

JavaScript Assessment

Code ▾

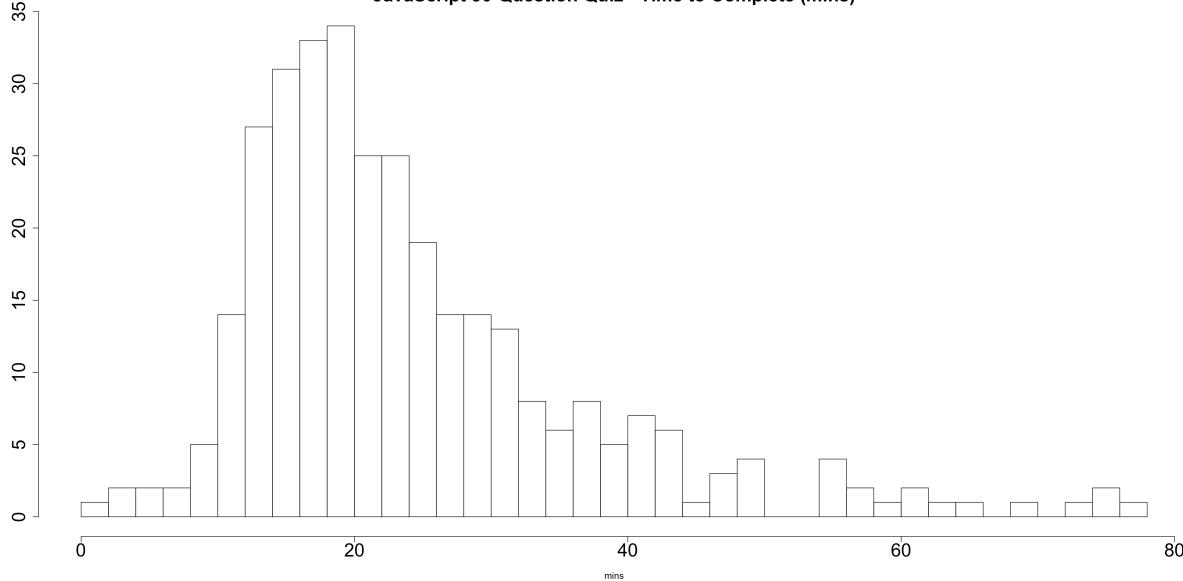
Calibration, Scoring, & Impact Analyses

Number of Questions Attempted

	1	2	3	4	5	6	7	8	10	11	12	13	14	16	18	21	30	NA
	6	3	2	3	5	2	1	2	1	1	2	1	1	2	1	1	325	0

Time spent on JavaScript Quiz

JavaScript 30-Question Quiz - Time to Complete (mins)



n	min	q1	avg	med	q3	max
325	1.5	15.68	24.42163	20.88	29.43	77.57

1 row

Time spent on Quiz, by Claim

Time spent on Quiz, by Question

Analytic Sample

Exclusion Criteria:

- test/fake accounts
- attempted all questions
- total time to complete > 15min (30s per question)

The proportion of candidates who completed the JavaScript Language Quiz of those who attempted it is 0.905.

The proportion of candidates who have a valid attempted is 0.713 of those who attempted the quiz.

Classical Test Theory Analysis

Item mean is an indicator of item difficulty. Items with extremely low or high means are indicators of poorly performing items as they are overly difficult or easy, respectively, for test takers. (parallels IRT item difficulty parameter)

Item-total correlation is the correlation between a score on a particular item and the performance on the rest of the test. High item-total correlations would indicate that test takers who score well on the overall test generally also performed well on the individual item. Low or negative item-total correlations are indicators of poor item performance as they suggest that test takers who score well on the overall test score lower on the individual item. This kind of item performance would suggest that the item may be measuring something different than the other items on the test or the item may be keyed in the wrong direction. (parallels IRT item discrimination parameter)

Alpha-deleted is a measure of the test reliability (internal consistency) - a measure of domain sampling and the impact of a flawed items if a particular item is removed. Alpha gives an estimate of average inter-item correlation among the items. An indicator of poor item performance is if once an item is removed then the overall test reliability is greatly improved.

Items are flagged to be reviewed or removed if:

- *Item means* are outside of the range 0.20 and 0.80 - indicating really easy or difficult items
- *Item-total correlations* are less than 0.20; the larger the better - indicating may be measuring something different from the rest of the test
- *Alpha-deleted increase* if the item is removed from the test - indicating may be measuring something different from the rest of the test or is adding random noise (unsystematic variation) to the overall test.

