of ability. This suggests that using the total scores could be an appropriate way to score this test.

Figure 7 displays the level of differential test functioning across different levels of ability. The value of sDTF is the roughly the difference between the baseline (no free lunch) and the focal group (free lunch). So a value of 0.4 at -2 suggests that for two students who both have an ability of -2 and whose only difference is that one receives free or reduced lunch one does not, that the student who does not receive free or reduced lunch would have a score 0.4 points higher if we assumed no differential test functioning in our scoring process. Ability estimates are likely to be a biased way to score this test.

Iterative Forward Anchor MTT

Given that the number of anchors is variable, I won't displayed the anchor information.

Table 10: MTT DIF results for FRL

Q	p
Q3	0.0173660
Q9	0.0258205
Q13	0.0006609
Q14	0.0024374
Q36	0.0204742

Q	p
Q38	0.0096738

Table 10 lists the results of the MTT DIF testing for FRL. This method suggests that items 3, 9, 13, 14, 36, and 38 have DIF. There is converging evidence for 3, 13, 14, and 38 having DIF.

Table 11: Item parameters for MTT DIF items.

Q	a_noFRL	a_FRL	b_noFRL	b_FRL
Q3	1.631231	2.1288660	-0.7863747	-0.8334896
Q9	1.242500	1.3224343	-0.6902808	-0.3945748
Q13	1.287050	2.0419520	-1.3552336	-1.2528199
Q14	1.424687	1.2297621	-1.8579901	-1.5738175
Q36	1.063023	0.8249904	-0.0685969	0.2878716
Q38	1.013686	1.3055537	-1.2192014	-1.3096530

Table 11 contains the item parameters for students with and without Free or Reduced lunch, for the items with DIF. Consistent with the results from the other method, questions 3 and 38 are easier for students who receive free or reduced lunch. Questions 9, 13, 14 and 36 were all harder for students who received free or reduced lunch.

Figure 8 displays the ICCs for items with DIF by FRL status using MTT anchors.

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##    0.07412844    0.18080108    0.07467652    0.18213786
##
## $CIs
##         sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5 0.13463526    0.32837868 0.13782589     0.3361607
## CI_2.5  0.02235903    0.05453422 0.04221367     0.1029602
##
## $tests
## P(sDTF.score = 0)
##       0.009538921
```

The DTF results suggest that there's an approximate 0.18% difference in scores between the two groups. Maybe not that big of a deal, but the significance test for DTF suggests that there is DTF somewhere.

Figure 9 displays the expected # correctly for FRL = 0 (no free or reduced lunch) and for FRL = 1 (free or reduced lunch) across different levels of ability. The two groups are not very different, consistent with previous results suggesting total item scoring might be appropriate.

Figure 10 displays the level of differential test functioning across different levels of ability. It appears as though there may be a small amount of differential test functioning around $\theta = -2$ (e.g. that students who do not receive free or reduced lunch will find the test easier), but the rest of the time the scores may not be affected. This is a little more comprehensible, and potentially troubling, and may once again be evidence of issues around question #13. Using estimated ability to score the exam would probably result in biases.

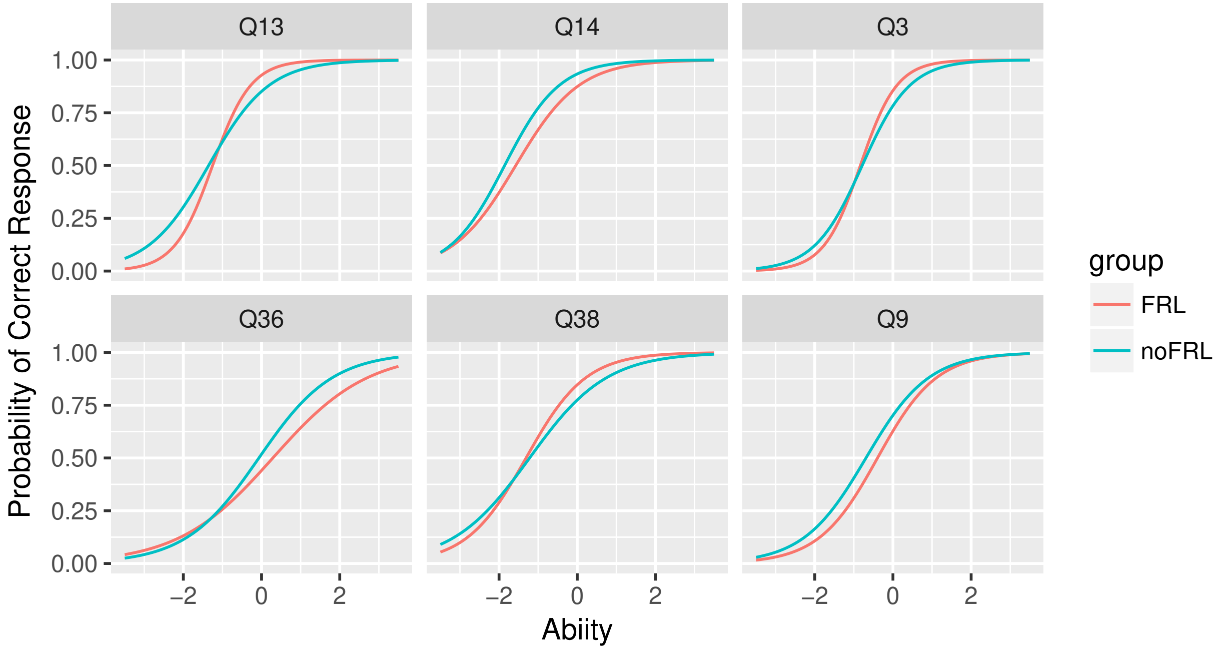

Figure 8: ICCs of items with DIF based on receipt of FRL using iterative MTT

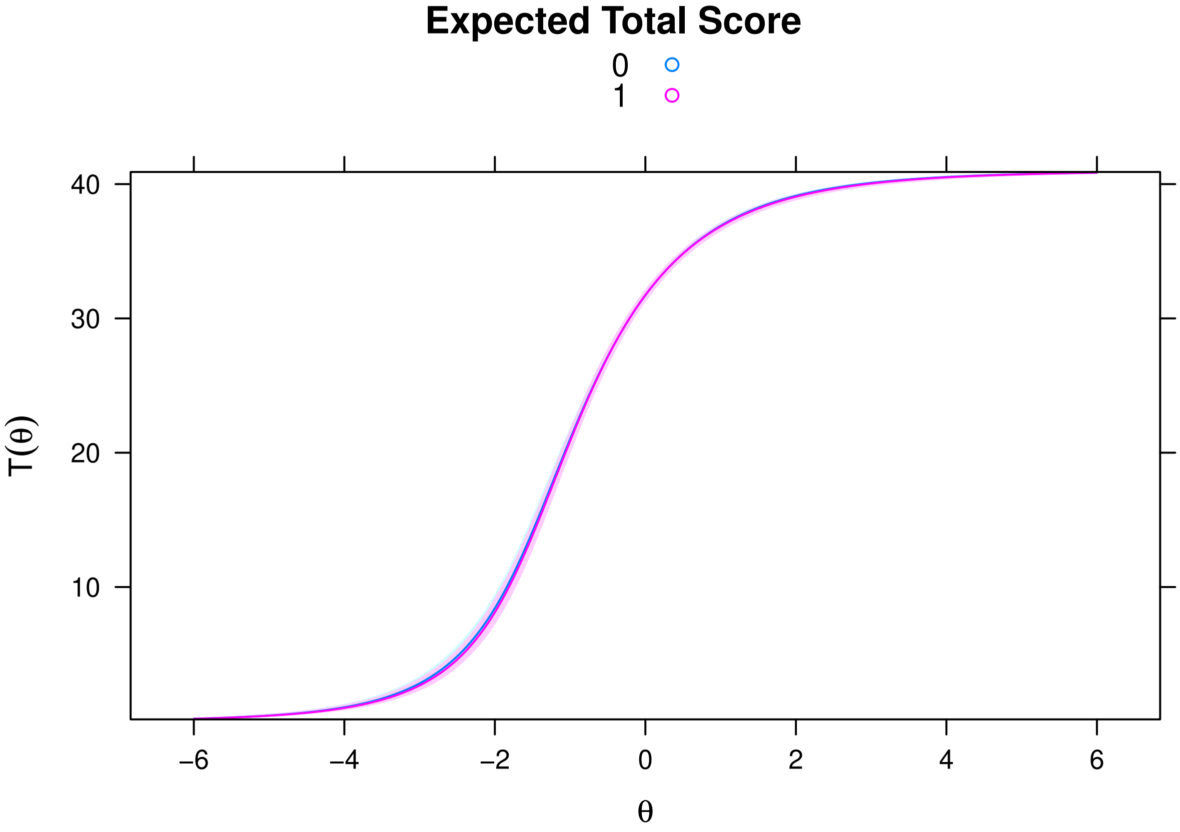

Figure 9: Expected total score for a given ability by FRL status via i-MTT

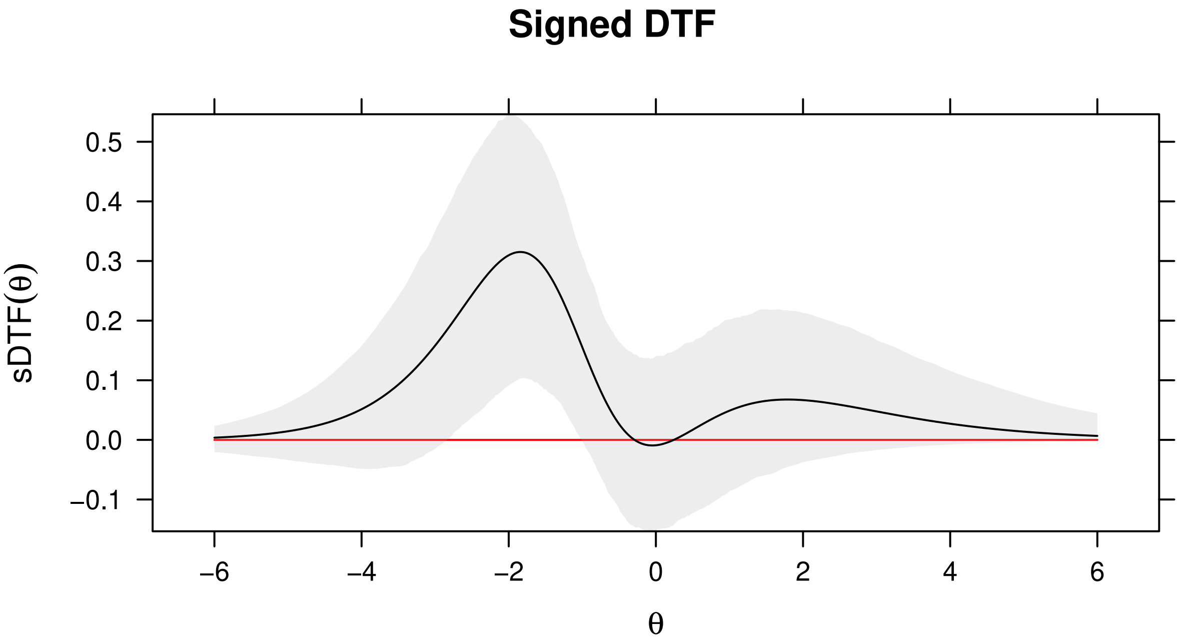

Figure 10: Differential test functioning at different levels of ability by FRL status via i-MTT

Expert anchors

I also recruited experts to identify anchors that they thought were most “fair.” They identified items they thought were most aligned with the curriculum, and also that they thought would not be too difficult. Combining their feedback, I selected items 2, 4, 7, 15, 22, 29, and 31 as anchor items. Items 9 and 14 were also identified as potential anchors, but both had been identified as an item with DIF by our two other methods, there were plenty of anchors without their inclusion, and their inclusion ended up providing results that were difficult to interpret and extremely inconsistent with the other results, so the analysis presented does not include them as anchors.

Table 12: Expert anchor DIF for FRL

Q	p
Q36	0.0232062
Q13	0.0011957
Q14	0.0036244
Q3	0.0319720
Q38	0.0141286

Table 12 contains the results of DIF testing using expert anchors. Questions 3, 13, 14, 36, and 38 all may have DIF.

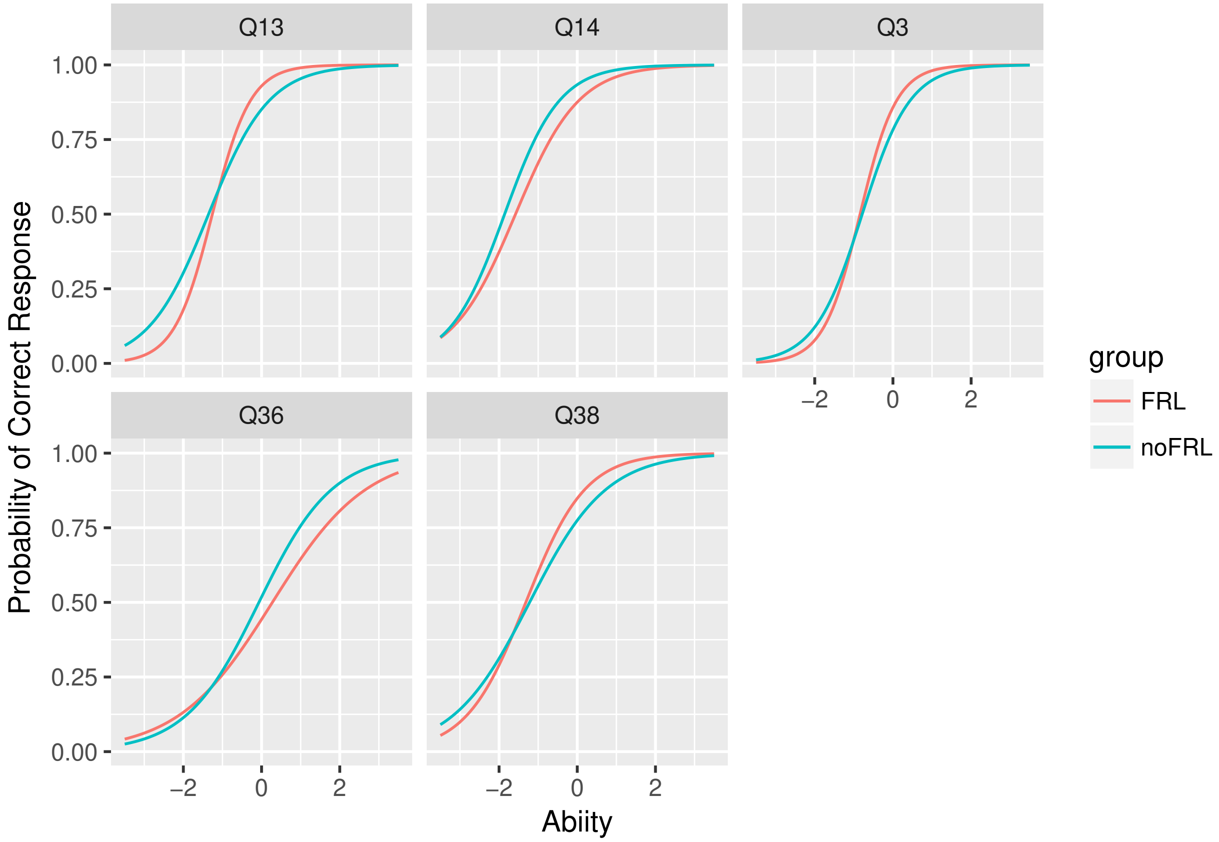

Figure 11: ICCs of items with DIF based on receipt of FRL using expert anchors.

Table 13: Item parameters for FRL status using expert anchor selection

Q	a_noFRL	a_FRL	b_noFRL	b_FRL
Q3	1.630467	2.135508	-0.7859933	-0.8408537
Q13	1.286517	2.048561	-1.3550983	-1.2588723
Q14	1.424645	1.233586	-1.8576479	-1.5789992
Q36	1.062377	0.827972	-0.0680363	0.2763822
Q38	1.012636	1.309846	-1.2195645	-1.3155477

Table 13 contains item parameters for students with and without Free or Reduced Lunch using the expert anchors. Similar to before, questions 3 and 38 are easier for students who receive free or reduced lunch, questions 13 an 14 are harder for students who received free or reduced lunch, and question 25 has some DIF but not terribly meaningful DIF.

Figure 11 displays the ICCs of items potentially having DIF.

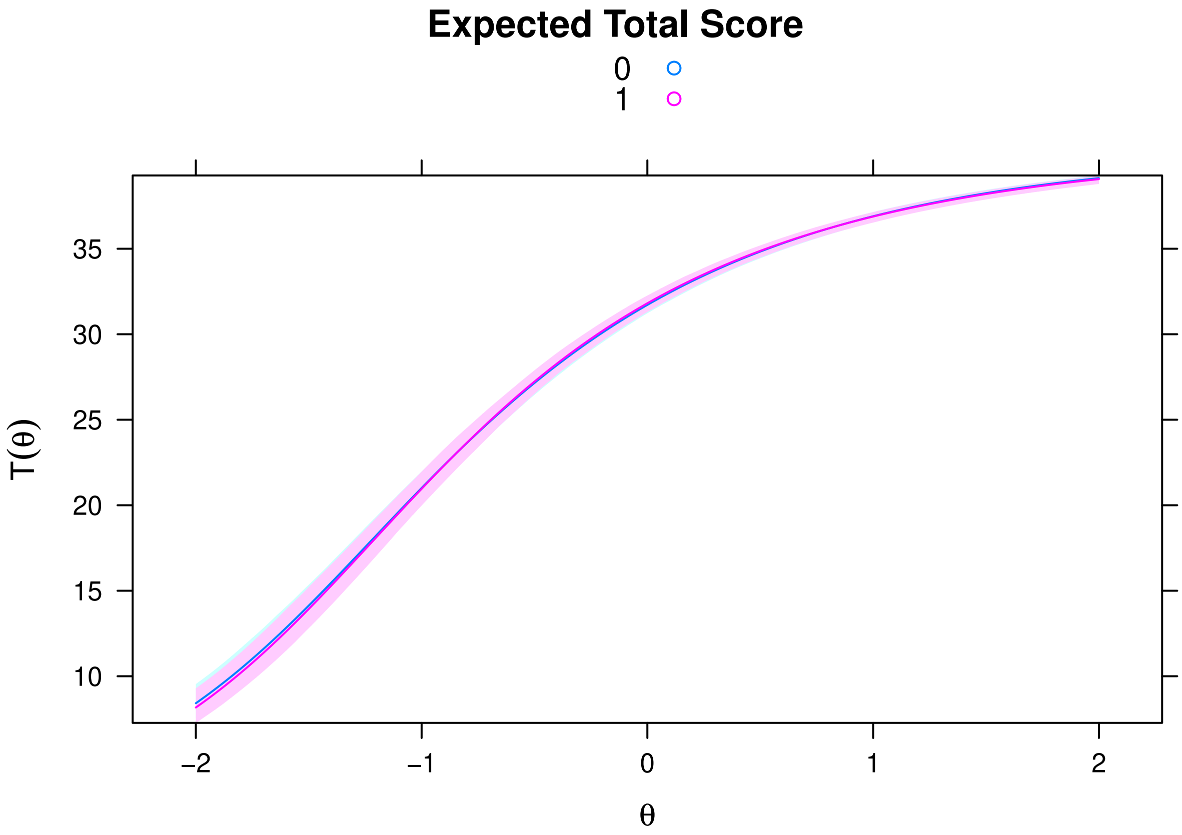

Figure 12: Expected # of items by ability, expert anchors

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##    0.04648711    0.11338320    0.06379419    0.15559558
##
## $CIs
##           sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5  0.100958586   0.246240454 0.12157864    0.29653326
## CI_2.5  -0.001047199  -0.002554145 0.03331908    0.08126604
##
## $tests
## P(sDTF.score = 0)
##        0.07853924
```

DTF test is not significant.

Figure 12 suggests that the expected number of correct items stays about the same across ability for the two groups. Consistent with the other anchor selection methods, the total item score appears to be an approximately unbiased way to estimate ability.

Figure 13 displays signed DTF for the expert anchor selection.There is some suggested DTF here as well, with students not receiving free or reduced lunch doing worse around an ability of -2. Once again the influence of question #14 is a problem.