Overall JavaScript Reliability is: 0.712

df	blueprint\$question_id	flag	claim	name	target	n_candidates	item_mean	item_total_correlation	alpha
	1327 <=	Claim	lang_javascript_02	String Immutability	1	256	0.262	-0.005	
	1328	Claim	lang_javascript_03	Comparison Operators	1	256	0.578	0.253	
	1329 <=	Claim	lang_javascript_04	Function Declarations	1	256	0.887	0.202	
	1330	Claim	lang_javascript_05	Logical Operators	1	256	0.363	0.396	
	1339	Claim	lang_javascript_14	Functions	1	256	0.648	0.406	
	1341	Claim	lang_javascript_16	Class Methods	2	256	0.363	0.291	
	1342 <=	Claim	lang_javascript_17	Class Constructors	2	256	0.758	0.070	
	1343 <=	Claim	lang_javascript_18	Object Types	2	256	0.445	0.188	
	1353	Claim	lang_javascript_28	Encapsulation	2	256	0.520	0.333	
	1354 <=	Claim	lang_javascript_29	Getters and setters	2	256	0.332	0.175	
	1326 <=	Claim	lang_javascript_01	Async Programming / Promises	3	256	0.492	0.192	
	1332 <=	Claim	lang_javascript_07	APIs	3	256	0.988	0.011	
	1333 <=	Claim	lang_javascript_08	Browser - Event Handling	3	256	0.516	0.172	
	1337 <=	Claim	lang_javascript_12	Browser-stroage	3	256	0.816	0.302	
	1340	Claim	lang_javascript_15	Requests	3	256	0.574	0.287	
	1331	Claim	lang_javascript_06	Packages-Modules	4	256	0.371	0.332	
	1334 <=	Claim	lang_javascript_09	Common Libraries	4	256	0.699	0.123	
	1336	Claim	lang_javascript_11	Common libraries	4	256	0.645	0.317	
	1344 <=	Claim	lang_javascript_19	Common libraries	4	256	0.426	0.013	
	1345 <=	Claim	lang_javascript_20	Common libraries	4	256	0.199	0.202	
	1335	Claim	lang_javascript_10	Design	5	256	0.656	0.232	
	1346 <=	Claim	lang_javascript_21	Testing Frameworks	5	256	0.938	0.156	
	1348	Claim	lang_javascript_23	Popular Frameworks	5	256	0.406	0.260	
	1349	Claim	lang_javascript_24	Popular Frameworks	5	256	0.738	0.231	

df_blueprint\$question_id	flag	claim_name	target	n_candidates	item_mean	item_total_correlation	alpha
1350	Claim	lang_javascript_25	Popular Frameworks 5	256	0.270	0.316	
1338 <=	Claim	lang_javascript_13	Error Handling 6	256	0.832	0.272	
1347	Claim	lang_javascript_22	Testing 6	256	0.512	0.420	
1351 <=	Claim	lang_javascript_26	Error Handling 6	256	0.887	0.272	
1352	Claim	lang_javascript_27	Error Handling 6	256	0.281	0.253	
1355	Claim	lang_javascript_30	Git Hooks 6	256	0.699	0.217	

IRT Analysis

Unidimensional Model

IRT Item Parameters

	a	b	g	u
1330	1.256	0.606	0	1
1352	0.677	1.528	0	1
1343	0.422	0.566	0	1
1341	0.809	0.811	0	1
1334	0.360	-2.344	0	1
1351	1.025	-2.257	0	1
1346	0.599	-4.471	0	1
1328	0.640	-0.503	0	1
1338	0.925	-1.930	0	1
1333	0.421	-0.123	0	1
1326	0.467	0.099	0	1
1327	0.160	6.420	0	1
1335	0.685	-0.998	0	1
1332	0.284	-12.878	0	1
1329	0.651	-3.253	0	1
1350	0.841	1.367	0	1
1347	1.267	-0.021	0	1
1340	0.695	-0.443	0	1
1353	0.882	-0.074	0	1
1348	0.659	0.653	0	1
1342	0.256	-4.400	0	1
1349	0.523	-2.034	0	1
1336	0.748	-0.852	0	1
1344	0.161	1.867	0	1
1331	0.984	0.666	0	1
1345	0.513	2.830	0	1
1354	0.415	1.754	0	1
1355	0.534	-1.622	0	1
1339	1.261	-0.602	0	1
1337	0.843	-1.940	0	1