Overall, it does appear as though there may be some meaningful concerns about DIF (and DTF) for the exam when comparing students who receive free or reduced lunch and those who don't. Items 3, 13, 14 and

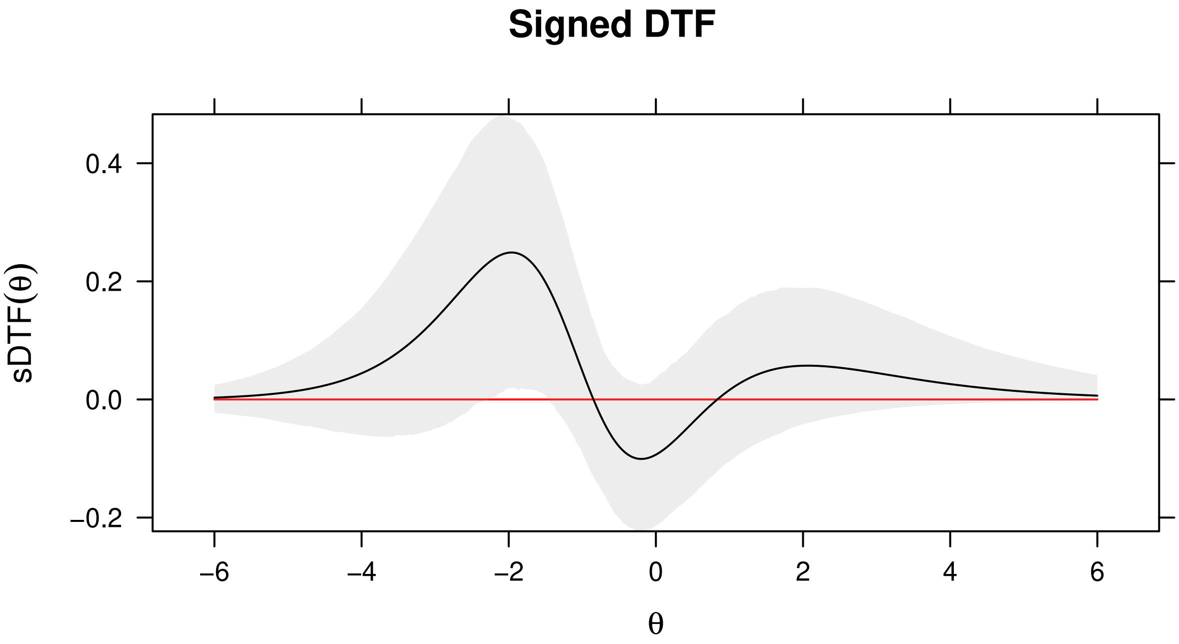

Figure 13: Expected ability difference expert anchors

38 appeared to have DIF across all versions of anchor selection and number 36 appeared in at least two.

Overall, the use of the total number of items correct seems like the best way to score this exam when accounting for potential DIF due to family income, insofar as FRL status is a useful proxy for that.

Limited English Proficency.

Similar to the exploration of DIF based on FRL status, those students who are not indicated as having limited English proficiency are LEP = 0, and those who are indicated as having any sort of limited English proficiency are LEP = 1. One considerable limitation of this comparison is that there are only 88 students with limited English proficiency and 1900+ without.

4-anchor MPT

Table 14: MPT anchors for LEP

Q	p_count
Q1	15
Q20	9
Q21	0
Q36	9

Table 14 contains the MPT anchors for LEP status. In this case, items 1, 20, 21, and 36 were selected as anchors.

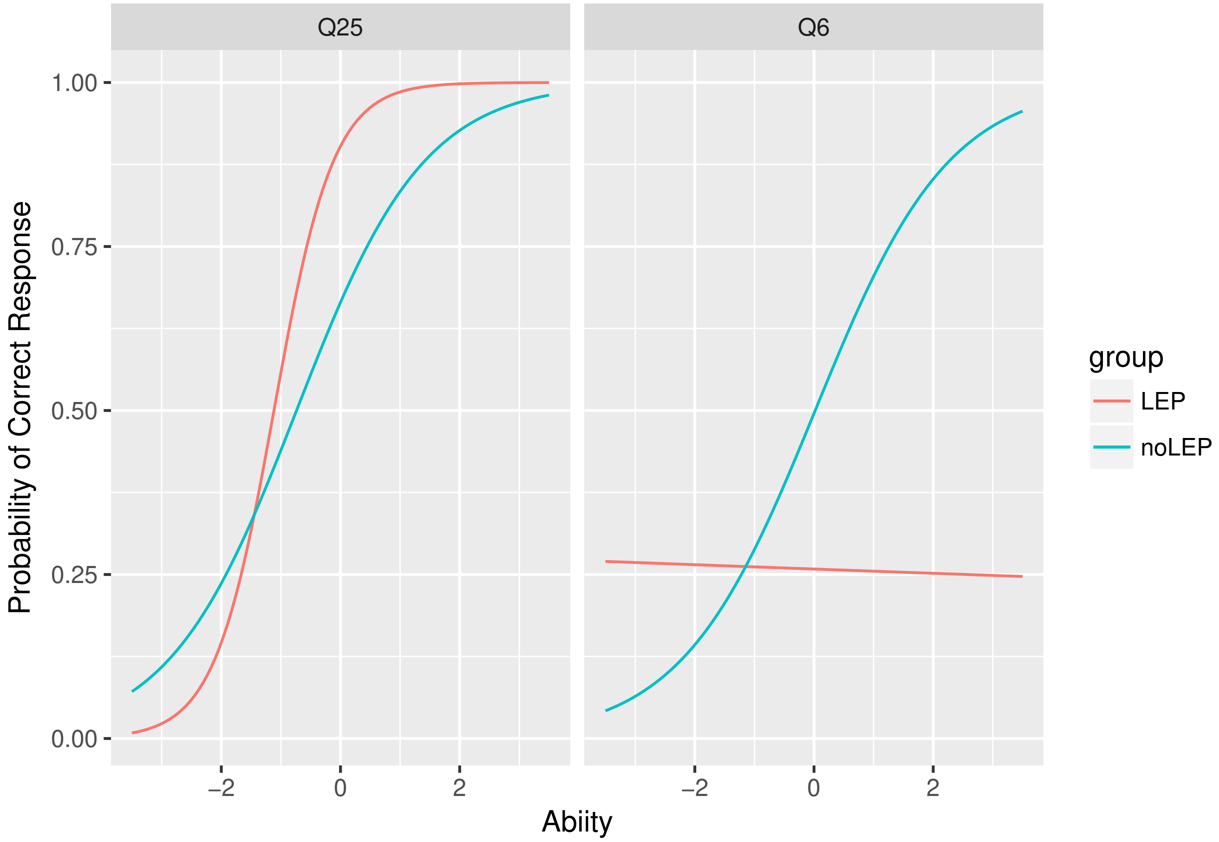

Figure 14: ICCs of items with DIF based on LEP status using 4-anchor MPT

Table 15: DIF items for LEP using MPT anchors

Q	p2
Q6	0.0365555
Q25	0.0320466

Table 15 contains the items that may have DIF for by LEP status using the MPT anchors. Items 6 and 25 may have DIF.

Table 16: Item parameters for LEP and non-LEP students with MPT anchors

Q	a_noLEP	a_LEP	b_noLEP	b_LEP
Q6	0.8870660	-0.0171424	0.0199205	-61.521575
Q25	0.9283743	1.9982869	-0.7375237	-1.116243

Table 16 contains the item parameters for items with potential DIF. Item 6 doesn't appear to discriminate for students with a LEP designation, and item 25 appears easier for students with LEP status.

Figure 14 displays the ICCs of the items with DIF.

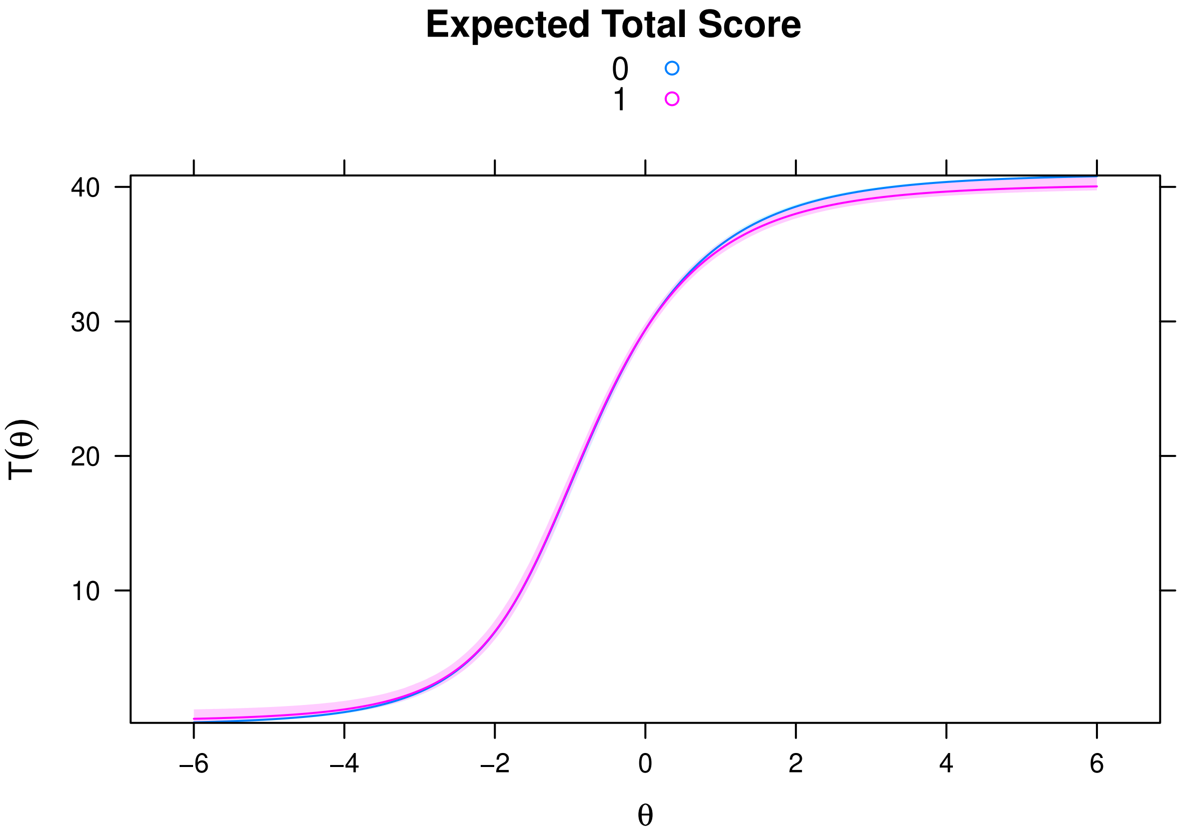

Figure 15: Expected total score for a given ability by LEP status via 4-MPT

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##     0.2078301     0.5069026     0.3483106     0.8495380
##
## $CIs
##          sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5  0.27176256    0.66283551  0.6711412     1.6369298
## CI_2.5  -0.02848968   -0.06948704  0.0505550     0.1233049
##
## $tests
## P(sDTF.score = 0)
##       0.009544526
```

Overall significance test indicate there's probably some DTF.

Figure 15 displays the expected # correctly by LEP status across ability levels. There is a lot of overlap in the curves, except near the top end where students who do not have LEP pull away slightly from students with LEP. This is probably related to the negatively-discriminating item.

Figure 16 displays signed DTF for the 4-MPT method. It appears as though there may be considerable DTF based on English proficiency.The same issue with the negatively discriminating item may be at fault here.

As with FRL, using the total items correct seems like a better way to score this exam.

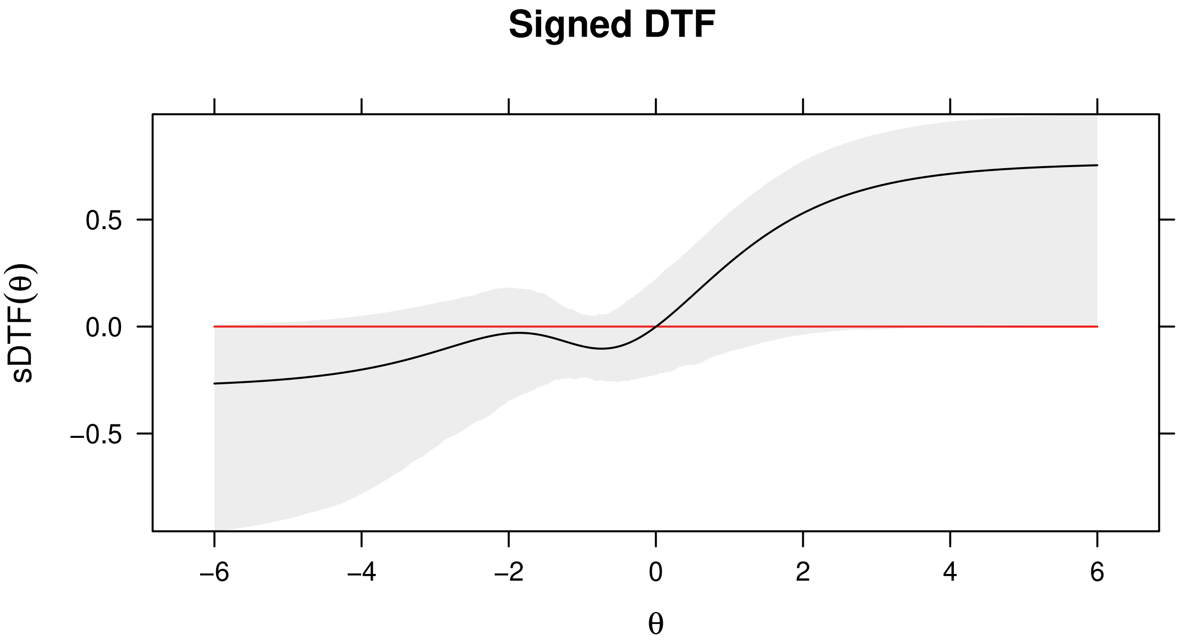

Figure 16: Differential test functioning at different levels of ability by LEP status via 4-MPT

Iterative Forward Anchor MTT

Table 17: Items with DIF for LEP status using MTT anchor selection.

Q	p
Q6	0.0092547
Q8	0.0332085
Q25	0.0150125
Q27	0.0385860
Q34	0.0337761

Table 17 contains the results of DIF testing using the MTT anchor selection method. This method suggests that items 6, 8, 25, 27, and 34 have DIF.

Table 18: Item parameters for items with DIF based on LEP status using MTT anchors.

Q	a_noLEP	a_LEP	b_noLEP	b_LEP
Q6	0.8868731	-0.0141776	0.0201283	-74.200653
Q8	0.6897522	-0.0406541	-0.3919683	-13.490579
Q25	0.9286886	1.9460545	-0.7371366	-1.133022
Q27	2.1369624	3.1762736	-1.1738288	-1.554541
Q34	2.0785676	1.4597612	-0.7123920	-0.412867

Table 18 contains item parameters for the items with DIF for students classified and not classified as LEP. This suggests 6 and 8 have no or negatively discriminating power for students with limited English proficiency, which makes the estimates of difficulty irrelevant. Otherwise items 25 and 27 are easier for students with LEP and item 34 is harder.

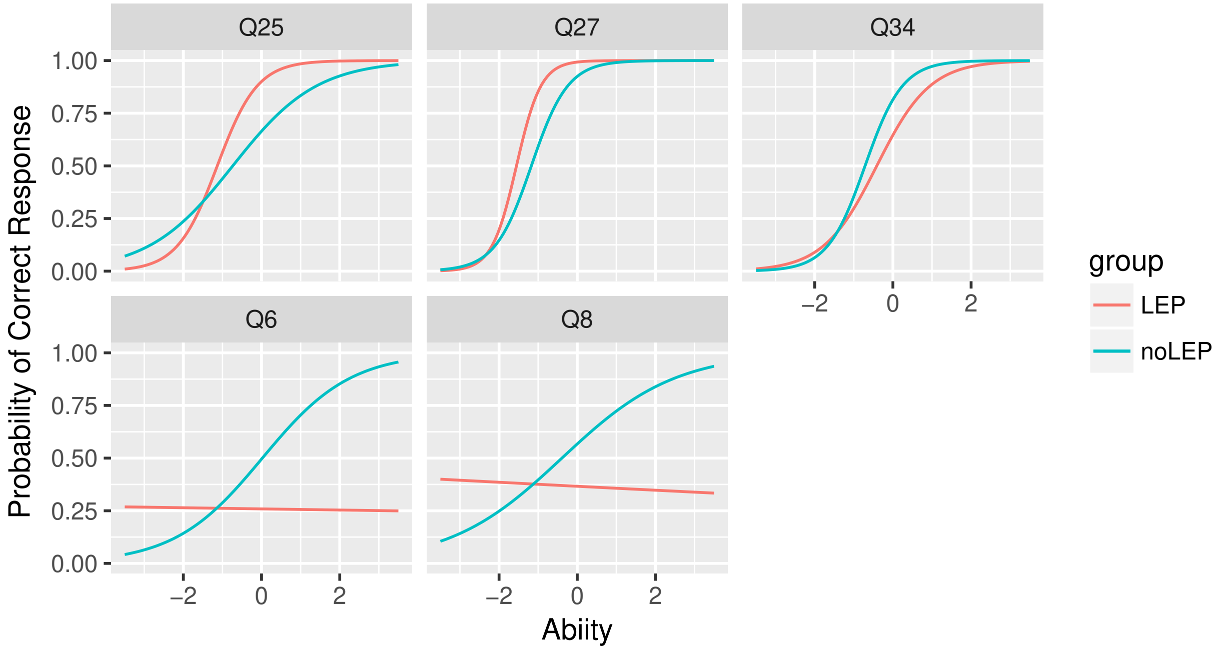

Figure 17: ICCs of items with DIF based on LEP status using iterative MTT

Figure 17 contains the ICCs for the items that may have DIF.

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##     0.3639431     0.8876660     0.7585206     1.8500502
##
## $CIs
##          sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5 0.46974605    1.14572207  1.2693506     3.0959771
## CI_2.5  0.04034426    0.09840064  0.2022943     0.4934007
##
## $tests
## P(sDTF.score = 0)
##       0.001091213
```

The test statistic suggests that there is DTF.

In Figure 18, the overlap between the two groups for their total score curve based on estimated theta is more limited. At low and high ends of the ability scale, the two curves diverge.

Figure 19 somewhat matches the DIF results for the MPT method. The two negatively-discriminating items may help explain the weird behavior here.

As before, the number of items correct is probably better than using estimated ability to score the test, but there may be some limitations.

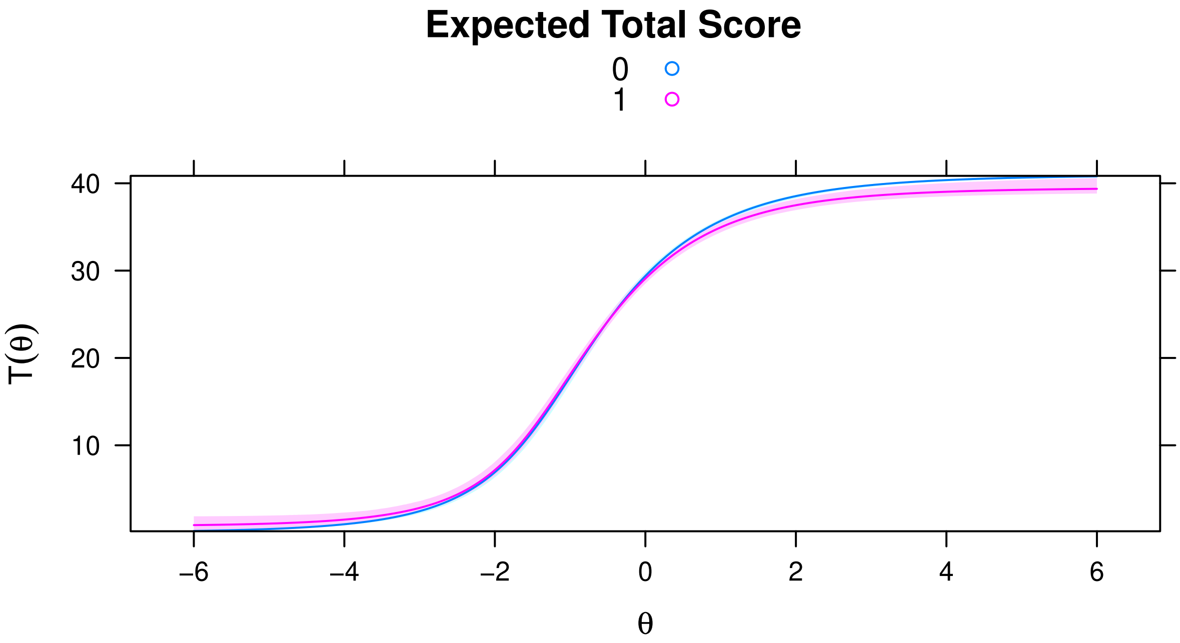

Figure 18: Expected Total Score by estimated ability for LEP using i-MTT

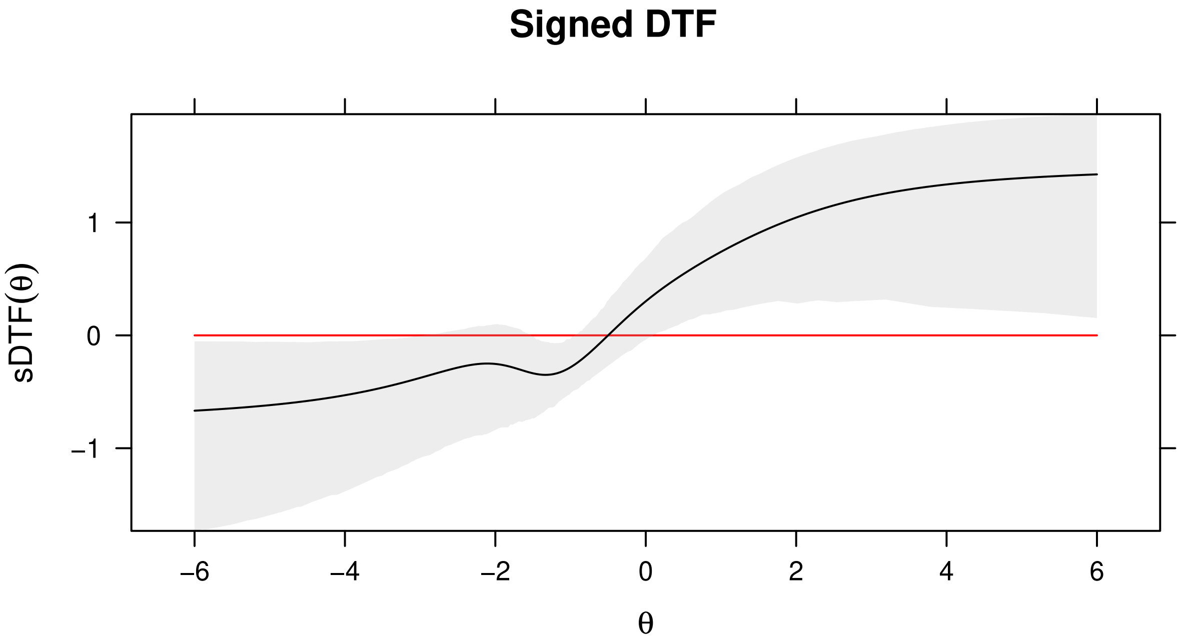

Figure 19: Signed DTF by estimated ability for LEP using i-MTT

Expert anchors

The possible pool of expert anchors is 2, 4, 7, 9, 14, 15, 22, 29, and 31.

Table 19: DIF results for LEP using expert anchors

Q	p
Q10	0.0332085
Q25	0.0022347
Q27	0.0097555
Q37	0.0369872
Q38	0.0211912

Table 19 contains the results of the DIF testing using expert anchors. Questions 10, 25, 27, 37, and 38 all my have DIF. These anchors didn't seem to catch any of the negatively discriminating items but is not totally inconsistent with previous results.

Table 20: Item parameters for items with DIF by LEP status using expert anchors

Q	a_noLEP	a_LEP	b_noLEP	b_LEP
Q10	1.4105551	2.173595	-0.8336767	-1.115519
Q25	0.9285274	2.166955	-0.7372872	-1.143031
Q27	2.1373978	3.432854	-1.1738165	-1.532695
Q37	1.2238979	2.513199	-0.7788889	-0.961090
Q38	0.9563485	2.209782	-1.1169660	-1.112383

Table 20 contains the item parameters for students with and without LEP status using expert anchors. Questions 10 25, and 27 and 37 all appear easier for students with a LEP classification, and the DIF on Q38 appears to be largely down to the discrimination.

Figure 20 displays the ICCs of the items with DIF using expert anchors.

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##    -0.1032471    -0.2518222     0.2459693     0.5999250
##
## $CIs
##          sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5 -0.04124728    -0.1006031  0.3014543     0.7352544
## CI_2.5  -0.17678856    -0.4311916  0.1262793     0.3079982
##
## $tests
## P(sDTF.score = 0)
##       0.003474011
```

DTF test is significant

Figure 21 displays the expected number of correct items by LEP category across ability. In this context, students with the LEP classification appear to do better than those students who do not, right around average ability.

Figure 22 contains the signed DTF by LEP category across ability. This suggests a similar situation, where there is DIF in favor of LEP students around average ability.

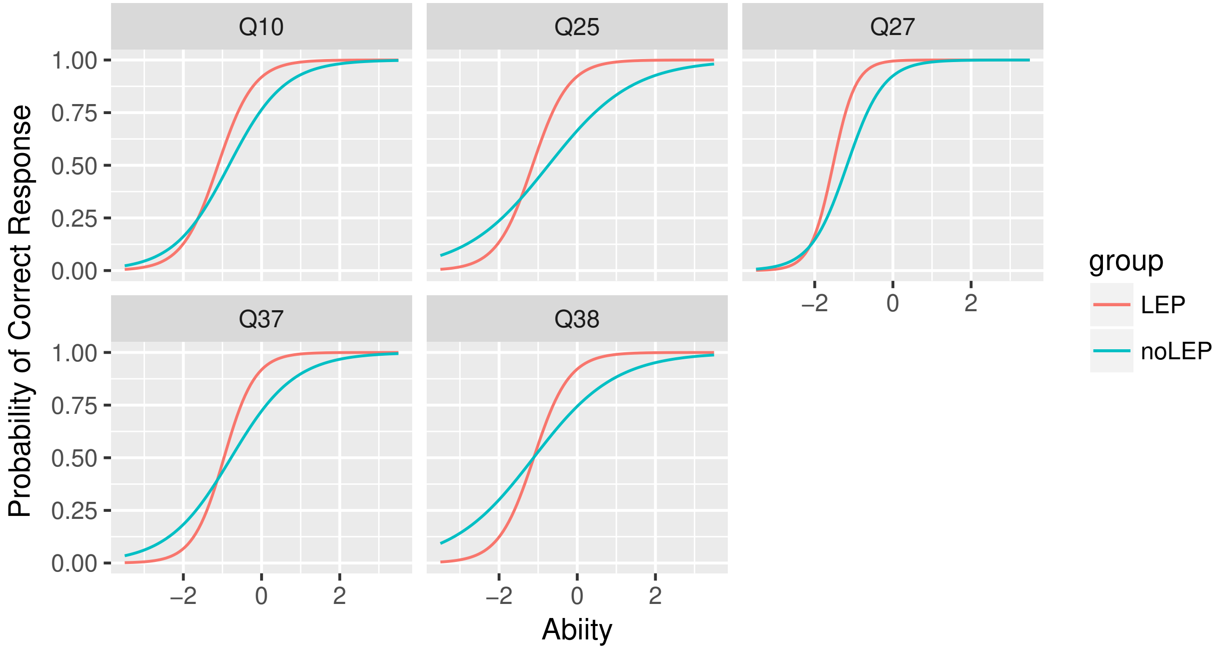

Figure 20: ICCs of items with DIF based on LEP status using expert anchors

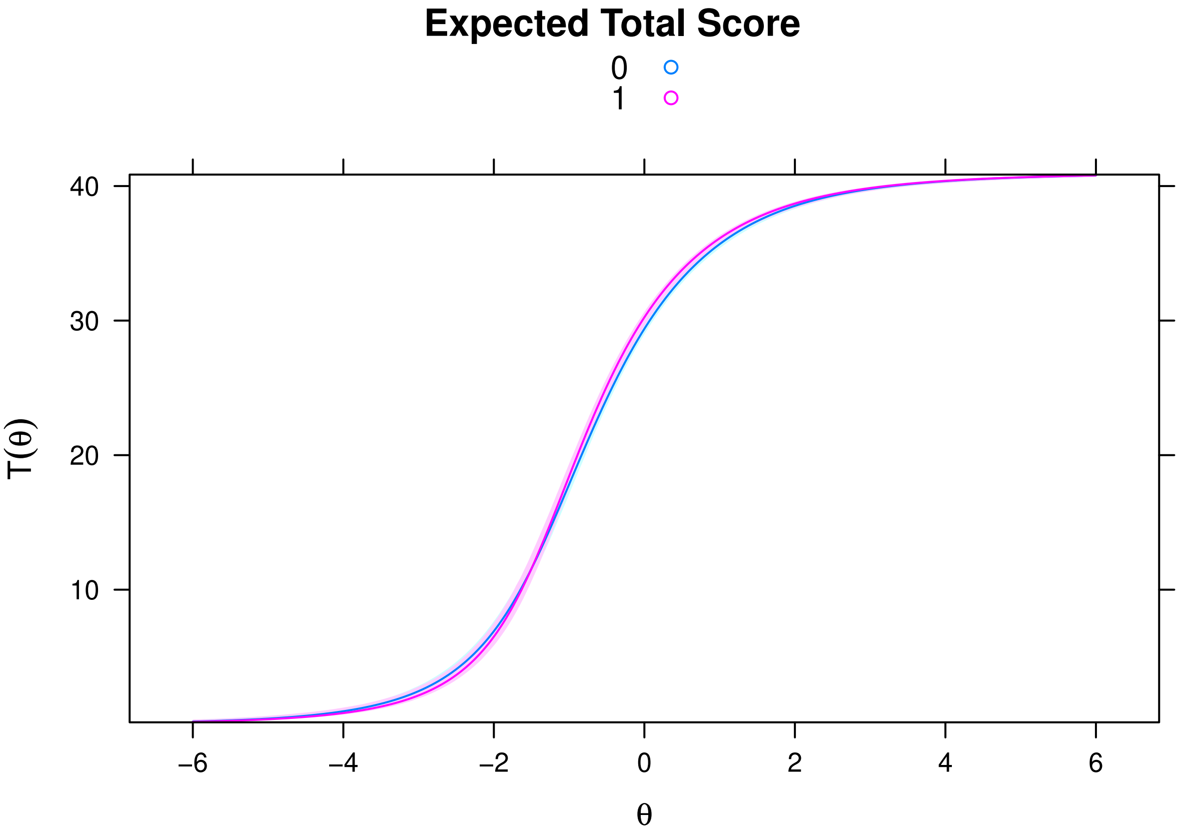

Figure 21: Expected number of correct of items by ability based on LEP, expert anchors

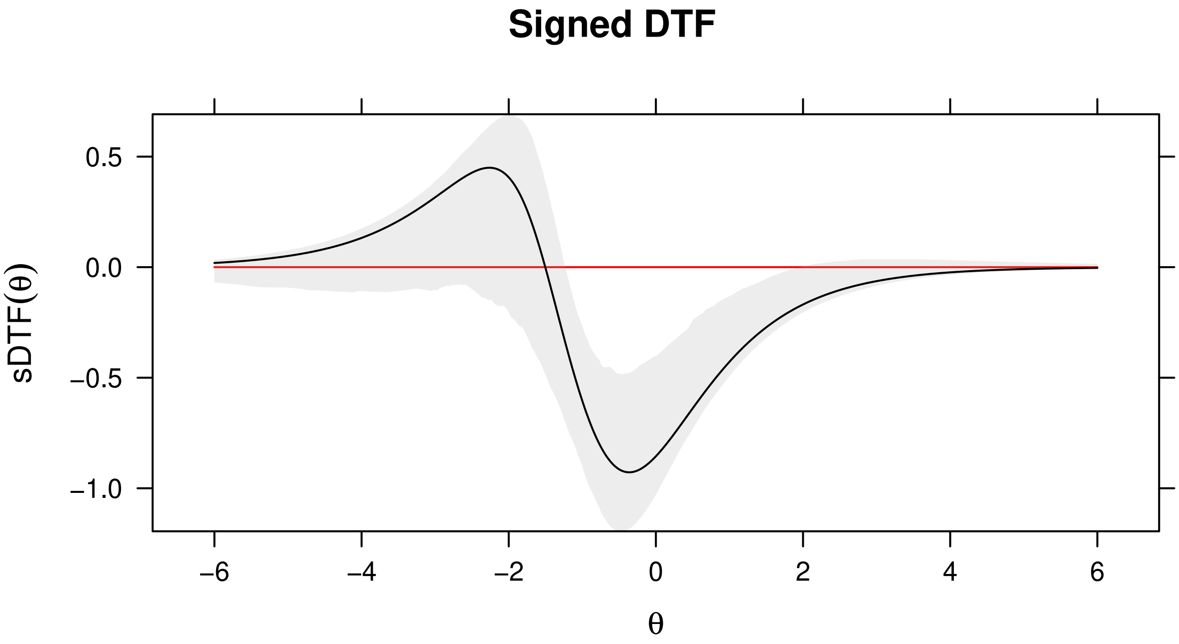

Figure 22: Expected difference in ability estimates based on LEP, expert anchors

Ultimately, I'm not sure strong conclusions can be drawn about DIF with regard to LEP classification. For many of these questions, it not clear to me how LEP classification – or considered the other direction, proficiency in some language other than English – might provide an advantage on these items.

Across all three anchor selection methods, question 25 had DIF. Items 6 and 27 appeared at least twice. Total item scoring is probably still superior to ability estimates for this exam, but there could be some issues.

Race

The analysis of DIF based on race focuses on White, Hispanic, and Black students. The small number of students from other categories would likely cause difficulties.

4-anchor MPT

Table 21: MPT anchors for race

Q	p_count
Q11	8
Q23	9
Q31	6
Q35	8

Table 21 contains the MPT anchors for race-based DIF testing. In this case, items 11, 23, 31, and 35 were selected as anchors.

Table 22: DIF results for race based on MPT anchors

Q	p2
Q9	0.0000295
Q14	0.0009804
Q16	0.0475725
Q25	0.0073028
Q26	0.0323222
Q34	0.0107861

Table 22 contains the DIF results for race using MPT anchors. Items 9, 14, 16, 25, 26, and 34 may all have DIF.

Table 23: Item parameters by race for items with DIF using MPT anchors.

Q	a__black	a__hispanic	a__white	b__black	b__hispanic	b__white
Q9	1.0640421	0.9788874	1.0655231	0.4029454	0.3825511	-0.1294512
Q14	1.0811932	1.1058249	1.1107672	-1.5391969	-0.8934914	-1.4796247
Q16	1.8424933	1.8025059	2.3179061	-1.1116512	-0.8048150	-0.9082697
Q25	0.5609851	0.9584912	0.7743859	-0.1414571	-0.6135394	-0.4473921
Q26	1.2861538	0.9680424	1.0707279	-0.2868585	0.2009142	0.0978455

Table 23 contains the item parameters for those items with DIF when using the MPT anchors. Question 9 appears to be easiest for White students, followed by Hispanic students and then Black students. Questions 14, 16 and 26 appear to be easiest for Black students, followed by White students, and then Hispanic students. Question 25 appears to be easiest for Hispanic students, followed by White students, and then Black students. One racial category does not appear to perform consistently better than another.

Figure 23 displays the ICCs of the items with DIF by race based MPT anchors.

Differential test functioning testing is more involved with three groups, because it would require performing three different pairwise comparisons (Black vs. Hispanic, Black vs. White, White vs. Hispanic). I'm going to leave it for later, time allowing. It seems probable that with this model, Hispanic students are likely to do a little worse than Black or White students.

Iterative Forward Anchor MTT

Table 24: DIF results for race using MTT anchors

Q	p
Q4	0.0457987
Q9	0.0000114
Q13	0.0202015
Q14	0.0002061
Q16	0.0213437
Q25	0.0050899
Q26	0.0114989
Q34	0.0063901
Q38	0.0421930

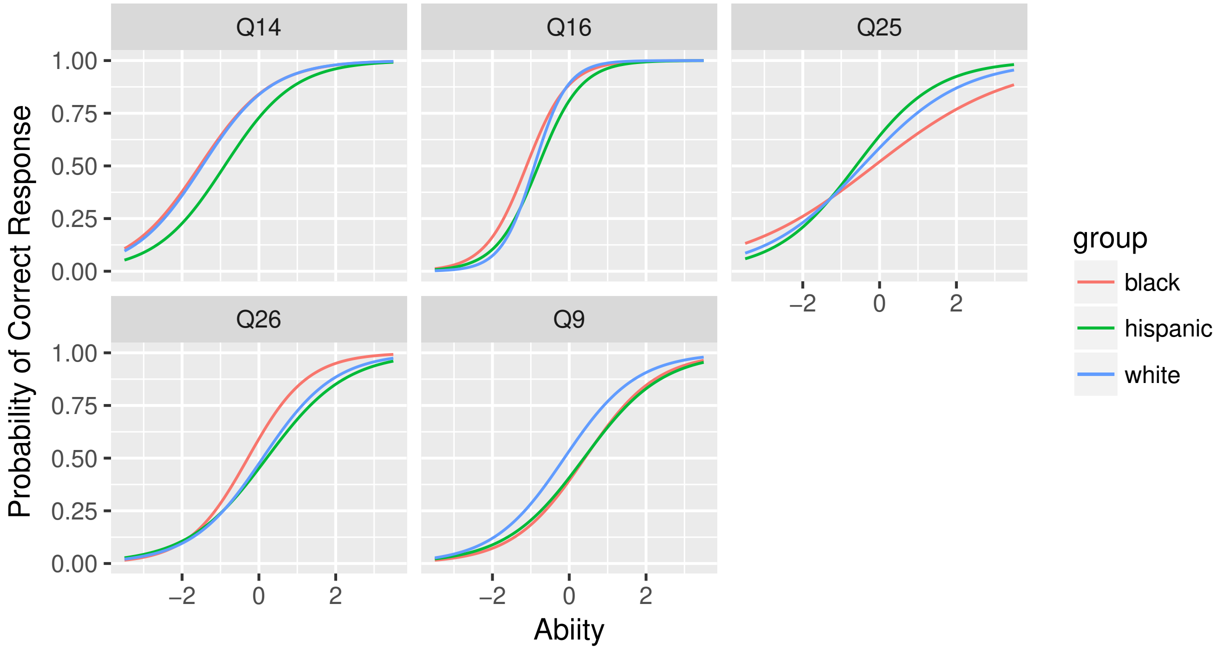

Figure 23: ICCs of items with DIF based on race using 4-anchor MPT

Table 24 contains the results of DIF testing by race using the MTT anchor selection. This method suggests that items 4, 9, 13, 14, 16, 25, 26, 34, and 38 all might have DIF.

Table 25: Item parameters by race for items with DIF using MTT anchors.

Q	a_black	a_hispanic	a_white	b_black	b_hispanic	b_white
Q4	0.6541425	0.6077572	0.4541485	-0.4396285	-0.3307324	0.0617526
Q9	1.0621163	0.9904132	1.0507629	0.4035832	0.3733014	-0.1134413
Q13	1.3711896	1.7384242	1.0588011	-0.9019246	-0.7281536	-0.7824278
Q14	1.0790570	1.1173191	1.0971808	-1.5413703	-0.8879122	-1.4798780
Q16	1.8398999	1.8234457	2.2928513	-1.1123980	-0.7993272	-0.9011531
Q25	0.5595776	0.9735022	0.7646213	-0.1416468	-0.6087585	-0.4347431
Q26	1.2837493	0.9767620	1.0563494	-0.2850798	0.1955829	0.1164977
Q34	1.6726408	1.6210252	1.8032772	-0.1992074	-0.4315799	-0.4460731
Q38	0.7586966	1.0845012	0.8421692	-0.8920561	-0.9263587	-0.7126693

Table 25 contains the item parameters for items with DIF by race using MTT anchors. Question 4 appears easiest for Black students, followed by Hispanic students and then White students. Question 9 appears easiest for White students, followed by Hispanic students, and then Black students. Question 13 appears easiest for Black students, with White and Hispanic students finding it a little harder. Questions 14 and 16 appear easiest for Black students, followed by White students, and then Hispanic students. Question 25 appears easiest for Hispanic students, followed by White students, and then Black students. Question 26 appears easiest for Black students, followed by White students, and then Hispanic students. Question 35 appears easiest for White and Hispanic students, with it being more difficult for Black students. Question 38 appears easiest for Hispanic students, followed by White students, and then Black students. Again, no one group appears to perform consistently better than another.

Figure 24 contains the ICCS for the items with DIF for race using MTT.

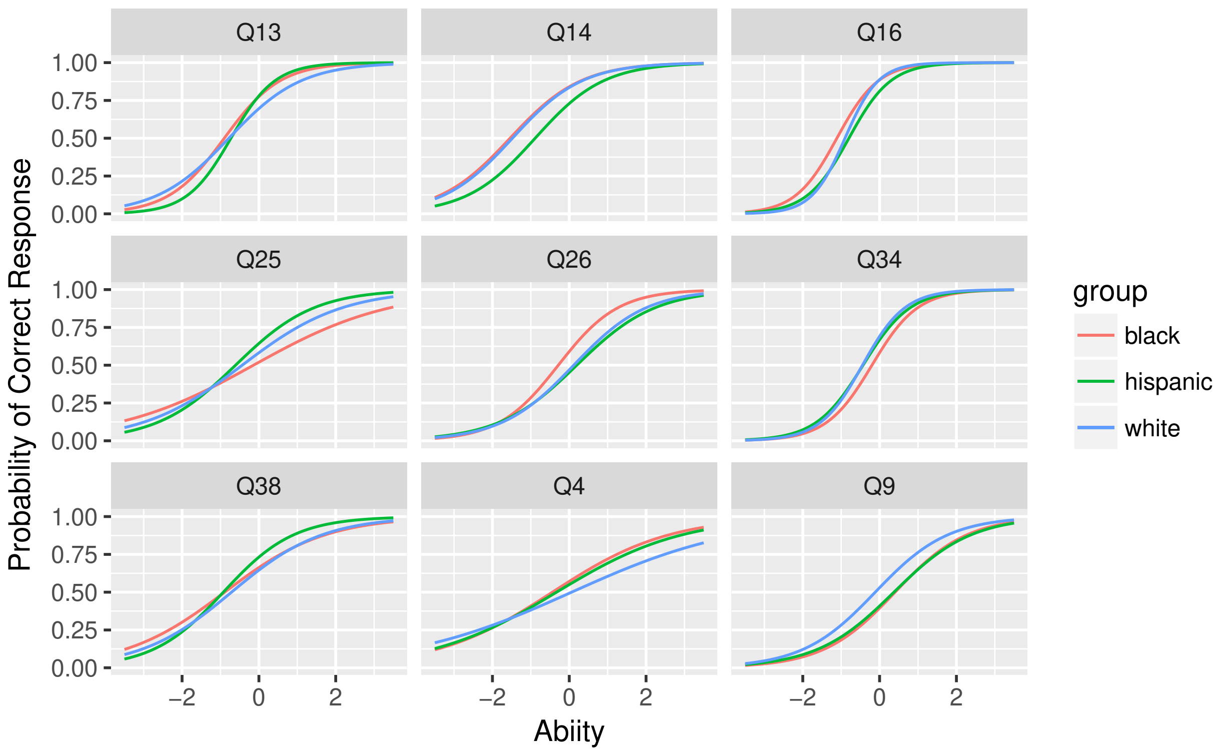

Figure 24: ICCs of items with DIF based on race using iterative MTT

Expert anchors

The possible pool of expert anchors is 2, 4, 7, 9, 14, 15, 22, 29, and 31. Items 4, 9, and 14 were suggested to have possible DIF earlier, so I used the pool of 2, 7, 15, 22, 29, and 31.

Table 26: DIF results for race based on the expert anchors.

Q	p
Q1	0.0389491
Q3	0.0480026
Q4	0.0201712
Q9	0.0012973
Q13	0.0031876
Q14	0.0030245
Q25	0.0065552
Q26	0.0100017
Q27	0.0399750
Q38	0.0134000

Table 26 contains the DIF results for race using expert anchors. Questions 1, 3, 4, 9, 13, 14, 23, 24, 25, 26, 27 and 38 all may have DIF. These results are fairly consistent with the other methods.

Table 27: Item parameters by race for items with DIF using expert anchors

Q	a_black	a_hispanic	a_white	b_black	b_hispanic	b_white
Q1	1.5967943	1.2664312	0.9562365	-1.5072380	-1.4586640	-1.8629575
Q3	1.3906633	1.3838993	1.3839409	-0.2738372	-0.2446710	-0.0489250
Q4	0.6565580	0.6015521	0.4410492	-0.4385290	-0.3355020	0.0935139
Q9	1.0611497	0.9826768	1.0164958	0.4029990	0.3746989	-0.0880127
Q13	1.3798741	1.7233018	1.0261822	-0.8986563	-0.7351626	-0.7776956
Q14	1.0947272	1.1098995	1.0604014	-1.5256566	-0.8953731	-1.5017469
Q23	0.8310361	1.1733717	0.9673600	0.3340454	0.2290841	0.4651012
Q24	2.0000313	1.9303406	1.9029427	-0.9682948	-0.8310488	-0.8491479
Q25	0.5611315	0.9642817	0.7410437	-0.1415035	-0.6155752	-0.4193998
Q26	1.2905701	0.9736415	1.0231231	-0.2842084	0.1957556	0.1487043
Q27	1.9838302	1.7273739	1.6561904	-0.8167007	-1.2172065	-0.9491416
Q38	0.7582668	1.0795826	0.8175794	-0.8923535	-0.9327384	-0.7045642

Table 27 contains the item parameters by race for items with DIF using expert anchors. Questions 1 and 23 appear easier for White students, followed by Black students, and then Hispanic students. Question 3 appears to be about the same for Black and Hispanic students, with White students finding it harder. Question 4 appears easiest for Black students, followed by Hispanic students and then White students. Question 9 appears easiest for White students, followed by Hispanic students, and then Black students. Question 13 appears easiest for Black students, with White and Hispanic students finding it a little harder. Question 14 appears equally difficult for Black and White students, with Hispanic students finding it more difficult. Question 24 appears easiest for White and Hispanic students, with it being more difficult for Black students. Questions 25 and 27 appear easiest for Hispanic students, followed by White students, and then Black students. Question 26 appears easiest for Black students, followed by White students, and then Hispanic students. Question 38 appears easiest for Hispanic students, followed by White students, and then Black students.

Figure 25 contains the ICCs of items with DIF based on race using expert anchors.

Consistently across all three methods for finding anchor items, items 9, 14, 25, and 38 showed up as having DIF. Items 3, 13, 16, 26, and 34 showed up as having DIF in at least 2 of the three. Although I do not have DTF available for race, I suspect that total item scoring may be superior here, given that the patterns (no consistent superior group) match those of the other categories I examined DIF in.

Discussion

There is evidence of some DIF present in the items used in the ASK. For free and reduced lunch status, there appeared to be consensus that items 2, 13, 14, and 38 had DIF, with possible evidence for item 36. Item 36 is notable because it is a convergence of expert anchors *and* one of the low-information methods. For LEP status, only item 25 showed DIF for all three anchor item selection methods. Items 6 and 27 showed up in two of the three methods. It is worth nothing that item 38 showed up as well, although only in the expert anchor condition. For race, items 9, 14, and 25 all showed DIF for all three anchor selection methods, with 2, 13, 16, 26 and 38 appearing in two of the three.

I would argue that there is fairly strong evidence that items 14 and 25 may have DIF across a wide variety of classifications. Typically question 14 was more difficult for groups of students sometimes thought to be disadvantaged. Question 25's behavior was less consistent, sometimes appearing to have DIF in a technical sense but perhaps not a in a meaningful sense, sometimes being easier for students that have been historically disadvantaged, and other times not. Question 38 also seems worth noting, as it appeared to have DIF using at least one anchor selection method in all three of FLR, LEP, and race. It's also interesting because when it had apparent DIF, it was generally easier for students who might be thought to have a historic disadvantage.

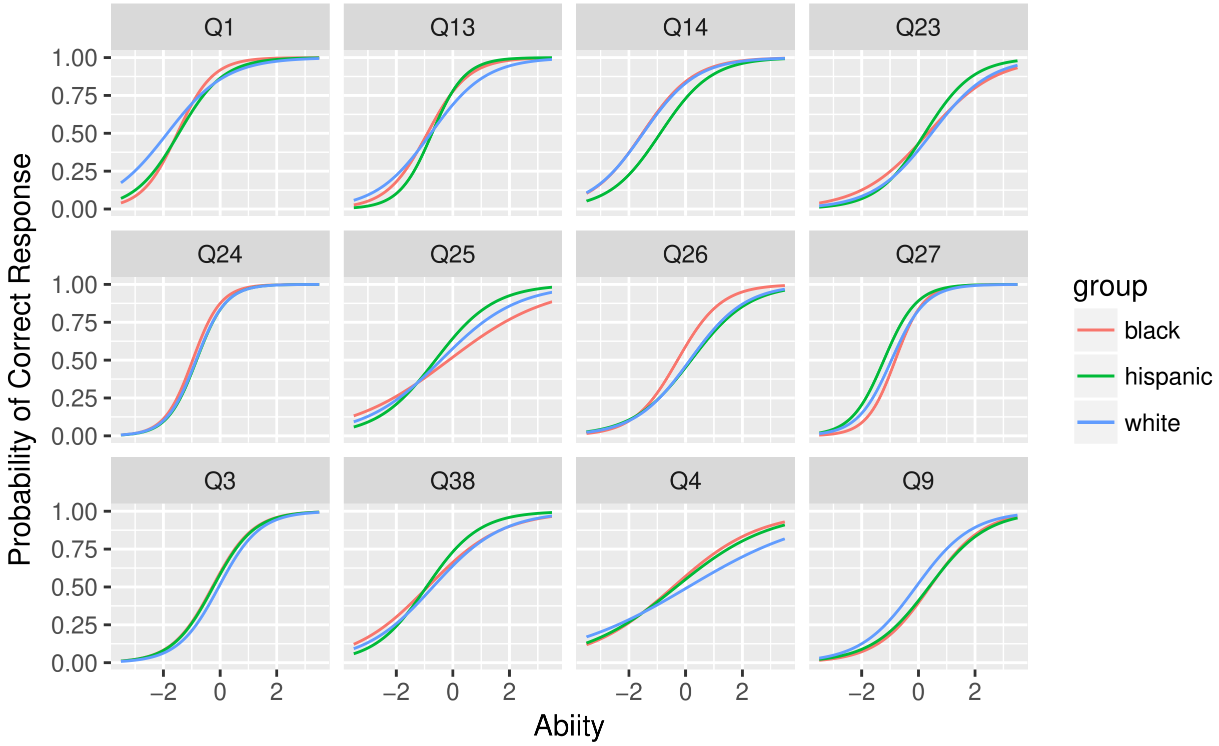

Figure 25: ICCs of items with DIF based on race using expert anchors

Across all three sets of categories, it is not entirely clear that there is evidence for really strong DIF across the whole exam, but perhaps the ability estimates that come out of the models should be viewed with caution. If the number of items correct is used as the benchmark, the bias caused by DIF appears as though it is probably *less* of a problem.

Reading Exam

For the purposes of the DIF analysis, I'll be using the 2PL model as discussed previously. Expert anchors were not solicited for this exam.

Free or Reduced Lunch

4-anchor MPT

Table 28: MPT FRL anchors for the reading exam

Q	p_count
Q43	10
Q44	10
Q51	10
Q61	10

Table 28 lists the FRL anchors for the reading exam using the MPT method. The anchor items selected are items 43, 44, 51, and 61. That involves at least one item from every passage, which seems positive.

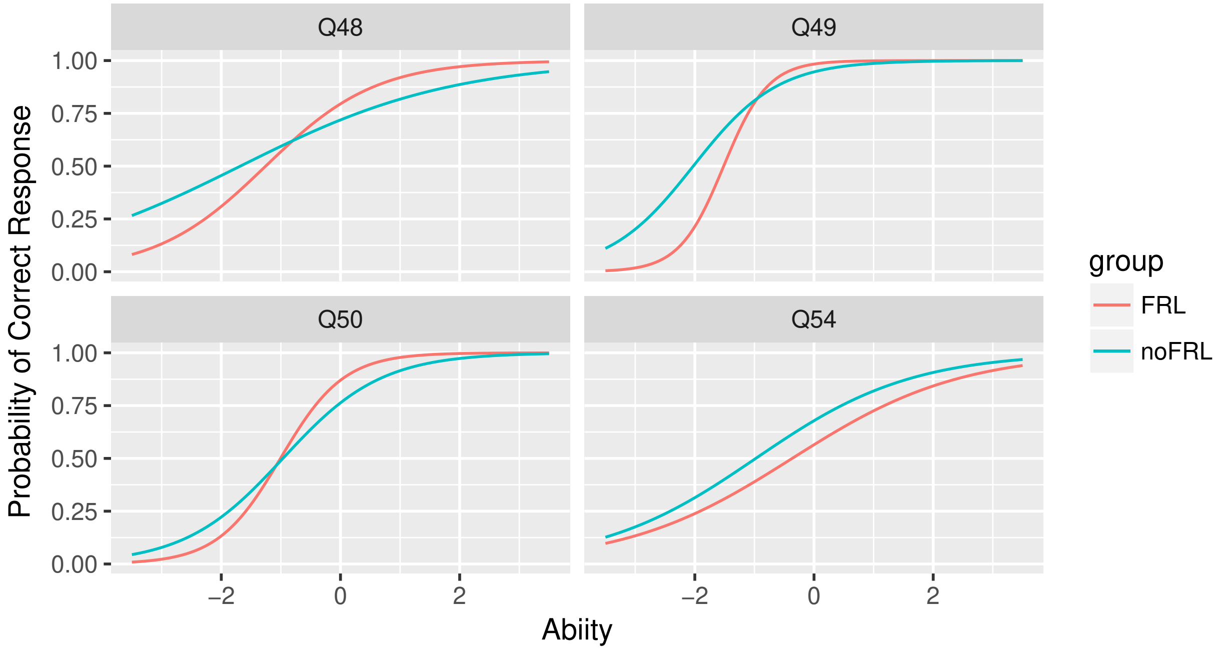

Figure 26: ICCs of items with DIF based on receipt of FRL using 4-anchor MPT

Table 29: DIF results for FRL using MPT anchors

Q	p2
Q48	0.0324796
Q49	0.0109896
Q50	0.0033067
Q54	0.0047219

Table 29 displays the results of DIF testing for FRL using MPT anchors. This method finds that questions 48, 49, 50, and 54 might have DIF.

Table 30: Item parameters for by FRL group for MPT anchors.

Q	a_noFRL	a_FRL	b_noFRL	b_FRL
Q48	0.5584204	1.0791786	-1.6816967	-1.2558016
Q49	1.4159344	2.6904855	-2.0265396	-1.5158106
Q50	1.2112993	1.8898846	-0.9657166	-1.0067616
Q54	0.7645408	0.7104565	-0.9770554	-0.3661432

Table 30 contains the item parameters for the items with DIF using MPT anchors. It appears as though students who receive Free or Reduced lunch may have had a more difficult time with Question 48, 49, and 54. Question 50 may have been easier for students who received Free or Reduced lunch, but not by a lot.

Figure 26 displays the ICCs of the items with DIF for FRL based on MPT anchors.

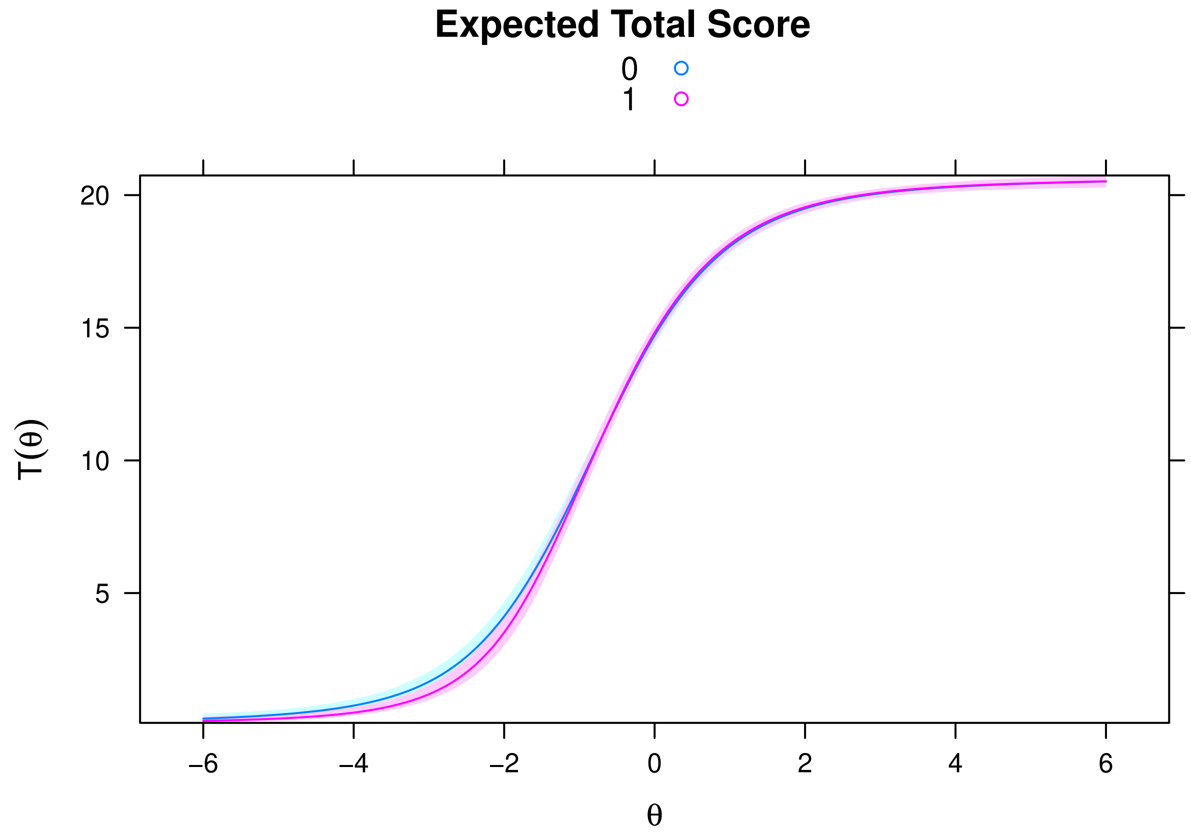

Figure 27: Expected total score on the reading exam for a given ability by FRL status via 4-MPT

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##     0.1138269     0.5420329     0.1629624     0.7760117
##
## $CIs
##         sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5 0.18302863     0.8715649  0.2643703     1.2589062
## CI_2.5  0.05709452     0.2718787  0.0874604     0.4164781
##
## $tests
## P(sDTF.score = 0)
##      0.0003203685
```

DTF test results are significant, but the magnitude of the DTF looks small.

Figure 27 displays the expected number of correct items based on FRL status. There is a fair amount of overlap at the high end, but at the low end of skill it looks like even using the number correct might be slightly biased in favor of those students who don't receive free or reduced lunch.

Figure 28 displays the level of differential test functioning across different levels of ability. Consistent with the other plot, it looks like there's considerable DIF at the low end of the test, but not much at the average to high end.

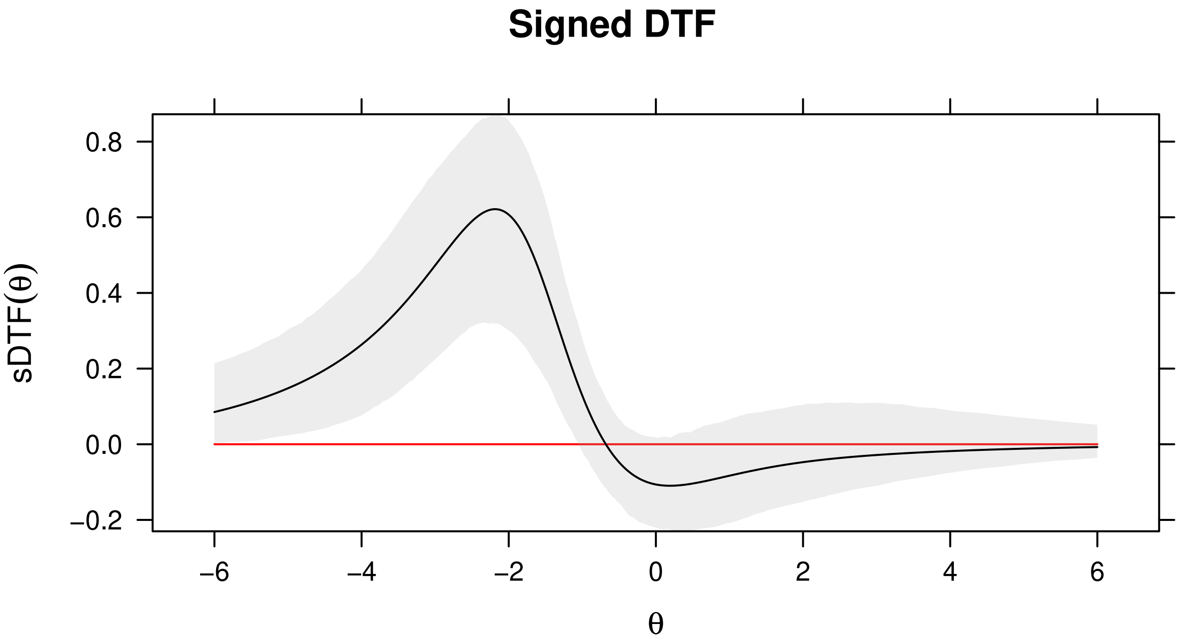

Figure 28: Differential test functioning at different levels of ability by FRL status via 4-MPT

Iterative Forward Anchor MTT

Table 31: DIF results for FRL using mTT

Q	p
Q54	0.000839

Table 31 contains the DIF results for FRL using MTT. This method suggests that question 54 might have DIF.

Table 32: Question 54 ICCs

Q	a_noFRL	a_FRL	b_noFRL	b_FRL
Q54	0.7646526	0.6769673	-0.9765041	-0.3457718

Table 32 displays the item parameters for question 54. Question 54 seems as though it might be more difficult for students who received free or reduced lunch. Figure 29 displays the ICCs for question 54 across the two groups.

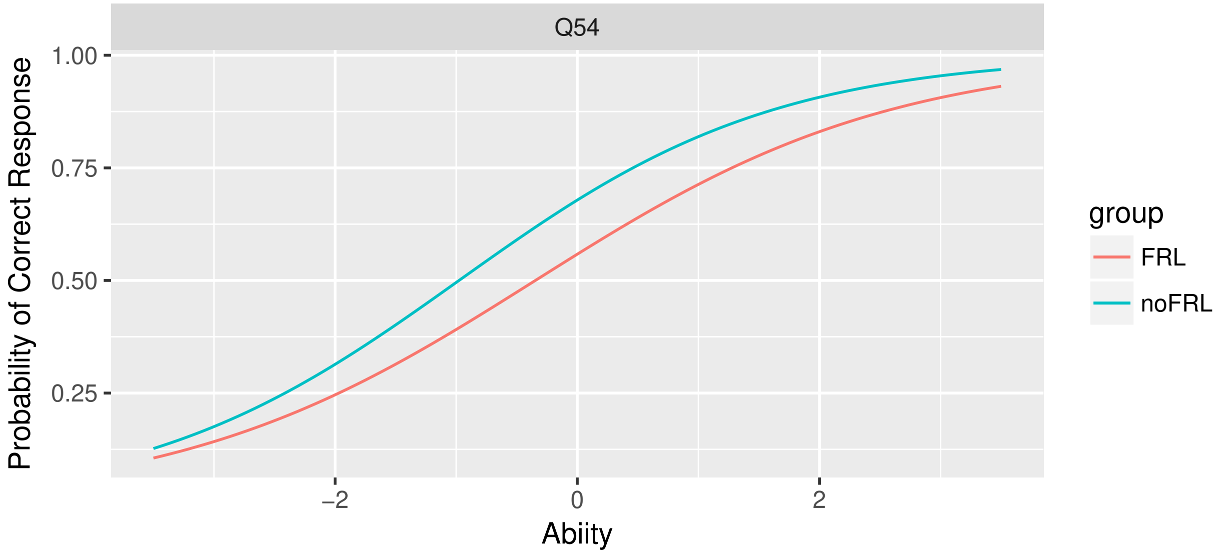

Figure 29: ICCs of Q54 based on receipt of FRL using iterative MTT

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##    0.05170683    0.24622299    0.05171082    0.24624202
##
## $CIs
##         sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5 0.08755382     0.4169230  0.1234840     0.5880192
## CI_2.5  0.02358810     0.1123243  0.0284164     0.1353162
##
## $tests
## P(sDTF.score = 0)
##       0.002122828
```

The DTF results are again small, but the test statistic is significant.

Figure 30 displays the expected number correct by estimated true reading score. In contrast to the MPT anchors, this method suggests that perhaps the use of total scores would be an acceptable scoring method for the reading exam.

Figure 31 displays the level of differential test functioning across different levels of ability. There appears to be differential test functioning near the mean. The use of estimated ability as a scoring method seems inadvisable.

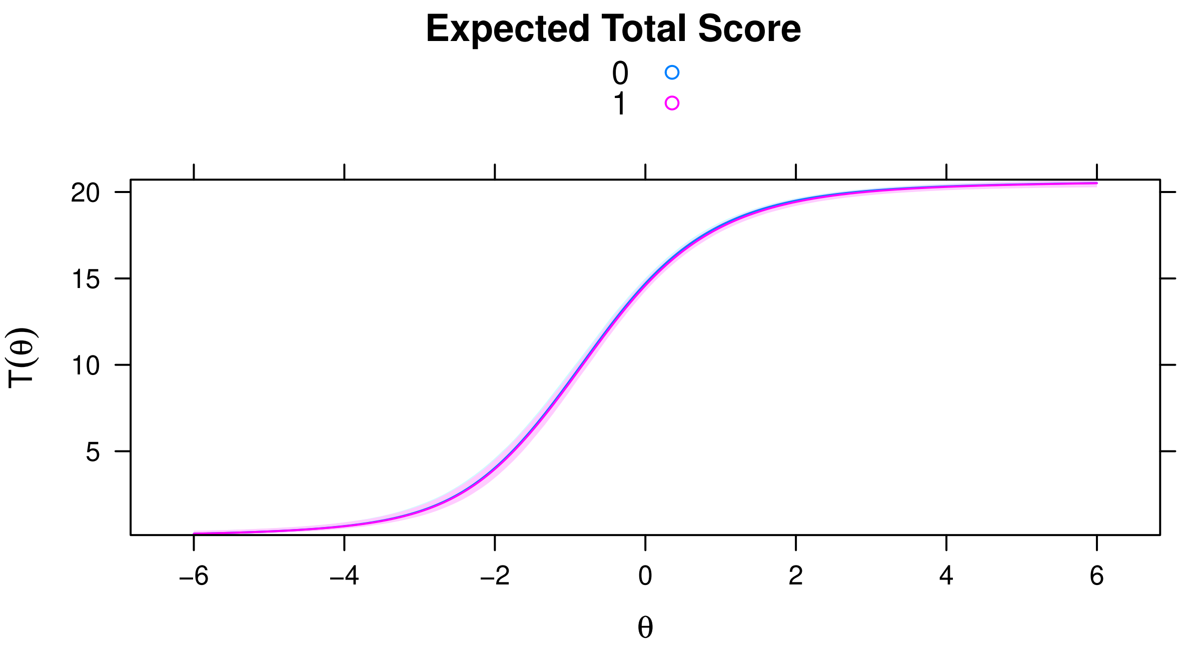

Figure 30: Expected total score for a given ability on the reading test by FRL status via i-MTT

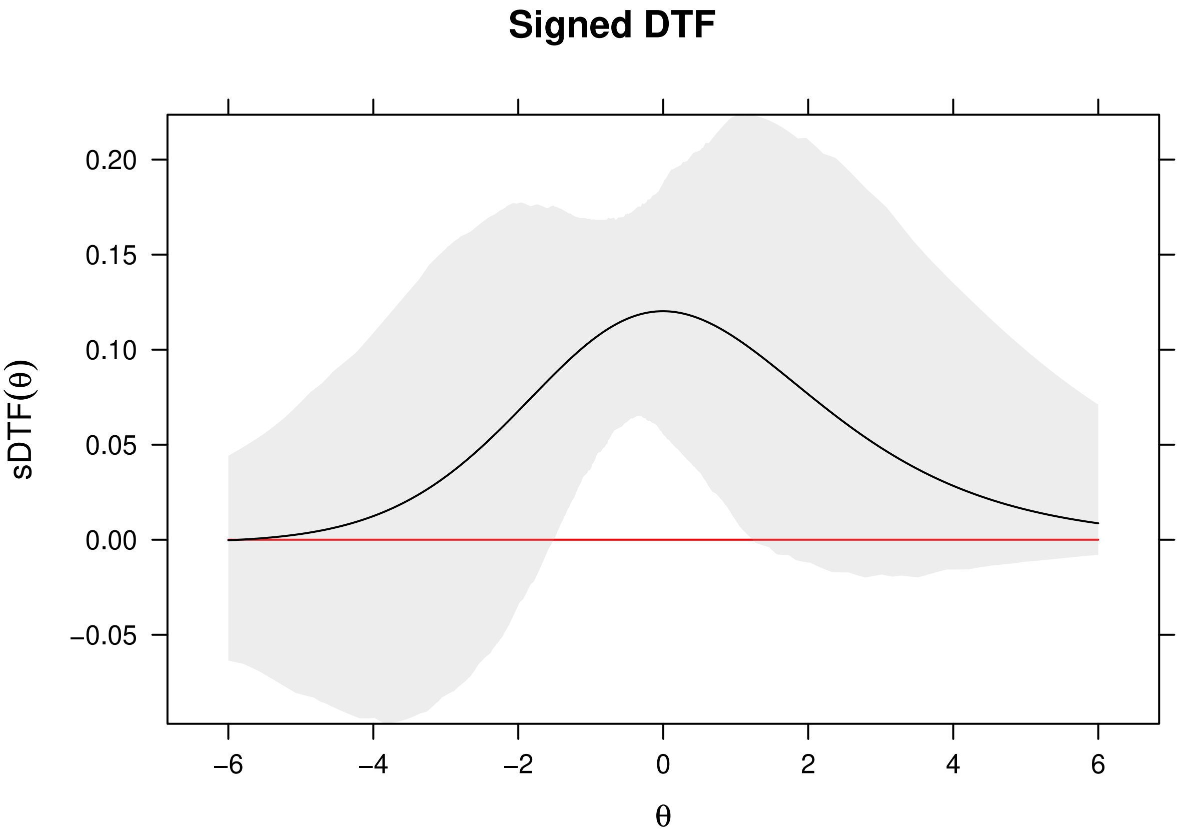

Figure 31: Differential test functioning at different levels of ability on the reading test by FRL status via i-MTT

Limited English Proficency

4-anchor MPT

Table 33: MPT anchors for LEP on the reading exam

Q	p_count
Q48	4
Q53	5
Q59	7
Q61	4

Table 33 list the MPT anchors for LEP on the reading exam. In this case, items 48, 53, 59, and 61 were selected as anchors.

Table 34: DIF results for the reading exam based on LEP using MPT anchors

Q	p2
Q60	0.03011

Table 45 displays the LEP DIF results for the reading exam using MPT anchors. Item 60 may have DIF.

Table 35: Item parameters for question 60

Q	a_noLEP	a_LEP	b_noLEP	b_LEP
Q60	0.1715317	-0.6296341	5.079868	-3.986275

Table 35 contains the item parameters for question 60. Question 60 is negatively discriminating for students who have a limited English proficiency classification. That item was removed in later versions of the exam, so this is perhaps not the worst news. Figure 32 contains the ICCs across groups for Question 60.

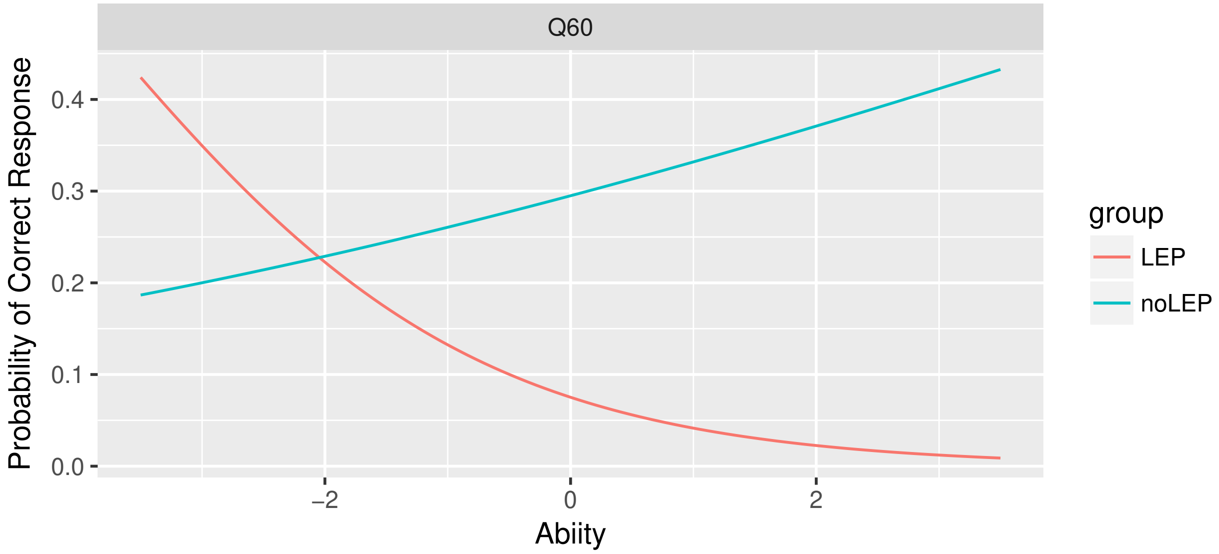

Figure 32: ICCs of Q60 based on LEP status using 4-anchor MPT

```
## $observed
##    sDTF.score sDTF(%).score    uDTF.score uDTF(%).score
##     0.1083632     0.5160153     0.3262042     1.5533534
##
## $CIs
##         sDTF.score sDTF(%).score uDTF.score uDTF(%).score
## CI_97.5  0.2186473     1.0411778  0.5177360     2.4654093
## CI_2.5  -0.1791852    -0.8532626  0.0858156     0.4086457
##
## $tests
## P(sDTF.score = 0)
##         0.2774834
```

Overall significance indicates there's not DTF, but the expected uDTF is higher than previously seen.

Figure 32 displays the expected total scores by estimated ability. At the outside edges of the test, the expected total scores diverge a little bit, but I would imagine that's got a lot to do with the negatively-discriminating item 60. With its removal, the issue would likely disappear.

Figure 32 displays signed DTF for the 4-MPT method. This is consistent with the issue of negative discrimination on item 60. With its removal we could likely assume no DIF or DTF based on the 4-MPT anchor method.

Iterative Forward Anchor MTT

MTT suggests no DIF.

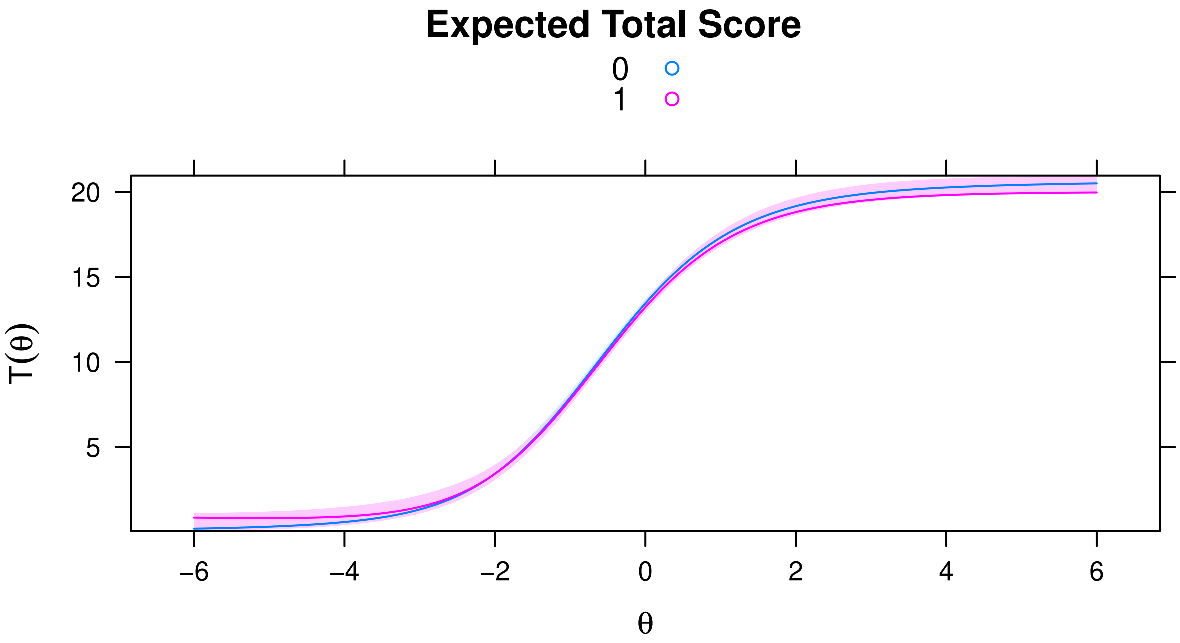

Figure 33: Expected total score for a given ability by LEP status via 4-MPT

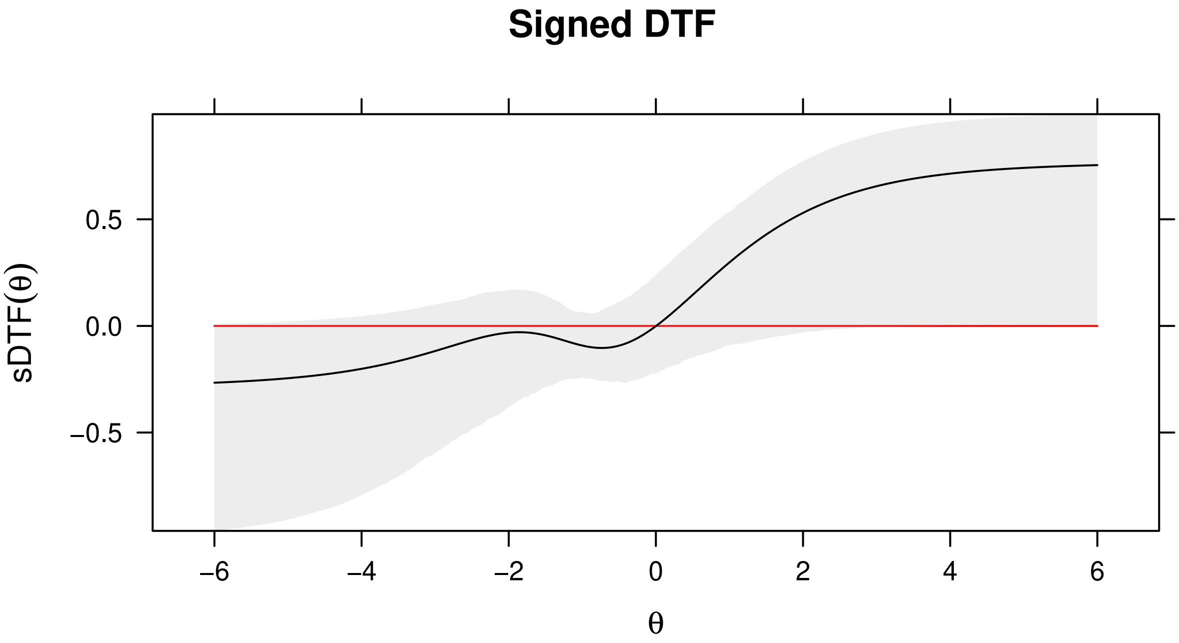

Figure 34: Differential test functioning at different levels of ability by LEP status via 4-MPT

Race

As with the multiple choice exam, the analysis of DIF based on race focuses on White, Hispanic, and Black students.

4-anchor MPT

Table 36: MPT anchors for race on the reading exam

Q	p_count
Q48	6
Q60	7
Q49	10
Q47	10
Q53	10

Table 36 lists the anchors using the MPT method. In this case, items 47, 48, 49, and 60 were chosen. Given potential issues with 60, I'm not going to use that anchor and am instead going to use the runner-up, question 53, in its place.

Table 37: Results of the MPT DIF test for race on the reading exam.

Q	p2
Q54	0.0028342
Q63	0.0493657

Table 37 displays the results of the DIF testing for race based on the MPT anchors. Questions 54 and 63 are suggested as potentially having DIF.

Table 38: Item parameters for questions with DIF across race based on the MPT anchors

Q	a_black	a_hispanic	a_white	b_black	b_hispanic	b_white
Q54	0.4526667	0.5655356	0.5884605	0.3787877	0.3411550	-0.4423288
Q63	0.8383183	1.1038571	1.0051427	-0.2185262	-0.3811992	0.0208383

Table 38 contains the item parameters of the questions with DIF across race. Question 54 appears easier for White students and about the same difficulty for Black and Hispanic students. Question 63 appears easiest for Hispanic students, then Black students, and finally White students.

Figure 35 displays the ICCs of the items with DIF across race using the MPT anchors.

No DTF for race.

Iterative Forward Anchor MTT

Due to issues with model convergence with questions 59 and 63, I have excluded them as possible anchors.

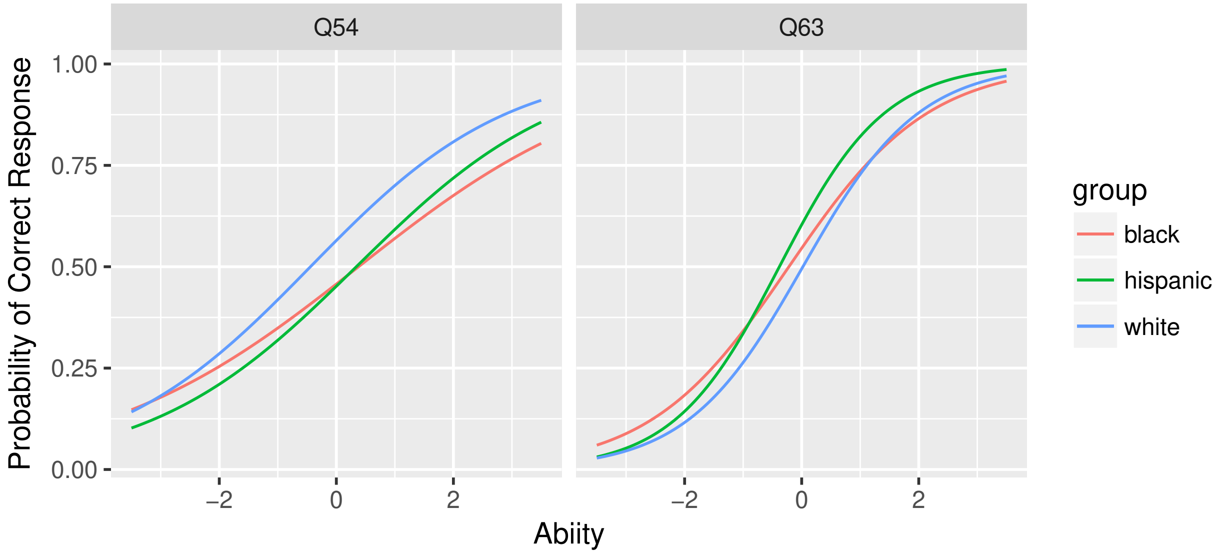

Figure 35: ICCs of items with DIF based on race using 4-anchor MPT

Table 39: Results of DIF testing by race using MTT anchors.

Q	p
Q59	0.0254637
Q63	0.0106361

Table 39 contains the results of the DIF testing. This method suggests that questions 59 and 63 might have DIF. The model estimation difficulties happened when I attempted to model these items equal across groups, so it is unsurprising they showed up as items with DIF.

Table 40: Item parameters for questions with DIF across race using MTT anchors

Q	a_black	a_hispanic	a_white	b_black	b_hispanic	b_white
Q59	1.0071309	1.144799	1.306484	0.4157610	0.4889962	0.6752081
Q63	0.8363717	1.102966	1.003761	-0.2188419	-0.3749140	0.0597896

Table 40 displays the item parameters for the questions with DIF. Both questions appear to be easier for Hispanic and Black students than White students.

Figure 35 contains the ICCs for the items with DIF for race using MTT anchors.

Discussion

The lack of expert anchors reduces the ability to point to converging evidence. However, there does appear to be more than one occasion where items 54, 60, and 63 exhibited DIF. Item 60 was removed in subsequent iterations of the exam, so the problems with that item are of less concern. It appears that, consistent with the subject matter questions, the use of total scores may be a relatively DIF-free way of scoring the exam.

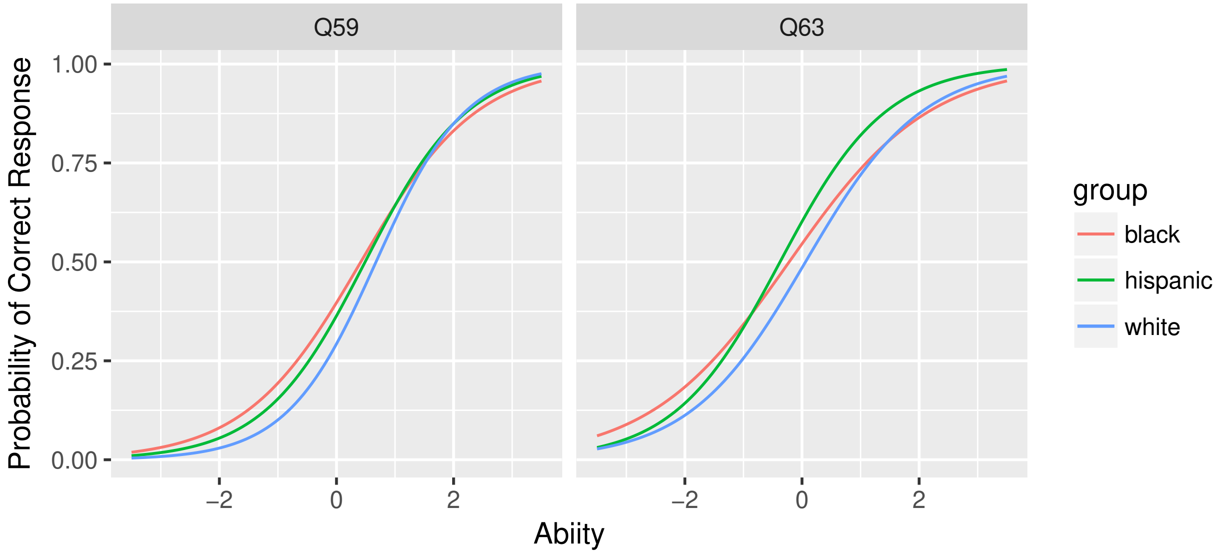

Figure 36: ICCs of items with DIF based on race using iterative MTT

Final Thoughts and Future Directions

There does appear to be some DIF in a variety of categories across the exam. It is difficult to imagine that estimated ability is an unbiased way to estimate the score for many individuals who took it. In general, analyses that depend upon the score of this test should focus on the number of correct items, rather than estimates of ability.

A more careful examination of the subtests with and without the items that might have DIF could be valuable.

All of the analysis contained here was single-level. There is some clustering in this exam, and because abiliy is a latent distribution that has to be estimated correctly for the item-level parameters to be correct, that mis-specification may have had an impact. Further investigation into the impact of clustering on item-level estimates might be worthwhile.

Works Cited

Bradlow, E. T., Wainer, H., & Wang, X. (1999). A Bayesian random effects model for testlets. Psychometrika, 64, 153-168.

Cai, L., & Hansen, M. (2013). Limited-information goodness-of-fit testing of hierarchical item factor models. British Journal of Mathematical and Statistical Psychology, 66(2), 245-276.

Chalmers, R., P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. Journal of Statistical Software, 48(6), 1-29. 10.18637/jss.v048.i06

Hallquist, M. N. & Wiley, J. F. (2018). MplusAutomation: An R Package for Facilitating Large-Scale Latent Variable Analyses in Mplus Structural Equation Modeling, 1-18. doi: 10.1080/10705511.2017.1402334.

Kopf, J., Zeileis, A., & Strobl, C. (2015). Anchor selection strategies for DIF analysis: Review, assessment, and new approaches. Educational and Psychological Measurement, 75(1), 22-56.

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

Appendix 1 - R code for replication of the full analysis.

```r
library(haven) #reads in datastets from other statistical packages, like SPSS
library(tidyverse) #useful data management tools
ask2 <- read_spss("ASK8_1112.sav") #Read in Y2
ask3 <- read_spss("ALLASK8_1213.sav") # Read in Y3

#Specify a year variable for use later
ask2$year <- "Y2"
ask3$year <- "Y3"

#combines the rows of the two datasets
#the code in ask2 specifies that we only want those columns from the
#ask2 data that are also in the ask3 data
ask23 <- bind_rows(ask3, ask2[,names(ask2) %in% names(ask3)])

#create a subset of the data that contains only dichotomously coded variables
ask23_postmc <- select(ask23, SID, ASK2_1:ASK2_31)

#remove anyone who did not answer any questions at all
ask23_postmc <- ask23_postmc[(rowSums(is.na(ask23_postmc))) < 42,]
```

Because the ordering of questions changed from Y2 to Y3, the “numbering” in the reduced dataset is a little bit wonky. This ordering reflects the ordering from Y3 onward, so the ordering is actually Q1-Q42 in Y3 order. I changed the name of these variables to reflect the Y3 ordering, and then ran a 2PL model. I removed item 39, because its discrimination parameter was negative, indicating that it higher-ability individuals were less likely to answer it correctly.

```r
#set the names to the Y3 order.
names(ask23_postmc)[-1] <- paste0("Q", 1:42)

#run the 2pl model
mc2pl <- mirt(ask23_postmc[,c(2:39,41:43)],
              'F = 1-41',
              verbose = FALSE,
              technical = list(removeEmptyRows = TRUE,
                               message = FALSE))

#build a dataframe of the IRT parameters
mc_params <- data.frame(coef(mc2pl, IRTpars = TRUE, as.data.frame = TRUE)) %>%
  mutate(label = row.names(.)) %>%
  slice(1:164) %>%
  separate(col = 2, into = c("Q", "statistic")) %>%
  spread(key = statistic, value = par) %>%
  mutate(qnum = as.numeric(gsub("Q", "", Q))) %>%
  arrange(qnum) %>%
  select(Question = Q, Discrimination = a,  Difficulty = b)
```

I built out some custom functions to calculate item information, test information, and the standard errors around ability estimates, then used the ggplot to display those results.

```
item_info <- function(a, b, theta){
  P = 1/(1  + exp(-a * (theta - b)))
  Q = 1-P
  a^2 * P * Q
}

total_info <- function(a, b, theta){
  x <- sapply(theta, item_info, a = a, b = b, simplify = TRUE)
  colSums(x)
}

total_se <- function(a,b,theta){
  1/total_info(a,b,theta)
}

#score the multiple choice test based on the 2PL model
mcability <- data.frame(fscores(mc2pl, full.scores.SE = TRUE))

#add ability scores back onto the original data.
ask23_mcabil <- bind_cols(ask23_postmc, mcability) %>%
  left_join(ask23)

#plot ability estimates along with standard errors
ggplot(ask23_mcabil, aes(x = F1, y = SE_F1, color = year)) +
  geom_point() +
  geom_hline(yintercept = 0.45, linetype = "dashed") +
  labs(x = "Estimated Ability", y = "Standard Error")

# This is a way of building a set of stat_function calls
# that can be appended onto a ggplot to stack them easily
curves_icc <- mc_params %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = Question),
                  fun = function(x) 1/(1  + exp(-dots$Discrimination * (x - dots$Difficulty))),
```

```r
                  xlim = c(-3.5, 3.5))
  })

# Print out ICCs
ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc +
  guides(color = FALSE) +
  labs(x = "Abiity", y = "Probability of Correct Response")

# Stacking information function stat_function calls
curves_inf <- mc_params %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = Question),
                  fun = function(x) item_info(dots$Difficulty, dots$Discrimination, theta = x),
                  xlim = c(-3.5, 3.5))
  })

#plot Information functions
ggplot(data.frame(x = 0), aes(x = x)) +
  curves_inf +
  guides(color = FALSE) +
  labs(x = "Ability", y = "Information")

# Plot total information
ggplot(data.frame(x=0), aes(x = x)) +
  stat_function(fun = total_info, args = list(a = mc_params$Discrimination,
                                              b = mc_params$Difficulty),
                xlim = c(-3.5, 3.5)) +
   labs(x = "Ability", y = "Information")
```

Reading Comprehension

```r
library(mirt)
#Get just the post-test scores
ask23_postrc <- select(ask23, SID, ASK2_52:ASK2_66)

# remove anyone missing on all answers
ask23_postrc <- ask23_postrc[(rowSums(is.na(ask23_postrc))) < 21,]

# This line gives the items sequential numbering and makes
# the output easier to read
names(ask23_postrc)[-1] <- paste0("Q", 43:63)

# this asks for an IRT model with a single factor
# by default MIRT runs a 2PL model
twopl <- mirt(ask23_postrc[,2:22], 1,
              verbose = FALSE)

# get item parameters in a dataframe, and then also calculate difficulties.
twopl_params <- data.frame(coef(twopl, simplify = TRUE)$item[,1:2]) %>%
  mutate(b = -d/a1)

# Calculate limited-information fit statistics
# imputation is necessary due to missing data.
twopl_M2 <- M2(twopl, impute = 50)


#estimate a bifactor model with one general factor
#and three specific factors
#takes just a little under 2500 iterations
bifactor <- bfactor(ask23_postrc[,2:22],
                    model = c(1,1,1,1,1,1,1,2,2,2,2,2,2,2,3,3,3,3,3,3,3),
                    technical = list(removeEmptyRows = TRUE,
                                     NCYCLES = 2500))

#Calcualte the M2 fit statistic
bif_M2 <- M2(bifactor, impute = 50)
```

DIF code - multiple choice

```
load("descriptives.Rdata")

#adds descriptives onto the data
ask23mcdesc <- left_join(ask23_postmc, descriptives)
```

```
# a function to gather all the test statistics and p-values for the test
# given a single anchor item
get_DIFs <- function(dat, difvect, anchoritem, cycles) {

  # DIF testing requires a character vector or factor
  # convert it to a character if it is not
  if(typeof(difvect) != "character") difvect <- as.character(difvect)

  #freely estimate a 2PL model for each group where the only things
  #constrained to be equal are the single anchor item
  modelmult <- multipleGroup(dat,
                             paste0('F = 1-41\nCONSTRAINB = (',
                                    anchoritem,
                                    ' , a1), (',
                                    anchoritem,', d)'),
                             group = difvect,
                             invariance = c("free_means", "free_var"),
                             SE = TRUE,
                             verbose = FALSE,
                             technical = list(NCYCLES = cycles))

  #estimate DIF for each item in the test other than the anchor item
  diff <- DIF(modelmult, which.par = ("d"),
              technical = list(NCYCLES = cycles),
              verbose = FALSE, warn = FALSE, message = FALSE)
```

```r
  #create a dataframe with just the item number, the p value, and the chi-square value
  diffdf <- data.frame(unlist(diff)) %>%
    mutate(var = row.names(.)) %>%
    rename("value" = !!names(.[1])) %>%
    separate(var, into = c("Q", "stat")) %>%
    spread(key = stat, value = value) %>%
    select(Q, p2, X22)

  return(diffdf)
}

#a vector of anchor item numbers
anchoritem <- 1:41

# starts a cluster for parallel processing. This speeds up this step considerably
# But even on an i-7 with 8 threads this will take several hours
mirtCluster()

#run the function for each anchor
dif_vals_FRL <- plyr::adply(anchoritem, 1,
                            .fun = get_DIFs,
                            dat = ask23mcdesc[,c(2:39,41:43)],
                            difvect = ask23mcdesc$FRL_RECODE, cycles = 5000)

#closes the cluster
mirtCluster(remove = TRUE)
#save the results
#save(dif_vals_FRL, file = "difresultsFRL.Rdata")

#load("difresultsFRL.Rdata")

#get summary mean of p-values and chi-squares for each item
dif_summary_FRL <- dif_vals_FRL %>%
  group_by(Q) %>%
  summarise(mean_p = mean(p2),
            mean_chi = mean(X22))

#determine the 0.5 * 41th mean for the p-values and chi-squares
#(in this case the median)
dif_medians_FRL <- dif_summary_FRL%>%
  summarise(p_cuttoff = median(mean_p),
            chi_cutoff = median(mean_chi))

#determine the counts for the number of p-values and
#chi-squares below their respective cutoffs
dif_counts_FRL <- dif_vals_FRL %>%
  mutate(p_above = p2 < dif_medians_FRL$p_cuttoff,
         chi_above = X22 < dif_medians_FRL$chi_cutoff) %>%
  group_by(Q) %>%
  summarise(p_count = sum(p_above),
            chi_count = sum(chi_above)) %>%
  ungroup()
```

```r
#select top 4 based on counts
top4 <- left_join(dif_counts_FRL, dif_summary_FRL) %>%
  arrange(p_count) %>%
  slice(1:4)

kable(top4)

#estimate multiple group model with the anchor items fixed.
model_4mpt_FRL <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (8,10,17,20, a1), (8,10,17,20, d)',
                           group = as.character(ask23mcdesc$FRL_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


#activate a cluster for parallel computation
mirtCluster()

#perform DIF testing
dif_4mpt_FRL <- DIF(model_4mpt_FRL, which.par = ("d"),
            technical = list(NCYCLES = 5000),
            items2test = c(1:7, 9, 11:16, 18:19, 21:41))

#deactivate cluster
mirtCluster(remove = TRUE)

# Turn it into a dataframe for easy manipulation to find
# Which items have p values < .05
difdf_4mpt_FRL<- data.frame(unlist(dif_4mpt_FRL)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value) %>%
  select(Q, p2) %>%
  filter(p2 < .05)

# Fix all items without DIF to be equal across groups and then
# Run a multiple group 2PL model
FRL_2pl_final <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-2, 4-12, 15-24, 26-37, 39-41, a1),
                           (1-2, 4-12, 15-24, 26-37, 39-41, d)',
                           group = as.character(ask23mcdesc$FRL_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

#extract item parameters for the no FRL group
FRL_gr1 <- data.frame(coef(FRL_2pl_final, simplify = TRUE)$`0`$items)[,1:2]
```

```r
#rename columns
names(FRL_gr1) <- c("a_noFRL", "d_noFRL")

FRL_gr2 <- data.frame(coef(FRL_2pl_final, simplify = TRUE)$`1`$items)[,1:2]
names(FRL_gr2) <- c("a_FRL", "d_FRL")

#Bind the columns together for easy comparison, calculate
#difficulty from the d parameter
FRL_dif_items <- bind_cols(FRL_gr1, FRL_gr2) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(3,13,14,25,38)) %>%
  mutate(b_noFRL = -d_noFRL/a_noFRL,
         b_FRL = -d_FRL/a_FRL) %>%
  select(Q, a_noFRL, a_FRL, b_noFRL, b_FRL)

# create a "long" version of the dataset in order to plot the ICCs
# by group
FRL_dif_items_long <- FRL_dif_items %>%
  gather(key = "param", value = "value", c(a_noFRL, a_FRL, b_noFRL, b_FRL)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

# stack stat function calls for ggplot later
curves_icc_dif <- FRL_dif_items_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

#plot the curves
ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

# set seed for the DTF random draws
set.seed(05142017)

# get DTF values
DTF(FRL_2pl_final, draws = 1000)

# plot expected items correct vs ability estimates
DTF(FRL_2pl_final, draws = 1000, plot = "func")

# plot ability bias along ability distribution
DTF(FRL_2pl_final, draws = 1000, plot = "sDTF")

#build out a sorted dataframe in order to
#use it to iteratively test DIF
```

```
mtt_sorted_FRL <- left_join(dif_counts_FRL, dif_summary_FRL) %>%
  arrange(desc(chi_count), mean_chi) %>%
  mutate(q_num = as.numeric(str_remove(Q, "Q")),
         q_num = ifelse(q_num < 40, q_num, q_num -1))

mirtCluster()

# This for loop starts with a single anchor, tests DIF
# For all other items and as long as there are more items
# remaining without DIF than there are anchors
# it loops around to the enxt time and adds the next
# best anchor. Can take a while
for(j in 1:40) {
  #multiple group model
model_mtt_FRL <- multipleGroup(ask23mcdesc[,c(2:39,41:43)],
                            paste0('F = 1-41\nCONSTRAINB = (',
                                   paste(mtt_sorted_FRL$q_num[(1:j)],
                                         collapse = ","),' , a1), (',
                                   paste(mtt_sorted_FRL$q_num[(1:j)],
                                         collapse = ","),', d)'),
                            group = as.character(ask23mcdesc$FRL_RECODE),
                            invariance = c("free_means", "free_var"),
                            SE = TRUE,
                            technical = list(NCYCLES = 5000))

#test DIF
dif_mtt_FRL <- DIF(model_mtt_FRL, which.par = ("d"),
                 technical = list(NCYCLES = 5000),
                 items2test = c(1:42)[-(mtt_sorted_FRL$q_num[(1:j)])],
                 verbose = FALSE)

#get parameters into a dataframe
difdf_mtt_FRL<- data.frame(unlist(dif_mtt_FRL)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

#get all items without DIF
undif_FRL <- difdf_mtt_FRL %>%
  select(Q, p2) %>%
  filter(p2 > .05)

# test to see if there are as many anchor items as items without DIF
if(j >= nrow(undif_FRL)) break()
}

mirtCluster(remove = TRUE)
#save(difdf_mtt_FRL, file = "FRL_mtt_results.RData")

#load("FRL_mtt_results.RData")

# Grab just those items with DIF
```

```r
mtt_dif_FRL <- filter(difdf_mtt_FRL, p2 < .05) %>%
  select(Q, p = p2)

#Similar DIF comparisons to the MTT method
FRL_2pl_final_mtt <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-2, 4-8, 10-12, 15-35, 37, 39-41, a1),
                           (1-2, 4-8, 10-12, 15-35, 37, 39-41, d)',
                           group = as.character(ask23mcdesc$FRL_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

FRL_gr1 <- data.frame(coef(FRL_2pl_final_mtt, simplify = TRUE)$`0`$items)[,1:2]
names(FRL_gr1) <- c("a_noFRL", "d_noFRL")

FRL_gr2 <- data.frame(coef(FRL_2pl_final_mtt, simplify = TRUE)$`1`$items)[,1:2]
names(FRL_gr2) <- c("a_FRL", "d_FRL")

FRL_dif_items <- bind_cols(FRL_gr1, FRL_gr2) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(3,9,13,14,36,38)) %>%
  mutate(b_noFRL = -d_noFRL/a_noFRL,
         b_FRL = -d_FRL/a_FRL) %>%
  select(Q, a_noFRL, a_FRL, b_noFRL, b_FRL)

FRL_dif_items_long <- FRL_dif_items %>%
  gather(key = "param", value = "value", c(a_noFRL, a_FRL, b_noFRL, b_FRL)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- FRL_dif_items_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

# More DTF testing
set.seed(05142018)
DTF(FRL_2pl_final_mtt, draws = 1000)

DTF(FRL_2pl_final_mtt, draws = 1000, plot = "func")
DTF(FRL_2pl_final_mtt, draws = 1000, plot = "sDTF")

expert_2pl_FRL2 <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                               'F = 1-41
                           CONSTRAINB = (2, 4, 7, 15, 22, 29, 31, a1), (2, 4, 7, 15, 22, 29, 31, d)',
                               group = as.character(ask23mcdesc$FRL_RECODE),
                               invariance = c("free_means", "free_var"),
                               SE = TRUE,
                               verbose = FALSE,
                               technical = list(NCYCLES = 5000))

mirtCluster()
expert_dif_FRL2 <- DIF(expert_2pl_FRL2, which.par = "d")
mirtCluster(remove = TRUE)
# save(expert_dif_FRL2, file = "expert_dif_frl2.RData")

# load("expert_dif_frl2.RData")
difdf_expert_FRL2<- data.frame(unlist(expert_dif_FRL2)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

dif_only_expert_FRL2 <- filter(difdf_expert_FRL2, p2 < .05) %>%
  select(Q, p = p2)

FRL_2pl_final_expert <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-2, 4-12, 15-35, 37, 39-41, a1),
                           (1-2, 4-12, 15-35, 37, 39-41, d)',
                           group = as.character(ask23mcdesc$FRL_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


FRL_gr1 <- data.frame(coef(FRL_2pl_final_expert, simplify = TRUE)$`0`$items)[,1:2]
names(FRL_gr1) <- c("a_noFRL", "d_noFRL")

FRL_gr2 <- data.frame(coef(FRL_2pl_final_expert, simplify = TRUE)$`1`$items)[,1:2]
names(FRL_gr2) <- c("a_FRL", "d_FRL")

FRL_dif_items <- bind_cols(FRL_gr1, FRL_gr2) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(3,13,14,36,38)) %>%
  mutate(b_noFRL = -d_noFRL/a_noFRL,
         b_FRL = -d_FRL/a_FRL) %>%
  select(Q, a_noFRL, a_FRL, b_noFRL, b_FRL)

kable(FRL_dif_items)

FRL_dif_items_long <- FRL_dif_items %>%
  gather(key = "param", value = "value", c(a_noFRL, a_FRL, b_noFRL, b_FRL)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- FRL_dif_items_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

DTF(FRL_2pl_final_expert, draws = 1000)

DTF(FRL_2pl_final_expert, draws = 1000, plot = "func")

DTF(FRL_2pl_final_expert, draws = 1000, plot = "sDTF")

## Perform LEP DIF testing
ask23mcdesc <- ask23mcdesc %>%
  mutate(LEP_RECODE = ifelse(LEP == 0, 0, 1))

LEPdif_vals <- plyr::adply(anchoritem, 1, .fun = get_DIFs,
                           dat = ask23mcdesc[,c(2:39,41:43)],
                           difvect = ask23mcdesc$LEP_RECODE,
                           cycles = 5000)

# save(LEPdif_vals, file = "LEPdifresults.Rdata")

# load("LEPdifresults.Rdata")

#get summary mean of p-values and chi-squares for each item
dif_summary_LEP <- LEPdif_vals %>%
  group_by(Q) %>%
  summarise(mean_p = mean(p2),
            mean_chi = mean(X22))

#determine the 0.5 * 41th mean for the p-values and chi-squares
#(in this case the median)
dif_medians_LEP <- dif_summary_LEP%>%
  summarise(p_cuttoff = median(mean_p),
            chi_cutoff = median(mean_chi))

#determine the counts for the number of p-values and
#chi-squares below their respective cutoffs
dif_counts_LEP <- LEPdif_vals %>%
  mutate(p_above = p2 < dif_medians_LEP$p_cuttoff,
         chi_above = X22 < dif_medians_LEP$chi_cutoff) %>%
  group_by(Q) %>%
  summarise(p_count = sum(p_above),
            chi_count = sum(chi_above)) %>%
  ungroup()

#select top 4 based on counts
top4_LEP <- left_join(dif_counts_LEP, dif_summary_LEP) %>%
  arrange(p_count) %>%
  slice(1:4)


#estimate multiple group model with the anchor items fixed.
model_4mpt_LEP <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1,20,21,36, a1), (1,20,21,36, d)',
                           group = as.character(ask23mcdesc$LEP_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


#activate a cluster for parallel computation
mirtCluster()

#perform DIF testing
dif_4mpt_LEP <- DIF(model_4mpt_LEP, which.par = ("d"),
            technical = list(NCYCLES = 5000))

#deactivate cluster
mirtCluster(remove = TRUE)
# save(dif_4mpt_LEP, file = "LEP_DIF_Results.RData")

# load("LEP_DIF_Results.RData")

difdf_4mpt_LEP<- data.frame(unlist(dif_4mpt_LEP)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value) %>%
  select(Q, p2) %>%
  filter(p2 < .05)

LEP_2pl_final_mpt <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-5, 7-24, 26-41, a1),
                           (1-5, 7-24, 26-41, d)',
                           group = as.character(ask23mcdesc$LEP_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

LEP_gr1_mpt <- data.frame(coef(LEP_2pl_final_mpt, simplify = TRUE)$`0`$items)[,1:2]
names(LEP_gr1_mpt) <- c("a_noLEP", "d_noLEP")

LEP_gr2_mpt <- data.frame(coef(LEP_2pl_final_mpt, simplify = TRUE)$`1`$items)[,1:2]
names(LEP_gr2_mpt) <- c("a_LEP", "d_LEP")

LEP_dif_items_mpt <- bind_cols(LEP_gr1_mpt, LEP_gr2_mpt) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(6,25)) %>%
  mutate(b_noLEP = -d_noLEP/a_noLEP,
         b_LEP = -d_LEP/a_LEP) %>%
  select(Q, a_noLEP, a_LEP, b_noLEP, b_LEP)


LEP_dif_items_mpt_long <- LEP_dif_items_mpt %>%
  gather(key = "param", value = "value", c(a_noLEP, a_LEP, b_noLEP, b_LEP)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- LEP_dif_items_mpt_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

set.seed(05152018)
DTF(LEP_2pl_final_mpt, draws = 1000)

DTF(LEP_2pl_final_mpt, draws = 1000, plot = "func")

DTF(LEP_2pl_final_mpt, draws = 1000, plot = "sDTF")

mtt_sorted_LEP <- left_join(dif_counts_LEP, dif_summary_LEP) %>%
  arrange(desc(chi_count), mean_chi) %>%
  mutate(q_num = as.numeric(str_remove(Q, "Q")),
         q_num = ifelse(q_num < 40, q_num, q_num -1))

mirtCluster()
for(j in 1:40) {
model_mtt_LEP <- multipleGroup(ask23mcdesc[,c(2:39,41:43)],
                            paste0('F = 1-41\nCONSTRAINB = (',
                                   paste(mtt_sorted_LEP$q_num[(1:j)],
                                         collapse = ","),' , a1), (',
                                   paste(mtt_sorted_LEP$q_num[(1:j)],
                                         collapse = ","),', d)'),
                            group = as.character(ask23mcdesc$LEP_RECODE),
                            invariance = c("free_means", "free_var"),
                            SE = TRUE,
                            technical = list(NCYCLES = 5000))

dif_mtt_LEP <- DIF(model_mtt_LEP, which.par = ("d"),
                 technical = list(NCYCLES = 5000),
                 items2test = c(1:42)[-(mtt_sorted_LEP$q_num[(1:j)])],
                 verbose = FALSE)


difdf_mtt_LEP<- data.frame(unlist(dif_mtt_LEP)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

undif_LEP <- difdf_mtt_LEP %>%
  select(Q, p2) %>%
  filter(p2 > .05)

if(j >= nrow(undif_LEP)) break()
}
mirtCluster(remove = TRUE)
save(difdf_mtt_LEP, file = "LEP_mtt_results.RData")

load("LEP_mtt_results.RData")
mtt_dif_LEP <- filter(difdf_mtt_LEP, p2 < .05) %>%
  select(Q, p = p2)

LEP_2pl_final_mtt <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-5,7, 9-24, 26, 28-33, 35-41, a1),
                           (1-5,7, 9-24, 26, 28-33, 35-41, d)',
                           group = as.character(ask23mcdesc$LEP_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

LEP_gr1_mtt <- data.frame(coef(LEP_2pl_final_mtt, simplify = TRUE)$`0`$items)[,1:2]
names(LEP_gr1_mtt) <- c("a_noLEP", "d_noLEP")

LEP_gr2_mtt <- data.frame(coef(LEP_2pl_final_mtt, simplify = TRUE)$`1`$items)[,1:2]
names(LEP_gr2_mtt) <- c("a_LEP", "d_LEP")

LEP_dif_items <- bind_cols(LEP_gr1_mtt, LEP_gr2_mtt) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(6,8,25,27,34)) %>%
  mutate(b_noLEP = -d_noLEP/a_noLEP,
         b_LEP = -d_LEP/a_LEP) %>%
  select(Q, a_noLEP, a_LEP, b_noLEP, b_LEP)


LEP_dif_items_long <- LEP_dif_items %>%
  gather(key = "param", value = "value", c(a_noLEP, a_LEP, b_noLEP, b_LEP)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- LEP_dif_items_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

set.seed(05152018)
DTF(LEP_2pl_final_mtt, draws = 1000)

DTF(LEP_2pl_final_mtt, draws = 1000, plot = "func")

DTF(LEP_2pl_final_mtt, draws = 1000, plot = "sDTF")

expert_2pl_LEP <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                               'F = 1-41
                           CONSTRAINB = (2, 4, 7, 9, 14, 15, 22, 29, 31, a1), (2, 4, 7, 9, 14, 15, 22, 2
                               group = as.character(ask23mcdesc$LEP_RECODE),
                               invariance = c("free_means", "free_var"),
                               SE = TRUE,
                               verbose = FALSE,
                               technical = list(NCYCLES = 5000))

mirtCluster()
expert_dif_LEP <- DIF(expert_2pl_LEP, which.par = "d")
mirtCluster(remove = TRUE)
# save(expert_dif_LEP, file = "expert_dif_LEP.RData")

# load("expert_dif_LEP.RData")
difdf_expert_LEP<- data.frame(unlist(expert_dif_LEP)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

dif_only_expert_LEP <- filter(difdf_expert_LEP, p2 < .05) %>%
  select(Q, p = p2)

LEP_2pl_final_expert <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-9, 11-24, 26, 28-36, 39-41, a1),
                           (1-9, 11-24, 26, 28-36, 39-41, d)',
                           group = as.character(ask23mcdesc$LEP_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


LEP_gr1_expert <- data.frame(coef(LEP_2pl_final_expert, simplify = TRUE)$`0`$items)[,1:2]
names(LEP_gr1_expert) <- c("a_noLEP", "d_noLEP")

LEP_gr2_expert <- data.frame(coef(LEP_2pl_final_expert, simplify = TRUE)$`1`$items)[,1:2]
names(LEP_gr2_expert) <- c("a_LEP", "d_LEP")

LEP_dif_items_expert <- bind_cols(LEP_gr1_expert, LEP_gr2_expert) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(10,25,27,37,38)) %>%
  mutate(b_noLEP = -d_noLEP/a_noLEP,
         b_LEP = -d_LEP/a_LEP) %>%
  select(Q, a_noLEP, a_LEP, b_noLEP, b_LEP)


set.seed(05212018)
DTF(LEP_2pl_final_expert, draws = 1000)

DTF(LEP_2pl_final_expert, draws = 1000, plot = "func", theta_lim = c(-2, 2))

DTF(LEP_2pl_final_expert, draws = 1000, plot = "sDTF")

ask23mcdesc <- ask23mcdesc %>%
  mutate_at(vars(starts_with("RE_")), function(x) ifelse(is.na(x) | x == 0, 0,1)) %>%
  mutate(race2 = ifelse(RE_HIS == 1, "Hispanic",
                        ifelse(RE_BK == 1, "Black",
                               ifelse(RE_WHT == 1, "White", NA))))

mirtCluster()
racedif_vals2 <- plyr::adply(anchoritem, 1, .fun = get_DIFs,
                             dat = ask23mcdesc[,c(2:39,41:43)],
                             difvect = ask23mcdesc$race2,
                             cycles = 20000)
mirtCluster(remove = TRUE)
save(racedif_vals2, file = "racedifresults2.Rdata")

load("racedifresults2.Rdata")

#get summary mean of p-values and chi-squares for each item
dif_summary_race <- racedif_vals2 %>%
  group_by(Q) %>%
  summarise(mean_p = mean(p2),
            mean_chi = mean(X22))

#determine the 0.5 * 41th mean for the p-values and chi-squares
#(in this case the median)
dif_medians_race <- dif_summary_race%>%
  summarise(p_cuttoff = median(mean_p),
            chi_cutoff = median(mean_chi))

#determine the counts for the number of p-values and
#chi-squares below their respective cutoffs
dif_counts_race <- racedif_vals2 %>%
  mutate(p_above = p2 < dif_medians_race$p_cuttoff,
         chi_above = X22 < dif_medians_race$chi_cutoff) %>%
  group_by(Q) %>%
  summarise(p_count = sum(p_above),
            chi_count = sum(chi_above)) %>%
  ungroup()

#select top 4 based on counts
top4_race <- left_join(dif_counts_race, dif_summary_race) %>%
  arrange(p_count) %>%
  slice(1:4)


model_4mpt_race <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (11,23,31,35, a1), (11,23,31,35, d)',
                           group = as.character(ask23mcdesc$race2),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


#activate a cluster for parallel computation
mirtCluster()

#perform DIF testing
dif_4mpt_race <- DIF(model_4mpt_race, which.par = ("d"),
            technical = list(NCYCLES = 5000))

#deactivate cluster
mirtCluster(remove = TRUE)
# save(dif_4mpt_race, file = "race_DIF_Results.RData")

# load("race_DIF_Results.RData")

difdf_4mpt_race<- data.frame(unlist(dif_4mpt_race)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value) %>%
  select(Q, p2) %>%
  filter(p2 < .05)


race_2pl_final_mpt <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-8, 10-13, 15, 17-24, 27-33, 35-41, a1),
                           (1-8, 10-13, 15, 17-24, 27-33, 35-41, d)',
                           group = as.character(ask23mcdesc$race2),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

race_gr1_mpt <- data.frame(coef(race_2pl_final_mpt, simplify = TRUE)$`Black`$items)[,1:2]
names(race_gr1_mpt) <- c("a_black", "d_black")

race_gr2_mpt <- data.frame(coef(race_2pl_final_mpt, simplify = TRUE)$`Hispanic`$items)[,1:2]
names(race_gr2_mpt) <- c("a_hispanic", "d_hispanic")

race_gr3_mpt <- data.frame(coef(race_2pl_final_mpt, simplify = TRUE)$`White`$items)[,1:2]
names(race_gr3_mpt) <- c("a_white", "d_white")

race_dif_items_mpt <- bind_cols(race_gr1_mpt, race_gr2_mpt, race_gr3_mpt) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(9,14,16,25,26)) %>%
  mutate(b_black = -d_black/a_black,
         b_hispanic = -d_hispanic/a_hispanic,
         b_white = -d_white/a_white) %>%
  select(Q, a_black, a_hispanic, a_white, b_black, b_hispanic, b_white)


race_dif_items_mpt_long <- race_dif_items_mpt %>%
  gather(key = "param", value = "value", c(a_black, a_hispanic, a_white, b_black, b_hispanic, b_white))
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- race_dif_items_mpt_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

mtt_sorted_race <- left_join(dif_counts_race, dif_summary_race) %>%
  arrange(desc(chi_count), mean_chi) %>%
  mutate(q_num = as.numeric(str_remove(Q, "Q")),
         q_num = ifelse(q_num < 40, q_num, q_num -1))

mirtCluster()
for(j in 1:40) {
model_mtt_race <- multipleGroup(ask23mcdesc[,c(2:39,41:43)],
                            paste0('F = 1-41\nCONSTRAINB = (',
                                   paste(mtt_sorted_race$q_num[(1:j)],
                                         collapse = ","),' , a1), (',
                                   paste(mtt_sorted_race$q_num[(1:j)],
                                         collapse = ","),', d)'),
                            group = as.character(ask23mcdesc$race2),
                            invariance = c("free_means", "free_var"),
                            SE = TRUE,
                            technical = list(NCYCLES = 5000))

dif_mtt_race <- DIF(model_mtt_race, which.par = ("d"),
                 technical = list(NCYCLES = 5000),
                 items2test = c(1:42)[-(mtt_sorted_race$q_num[(1:j)])],
                 verbose = FALSE)


difdf_mtt_race<- data.frame(unlist(dif_mtt_race)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

undif_race <- difdf_mtt_race %>%
  select(Q, p2) %>%
  filter(p2 > .05)

if(j >= nrow(undif_race)) break()
}
mirtCluster(remove = TRUE)
# save(difdf_mtt_race, file = "race_mtt_results.RData")

# load("race_mtt_results.RData")
mtt_dif_race <- filter(difdf_mtt_race, p2 < .05) %>%
  select(Q, p = p2)

race_2pl_final_mtt <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (1-3,5-8, 10-12, 15, 17-24, 27-33, 35-37, 39-41, a1),
                           (1-3,5-8, 10-12, 15, 17-24, 27-33, 35-37, 39-41, d)',
                           group = as.character(ask23mcdesc$race2),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

race_gr1_mtt <- data.frame(coef(race_2pl_final_mtt, simplify = TRUE)$`Black`$items)[,1:2]
```

```
names(race_gr1_mtt) <- c("a_black", "d_black")

race_gr2_mtt <- data.frame(coef(race_2pl_final_mtt, simplify = TRUE)$`Hispanic`$items)[,1:2]
names(race_gr2_mtt) <- c("a_hispanic", "d_hispanic")

race_gr3_mtt <- data.frame(coef(race_2pl_final_mtt, simplify = TRUE)$`White`$items)[,1:2]
names(race_gr3_mtt) <- c("a_white", "d_white")

race_dif_items_mtt <- bind_cols(race_gr1_mtt, race_gr2_mtt, race_gr3_mtt) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(4, 9, 13, 14, 16, 25, 26, 34, 38)) %>%
  mutate(b_black = -d_black/a_black,
         b_hispanic = -d_hispanic/a_hispanic,
         b_white = -d_white/a_white) %>%
  select(Q, a_black, a_hispanic, a_white, b_black, b_hispanic, b_white)


race_dif_items_long_mtt <- race_dif_items_mtt %>%
  gather(key = "param", value = "value", c(a_black, a_hispanic, a_white, b_black, b_hispanic, b_white))
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value) %>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- race_dif_items_long_mtt %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

expert_2pl_race <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                               'F = 1-41
                           CONSTRAINB = (2, 7, 15, 22, 29, 31, a1), (2, 7,  15, 22, 29, 31, d)',
                               group = as.character(ask23mcdesc$race2),
                               invariance = c("free_means", "free_var"),
                               SE = TRUE,
                               verbose = FALSE,
                               technical = list(NCYCLES = 5000))

mirtCluster()
expert_dif_race <- DIF(expert_2pl_race, which.par = "d")
mirtCluster(remove = TRUE)
# save(expert_dif_race, file = "expert_dif_race.RData")

# load("expert_dif_race.RData")
difdf_expert_race<- data.frame(unlist(expert_dif_race)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

dif_only_expert_race <- filter(difdf_expert_race, p2 < .05) %>%
  select(Q, p = p2)

race_2pl_final_expert <- multipleGroup(ask23_postmc[,c(2:39,41:43)],
                           'F = 1-41
                           CONSTRAINB = (2, 5-8, 10-12, 16-22, 28-37, 39-41, a1),
                           (2, 5-8, 10-12, 16-22, 28-37, 39-41, d)',
                           group = as.character(ask23mcdesc$race2),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


race_gr1_expert <- data.frame(coef(race_2pl_final_expert, simplify = TRUE)$`Black`$items)[,1:2]
names(race_gr1_expert) <- c("a_black", "d_black")

race_gr2_expert <- data.frame(coef(race_2pl_final_expert, simplify = TRUE)$`Hispanic`$items)[,1:2]
names(race_gr2_expert) <- c("a_hispanic", "d_hispanic")

race_gr3_expert <- data.frame(coef(race_2pl_final_expert, simplify = TRUE)$`White`$items)[,1:2]
names(race_gr3_expert) <- c("a_white", "d_white")

race_dif_items_expert <- bind_cols(race_gr1_expert, race_gr2_expert, race_gr3_expert) %>%
  mutate(Q = paste0("Q", c(1:38, 40:42))) %>%
  slice(c(1, 3, 4, 9, 13, 14, 23, 24, 25, 26, 27, 38)) %>%
  mutate(b_black = -d_black/a_black,
         b_hispanic = -d_hispanic/a_hispanic,
         b_white = -d_white/a_white) %>%
  select(Q, a_black, a_hispanic, a_white, b_black, b_hispanic, b_white)
```

DIF - Reading Exam

For the purposes of the DIF analysis, I'll be using the 2PL model as discussed previously. Expert anchors were not solicited for this exam.

```
#code the descriptive scorrectly for this analysis.
ask23rcdesc <- left_join(ask23_postrc, descriptives) %>%
  mutate(LEP_RECODE = ifelse(LEP == 0, 0, 1),
         FRL_RECODE = ifelse(FRL == 0, 0, 1)) %>%
  mutate_at(vars(starts_with("RE_")), function(x) ifelse(is.na(x) | x == 0, 0,1)) %>%
  rowwise() %>%
  mutate(total_race = sum(c(RE_HIS, RE_AI, RE_AS, RE_BK, RE_PI, RE_WHT, RE_MX))) %>%
  ungroup() %>%
  mutate(race2 = ifelse(RE_HIS == 1, "Hispanic",
                        ifelse(RE_BK == 1, "Black",
                               ifelse(RE_WHT == 1, "White", NA))))
```

```r
# a function to gather all the test statistics and p-values for the test
# given a single anchor item
get_DIFs_reading <- function(dat, difvect, anchoritem, cycles) {

  twopl <- mirt(dat, 1, verbose = FALSE)

  if(typeof(difvect) != "character") difvect <- as.character(difvect)

  modelmult <- multipleGroup(dat,
                             paste0('F = 1-21\nCONSTRAINB = (', anchoritem,' , a1), (', anchoritem,', d
                             group = difvect,
                             invariance = c("free_means", "free_var"),
                             SE = TRUE,
                             technical = list(NCYCLES = cycles))

  diff <- DIF(modelmult, which.par = ("d"),
              technical = list(NCYCLES = cycles),
              items2test = (1:21)[-anchoritem],
              verbose = FALSE, warn = FALSE, message = FALSE)

  diffdf <- data.frame(unlist(diff)) %>%
    mutate(var = row.names(.)) %>%
    rename("value" = !!names(.[1])) %>%
    separate(var, into = c("Q", "stat")) %>%
    spread(key = stat, value = value) %>%
    select(Q, p2, X22)

  return(diffdf)
}

mirtCluster()
anchoritem <- 1:21

reading_FRL_dif_vals <- plyr::adply(anchoritem, 1, .fun = get_DIFs_reading,
                                    dat = ask23rcdesc[,c(2:22)],
                                    difvect = ask23rcdesc$FRL_RECODE,
                                    cycles = 5000)

save(reading_FRL_dif_vals, file = "readingFRLdif.Rdata")

mirtCluster(remove = TRUE)

load("readingFRLdif.Rdata")

#get summary mean of p-values and chi-squares for each item
reading_dif_summary_FRL <- reading_FRL_dif_vals %>%
  group_by(Q) %>%
  summarise(mean_p = mean(p2),
            mean_chi = mean(X22))

#determine the 0.5 * 41th mean for the p-values and chi-squares
#(in this case the median)
```

```r
readig_dif_medians_FRL <- reading_dif_summary_FRL%>%
  summarise(p_cuttoff = median(mean_p),
            chi_cutoff = median(mean_chi))

#determine the counts for the number of p-values and
#chi-squares below their respective cutoffs
reading_dif_counts_FRL <- reading_FRL_dif_vals %>%
  mutate(p_above = p2 < dif_medians_FRL$p_cuttoff,
         chi_above = X22 < dif_medians_FRL$chi_cutoff) %>%
  group_by(Q) %>%
  summarise(p_count = sum(p_above),
            chi_count = sum(chi_above)) %>%
  ungroup()

#select top 4 based on counts
top4 <- left_join(reading_dif_counts_FRL, reading_dif_summary_FRL) %>%
  arrange(p_count) %>%
  slice(1:4)

kable(top4)

#estimate multiple group model wit the items fixed.
reading_model_4mpt_FRL <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (1,2,9,19, a1), (1,2,9,19, d)',
                           group = as.character(ask23rcdesc$FRL_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


#activate a cluster for parallel computation
mirtCluster()

#perform DIF testing
reading_dif_4mpt_FRL <- DIF(reading_model_4mpt_FRL, which.par = ("d"),
            technical = list(NCYCLES = 5000))

#deactivate cluster
mirtCluster(remove = TRUE)
# save(reading_dif_4mpt_FRL, file = "reading_FRL_DIF_Results.RData")

# load("reading_FRL_DIF_Results.RData")

reading_difdf_4mpt_FRL<- data.frame(unlist(reading_dif_4mpt_FRL)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value) %>%
  select(Q, p2) %>%
  filter(p2 < .05)
```

```
reading_FRL_2pl_final <- multipleGroup(ask23_postrc[,c(2:22)],
                          'F = 1-21
                          CONSTRAINB = (1-5, 9-11, 13-21, a1),
                          (1-5, 9-11, 13-21, d)',
                          group = as.character(ask23rcdesc$FRL_RECODE),
                          invariance = c("free_means", "free_var"),
                          SE = TRUE,
                          verbose = FALSE,
                          technical = list(NCYCLES = 5000))

reading_FRL_gr1 <- data.frame(coef(reading_FRL_2pl_final, simplify = TRUE)$`0`$items)[,1:2]
names(reading_FRL_gr1) <- c("a_noFRL", "d_noFRL")

reading_FRL_gr2 <- data.frame(coef(reading_FRL_2pl_final, simplify = TRUE)$`1`$items)[,1:2]
names(reading_FRL_gr2) <- c("a_FRL", "d_FRL")

reading_FRL_dif_items <- bind_cols(reading_FRL_gr1, reading_FRL_gr2) %>%
  mutate(Q = paste0("Q", c(43:63))) %>%
  slice(c(6,7,8,12)) %>%
  mutate(b_noFRL = -d_noFRL/a_noFRL,
         b_FRL = -d_FRL/a_FRL) %>%
  select(Q, a_noFRL, a_FRL, b_noFRL, b_FRL)


reading_FRL_dif_items_long <- reading_FRL_dif_items %>%
  gather(key = "param", value = "value", c(a_noFRL, a_FRL, b_noFRL, b_FRL)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- reading_FRL_dif_items_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

set.seed(05232018)
DTF(reading_FRL_2pl_final, draws = 1000)

DTF(reading_FRL_2pl_final, draws = 1000, plot = "func")

DTF(reading_FRL_2pl_final, draws = 1000, plot = "sDTF")

reading_mtt_sorted_FRL <- left_join(reading_dif_counts_FRL, reading_dif_summary_FRL) %>%
  arrange(desc(chi_count), mean_chi) %>%
  mutate(q_num = as.numeric(str_remove(Q, "Q"))-42)

mirtCluster()
for(j in 1:21) {
reading_model_mtt_FRL <- multipleGroup(ask23rcdesc[,2:22],
                            paste0('F = 1-21\nCONSTRAINB = (',
                                   paste(reading_mtt_sorted_FRL$q_num[(1:j)],
                                         collapse = ","),' , a1), (',
                                   paste(reading_mtt_sorted_FRL$q_num[(1:j)],
                                         collapse = ","),', d)'),
                            group = as.character(ask23rcdesc$FRL_RECODE),
                            invariance = c("free_means", "free_var"),
                            SE = TRUE,
                            technical = list(NCYCLES = 5000))

reading_dif_mtt_FRL <- DIF(reading_model_mtt_FRL, which.par = ("d"),
                 technical = list(NCYCLES = 5000),
                 verbose = FALSE)

reading_difdf_mtt_FRL<- data.frame(unlist(reading_dif_mtt_FRL)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

reading_undif_FRL <- reading_difdf_mtt_FRL %>%
  select(Q, p2) %>%
  filter(p2 > .05)

if(j >= nrow(reading_undif_FRL)) break()
}
mirtCluster(remove = TRUE)
# save(reading_difdf_mtt_FRL, file = "reading_FRL_mtt_results.RData")

# load("reading_FRL_mtt_results.RData")
reading_mtt_dif_FRL <- filter(reading_difdf_mtt_FRL, p2 < .05) %>%
  select(Q, p = p2)

reading_FRL_2pl_final_mtt <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (1-11, 13-21, a1),
                           (1-11, 13-21, d)',
                           group = as.character(ask23rcdesc$FRL_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

reading_FRL_gr1 <- data.frame(coef(reading_FRL_2pl_final_mtt, simplify = TRUE)$`0`$items)[,1:2]
names(reading_FRL_gr1) <- c("a_noFRL", "d_noFRL")

reading_FRL_gr2 <- data.frame(coef(reading_FRL_2pl_final_mtt, simplify = TRUE)$`1`$items)[,1:2]
names(reading_FRL_gr2) <- c("a_FRL", "d_FRL")
```

```
reading_FRL_dif_items <- bind_cols(reading_FRL_gr1, reading_FRL_gr2) %>%
  mutate(Q = paste0("Q", 43:63)) %>%
  slice(c(12)) %>%
  mutate(b_noFRL = -d_noFRL/a_noFRL,
         b_FRL = -d_FRL/a_FRL) %>%
  select(Q, a_noFRL, a_FRL, b_noFRL, b_FRL)

reading_FRL_dif_items_long <- reading_FRL_dif_items %>%
  gather(key = "param", value = "value", c(a_noFRL, a_FRL, b_noFRL, b_FRL)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- reading_FRL_dif_items_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

set.seed(05242018)
DTF(reading_FRL_2pl_final_mtt, draws = 1000)

DTF(reading_FRL_2pl_final_mtt, draws = 1000, plot = "func")

DTF(reading_FRL_2pl_final_mtt, draws = 1000, plot = "sDTF")

mirtCluster()
reading_LEP_dif_vals <- plyr::adply(anchoritem, 1, .fun = get_DIFs_reading,
                                    dat = ask23rcdesc[,c(2:22)],
                                    difvect = ask23rcdesc$LEP_RECODE,
                                    cycles = 20000)

save(reading_LEP_dif_vals, file = "readingLEPdif.Rdata")
mirtCluster(remove = TRUE)

load("readingLEPdif.Rdata")

#get summary mean of p-values and chi-squares for each item
reading_dif_summary_LEP <- reading_LEP_dif_vals %>%
  group_by(Q) %>%
  summarise(mean_p = mean(p2),
            mean_chi = mean(X22))

#determine the 0.5 * 41th mean for the p-values and chi-squares
#(in this case the median)
```

```r
reading_dif_medians_LEP <- reading_dif_summary_LEP%>%
  summarise(p_cuttoff = median(mean_p),
            chi_cutoff = median(mean_chi))

#determine the counts for the number of p-values and
#chi-squares below their respective cutoffs
reading_dif_counts_LEP <- reading_LEP_dif_vals %>%
  mutate(p_above = p2 < dif_medians_LEP$p_cuttoff,
         chi_above = X22 < dif_medians_LEP$chi_cutoff) %>%
  group_by(Q) %>%
  summarise(p_count = sum(p_above),
            chi_count = sum(chi_above)) %>%
  ungroup()

#select top 4 based on counts
reading_top4_LEP <- left_join(reading_dif_counts_LEP, reading_dif_summary_LEP) %>%
  arrange(p_count, desc(mean_p)) %>%
  slice(1:4)


#estimate multiple group model wit the items fixed.
reading_model_4mpt_LEP <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (6,11,17,21, a1), (6,11,17,21, d)',
                           group = as.character(ask23rcdesc$LEP_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


#activate a cluster for parallel computation
mirtCluster()

#perform DIF testing
reading_dif_4mpt_LEP <- DIF(reading_model_4mpt_LEP, which.par = ("d"),
            technical = list(NCYCLES = 5000))

#deactivate cluster
mirtCluster(remove = TRUE)
# save(reading_dif_4mpt_LEP, file = "reading_LEP_DIF_Results.RData")

# load("reading_LEP_DIF_Results.RData")

reading_difdf_4mpt_LEP<- data.frame(unlist(reading_dif_4mpt_LEP)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value) %>%
  select(Q, p2) %>%
  filter(p2 < .05)

kable(reading_difdf_4mpt_LEP, row.names = FALSE)
```

```
reading_LEP_2pl_final_mpt <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (1-17, 19-21, a1),
                           (1-17, 19-21, d)',
                           group = as.character(ask23rcdesc$LEP_RECODE),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

reading_LEP_gr1_mpt <- data.frame(coef(reading_LEP_2pl_final_mpt, simplify = TRUE)$`0`$items)[,1:2]
names(reading_LEP_gr1_mpt) <- c("a_noLEP", "d_noLEP")

reading_LEP_gr2_mpt <- data.frame(coef(reading_LEP_2pl_final_mpt, simplify = TRUE)$`1`$items)[,1:2]
names(reading_LEP_gr2_mpt) <- c("a_LEP", "d_LEP")

reading_LEP_dif_items_mpt <- bind_cols(reading_LEP_gr1_mpt, reading_LEP_gr2_mpt) %>%
  mutate(Q = paste0("Q", 43:63)) %>%
  slice(c(18)) %>%
  mutate(b_noLEP = -d_noLEP/a_noLEP,
         b_LEP = -d_LEP/a_LEP) %>%
  select(Q, a_noLEP, a_LEP, b_noLEP, b_LEP)


reading_LEP_dif_items_mpt_long <- reading_LEP_dif_items_mpt %>%
  gather(key = "param", value = "value", c(a_noLEP, a_LEP, b_noLEP, b_LEP)) %>%
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- reading_LEP_dif_items_mpt_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

set.seed(05252018)
DTF(reading_LEP_2pl_final_mpt, draws = 1000)

DTF(reading_LEP_2pl_final_mpt, draws = 1000, plot = "func")

DTF(LEP_2pl_final_mpt, draws = 1000, plot = "sDTF")

reading_mtt_sorted_LEP <- left_join(reading_dif_counts_LEP, reading_dif_summary_LEP) %>%
  arrange(desc(chi_count), mean_chi) %>%
  mutate(q_num = as.numeric(str_remove(Q, "Q"))-42)

mirtCluster()
for(j in 1:21) {
reading_model_mtt_LEP <- multipleGroup(ask23rcdesc[,2:22],
                            paste0('F = 1-21\nCONSTRAINB = (',
                                   paste(reading_mtt_sorted_LEP$q_num[(1:j)],
                                         collapse = ","),' , a1), (',
                                   paste(reading_mtt_sorted_LEP$q_num[(1:j)],
                                         collapse = ","),', d)'),
                            group = as.character(ask23rcdesc$LEP_RECODE),
                            invariance = c("free_means", "free_var"),
                            SE = TRUE,
                            technical = list(NCYCLES = 5000))

reading_dif_mtt_LEP <- DIF(reading_model_mtt_LEP, which.par = ("d"),
                 technical = list(NCYCLES = 5000),
                 verbose = FALSE)


reading_difdf_mtt_LEP<- data.frame(unlist(reading_dif_mtt_LEP)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

reading_undif_LEP <- reading_difdf_mtt_LEP %>%
  select(Q, p2) %>%
  filter(p2 > .05)

if(j >= nrow(reading_undif_LEP)) break()
}
mirtCluster(remove = TRUE)
# save(reading_difdf_mtt_LEP, file = "reading_LEP_mtt_results.RData")

# load("reading_LEP_mtt_results.RData")
reading_mtt_dif_LEP <- filter(reading_difdf_mtt_LEP, p2 < .05) %>%
  select(Q, p = p2)
kable(reading_mtt_dif_LEP[c(4:5,1:3),], row.names = FALSE)

mirtCluster()
reading_race_dif_vals <- plyr::adply(anchoritem, 1, .fun = get_DIFs_reading,
                                     dat = ask23rcdesc[,c(2:22)],
                                     difvect = ask23rcdesc$race2,
                                     cycles = 20000)

save(reading_race_dif_vals, file = "readingracedif.Rdata")
mirtCluster(remove = TRUE)

load("readingracedif.Rdata")

#get summary mean of p-values and chi-squares for each item
reading_dif_summary_race <- reading_race_dif_vals %>%
  group_by(Q) %>%
  summarise(mean_p = mean(p2),
            mean_chi = mean(X22))

#determine the 0.5 * 41th mean for the p-values and chi-squares
#(in this case the median)
reading_dif_medians_race <- reading_dif_summary_race%>%
  summarise(p_cuttoff = median(mean_p),
            chi_cutoff = median(mean_chi))

#determine the counts for the number of p-values and
#chi-squares below their respective cutoffs
reading_dif_counts_race <- reading_race_dif_vals %>%
  mutate(p_above = p2 < dif_medians_race$p_cuttoff,
         chi_above = X22 < dif_medians_race$chi_cutoff) %>%
  group_by(Q) %>%
  summarise(p_count = sum(p_above),
            chi_count = sum(chi_above)) %>%
  ungroup()

#select top 4 based on counts
reading_top4_race <- left_join(reading_dif_counts_race, reading_dif_summary_race) %>%
  arrange(p_count, desc(mean_p)) %>%
  slice(1:5)


#estimate multiple group model with the items fixed.
reading_model_4mpt_race <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (5,6,7,11, a1), (5,6,7,11, d)',
                           group = as.character(ask23rcdesc$race2),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))


#activate a cluster for parallel computation
mirtCluster()

#perform DIF testing
reading_dif_4mpt_race <- DIF(reading_model_4mpt_race, which.par = ("d"),
            technical = list(NCYCLES = 5000))

#deactivate cluster
mirtCluster(remove = TRUE)
# save(reading_dif_4mpt_race, file = "reading_race_DIF_Results.RData")

# load("reading_race_DIF_Results.RData")

reading_difdf_4mpt_race<- data.frame(unlist(reading_dif_4mpt_race)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value) %>%
  select(Q, p2) %>%
  filter(p2 < .05)


reading_race_2pl_final_mpt <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (1-11, 13-20, a1),
                           (1-11, 13-20, d)',
                           group = as.character(ask23rcdesc$race2),
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

reading_race_gr1_mpt <- data.frame(coef(reading_race_2pl_final_mpt,
                                        simplify = TRUE)$`Black`$items)[,1:2]
names(reading_race_gr1_mpt) <- c("a_black", "d_black")

reading_race_gr2_mpt <- data.frame(coef(reading_race_2pl_final_mpt,
                                        simplify = TRUE)$`Hispanic`$items)[,1:2]
names(reading_race_gr2_mpt) <- c("a_hispanic", "d_hispanic")

reading_race_gr3_mpt <- data.frame(coef(reading_race_2pl_final_mpt,
                                        simplify = TRUE)$`White`$items)[,1:2]
names(reading_race_gr3_mpt) <- c("a_white", "d_white")

reading_race_dif_items_mpt <- bind_cols(reading_race_gr1_mpt,
                                        reading_race_gr2_mpt,
                                        reading_race_gr3_mpt) %>%
  mutate(Q = paste0("Q", 43:63)) %>%
  slice(c(12,21)) %>%
  mutate(b_black = -d_black/a_black,
         b_hispanic = -d_hispanic/a_hispanic,
         b_white = -d_white/a_white) %>%
  select(Q, a_black, a_hispanic, a_white, b_black, b_hispanic, b_white)


reading_race_dif_items_mpt_long <- reading_race_dif_items_mpt %>%
  gather(key = "param", value = "value", c(a_black, a_hispanic, a_white, b_black, b_hispanic, b_white))
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value)%>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- reading_race_dif_items_mpt_long %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })
```

```
ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)

#excluding question 59 and 63 as anchors
reading_mtt_sorted_race <- left_join(reading_dif_counts_race, reading_dif_summary_race) %>%
  arrange(desc(chi_count), mean_chi) %>%
  filter(!(Q %in% c("Q59", "Q63"))) %>%
  mutate(q_num = as.numeric(str_remove(Q, "Q"))-42)

mirtCluster()
for(j in 1:19) {
reading_model_mtt_race <- multipleGroup(ask23rcdesc[,2:22],
                            paste0('F = 1-21\nCONSTRAINB = (',
                                   paste(reading_mtt_sorted_race$q_num[(1:j)],
                                         collapse = ","),' , a1), (',
                                   paste(reading_mtt_sorted_race$q_num[(1:j)],
                                         collapse = ","),', d)'),
                            group = ask23rcdesc$race2,
                            invariance = c("free_means", "free_var"),
                            SE = TRUE,
                            technical = list(NCYCLES = 5000))

reading_dif_mtt_race <- DIF(reading_model_mtt_race, which.par = ("d"),
                 technical = list(NCYCLES = 5000),
                 verbose = FALSE)


reading_difdf_mtt_race<- data.frame(unlist(reading_dif_mtt_race)) %>%
  mutate(var = row.names(.)) %>%
  rename("value" = !!names(.[1])) %>%
  separate(var, into = c("Q", "stat")) %>%
  spread(key = stat, value = value)

reading_undif_race <- reading_difdf_mtt_race %>%
  select(Q, p2) %>%
  filter(p2 > .05)

if(j >= nrow(reading_undif_race)) break()
}
mirtCluster(remove = TRUE)
save(reading_difdf_mtt_race, file = "reading_race_mtt_results.RData")
```

```
load("reading_race_mtt_results.RData")
reading_mtt_dif_race <- filter(reading_difdf_mtt_race, p2 < .05) %>%
  select(Q, p = p2)

reading_race_2pl_final_mtt <- multipleGroup(ask23_postrc[,2:22],
                           'F = 1-21
                           CONSTRAINB = (1-16, 18-20, a1),
                           (1-16, 18-20, d)',
                           group = as.character(ask23rcdesc$race2),
```

```r
                           invariance = c("free_means", "free_var"),
                           SE = TRUE,
                           verbose = FALSE,
                           technical = list(NCYCLES = 5000))

reading_race_gr1_mtt <- data.frame(coef(reading_race_2pl_final_mtt,
                                        simplify = TRUE)$`Black`$items)[,1:2]
names(reading_race_gr1_mtt) <- c("a_black", "d_black")

reading_race_gr2_mtt <- data.frame(coef(reading_race_2pl_final_mtt,
                                        simplify = TRUE)$`Hispanic`$items)[,1:2]
names(reading_race_gr2_mtt) <- c("a_hispanic", "d_hispanic")

reading_race_gr3_mtt <- data.frame(coef(reading_race_2pl_final_mtt,
                                        simplify = TRUE)$`White`$items)[,1:2]
names(reading_race_gr3_mtt) <- c("a_white", "d_white")

reading_race_dif_items_mtt <- bind_cols(reading_race_gr1_mtt,
                                        reading_race_gr2_mtt,
                                        reading_race_gr3_mtt) %>%
  mutate(Q = paste0("Q", 43:63)) %>%
  slice(c(17,21)) %>%
  mutate(b_black = -d_black/a_black,
         b_hispanic = -d_hispanic/a_hispanic,
         b_white = -d_white/a_white) %>%
  select(Q, a_black, a_hispanic, a_white, b_black, b_hispanic, b_white)


reading_race_dif_items_long_mtt <- reading_race_dif_items_mtt %>%
  gather(key = "param", value = "value", c(a_black, a_hispanic, a_white, b_black, b_hispanic, b_white))
  separate("param", c("param", "group")) %>%
  spread(key = param, value = value) %>%
  mutate(item_num = as.numeric(gsub("Q","",Q))) %>%
  arrange(item_num)

curves_icc_dif <- reading_race_dif_items_long_mtt %>%
  pmap(function(...){
    dots <- data_frame(...)
    stat_function(data = dots, aes(0, color = group),
                  fun = function(x) 1/(1  + exp(-dots$a * (x - dots$b))),
                  xlim = c(-3.5, 3.5))
  })

ggplot(data.frame(x = 0), aes(x = x)) +
  curves_icc_dif +
  labs(x = "Abiity", y = "Probability of Correct Response") +
  facet_wrap(~Q)
```

Appendix 2 - MPLUS code for a 2PL model for the multiple choice section.

I initially used MPLUS for the model estimation, but had some difficulties. Those difficulties increased when I hit the reading portion of the exam, so I moved to the `mirt` package in R. This package can also estimate a 2PL model, so I shifted to doing the whole analysis in MPLUS. This section contains MPLUS code for anyone interested in replication in that software.

I exported the dataset to an MPLUS native dataset using the `MplusAutomation` package. The following code is not actually executed in this document - if you do execute it in R, it will provide a simple data input skeleton for MPLUS. I wanted to use slightly different code so the automatic output from the function is not included.

```r
library(MplusAutomation) #useful functions for working in MPLUS
prepareMplusData(ask23_postmc, "Y2Y3 MPLUS/y23mcpost.dat")
```

At this point, I moved to MPLUS for IRT estimation. MPLUS makes the estimation of 2PL models with missing data very easy, so the fact that Y2 students are missing on a subset of the items is solved by simply using the “MLR” estimator. The following code produces a full-information 2PL IRT model.

```
TITLE: Your title goes here
DATA: FILE = "Y2Y3 MPLUS/y23mcpost.dat";
VARIABLE:
NAMES = SID Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q39 Q40 Q41 Q42;
MISSING=.;

USEVARIABLES = Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q39 Q40 Q41 Q42;

CATEGORICAL = Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q39 Q40 Q41 Q42;

ANALYSIS: ESTIMATOR IS MLR;

MODEL:
know_post by Q1* Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q39 Q40 Q41 Q42;

know_post@1;
```

The full output for this model can be seen in the “Y2Y3 Combined Multiple Choice All Items MPLUS Output” PDF document. Rather than clutter this document with a lot of output, I’ll highlight a few important things. The first is that MPLUS does not compute a chi-square for this model, saying that “…the frequency table for the categorical variables is too large.” This means that assessing model fit isn’t possible, and only model fit parameters that are produced are the AIC, BIC, and the sample-size adjusted BIC which are only useful for descriptive or comparative purposes. The second thing that is important to note is that Q39 has a negative discrimination parameter. Individuals whose ability estimates are higher are *less* likely to answer this item correctly. This question asks “When people study America’s character during the American Revolution, they are primarily interested in learning about:" (I do not have distractors for the items). Items with negative discrimination are not informative and should be discarded. So I re-ran the model with Q39 excluded.

```
TITLE: Your title goes here
DATA: FILE = "Y2Y3 MPLUS/y23mcpost.dat";
VARIABLE:
NAMES = SID Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q39 Q40 Q41 Q42;
MISSING=.;

USEVARIABLES = Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q40 Q41 Q42;

IDVARIABLE = SID;

CATEGORICAL = Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q40 Q41 Q42;

ANALYSIS: ESTIMATOR IS MLR;

MODEL:
know_post by Q1* Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
            Q20 Q21 Q22 Q23 Q24 Q25 Q26 Q27 Q28 Q29 Q30 Q31 Q32 Q33 Q34 Q35 Q36
            Q37 Q38 Q40 Q41 Q42;

know_post@1;

savedata:
   file is y2y3knowledgeposttest.dat;
   save is fscores;
```