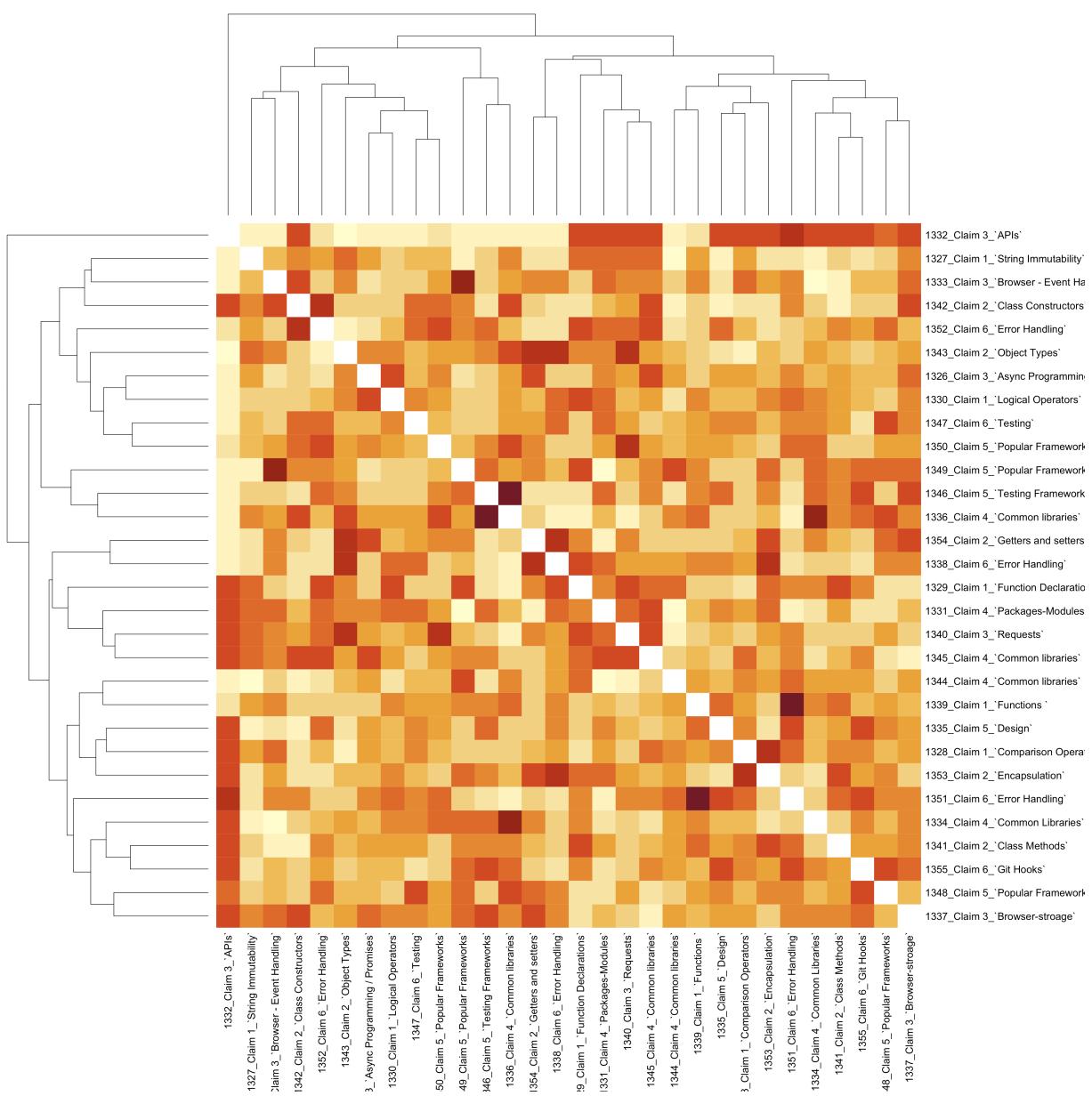
Item Fit Statistics (S-X2): p-val < 0.05 <=> generally indicates model misfit

claim	name	target	item	S_X2	df.S_X2	RMSEA.S_X2	p.S_X2	flag
Claim 1	lang_javascript_05	Logical Operators	1330	19.988	13	0.046	0.096	
Claim 6	lang_javascript_27	Error Handling	1352	20.852	14	0.044	0.105	
Claim 2	lang_javascript_18	Object Types	1343	5.201	14	0.000	0.983	
Claim 2	lang_javascript_16	Class Methods	1341	18.512	14	0.036	0.184	
Claim 4	lang_javascript_09	Common Libraries	1334	16.347	15	0.019	0.359	
Claim 6	lang_javascript_26	Error Handling	1351	9.709	11	0.000	0.557	
Claim 5	lang_javascript_21	Testing Frameworks	1346	9.217	8	0.024	0.324	
Claim 1	lang_javascript_03	Comparison Operators	1328	5.180	14	0.000	0.983	
Claim 6	lang_javascript_13	Error Handling	1338	12.795	12	0.016	0.384	
Claim 3	lang_javascript_08	Browser - Event Handling	1333	8.177	15	0.000	0.916	
Claim 3	lang_javascript_01	Async Programming / Promises	1326	9.432	15	0.000	0.854	
Claim 1	lang_javascript_02	String Immutability	1327	12.699	15	0.000	0.626	
Claim 5	lang_javascript_10	Design	1335	8.696	14	0.000	0.850	
Claim 3	lang_javascript_07	APIs	1332	2.925	3	0.000	0.403	
Claim 1	lang_javascript_04	Function Declarations	1329	9.924	12	0.000	0.623	
Claim 5	lang_javascript_25	Popular Frameworks	1350	12.702	13	0.000	0.471	
Claim 6	lang_javascript_22	Testing	1347	6.760	13	0.000	0.914	
Claim 3	lang_javascript_15	Requests	1340	17.806	13	0.038	0.165	
Claim 2	lang_javascript_28	Encapsulation	1353	11.151	13	0.000	0.598	
Claim 5	lang_javascript_23	Popular Frameworks	1348	17.077	14	0.029	0.252	
Claim 2	lang_javascript_17	Class Constructors	1342	3.355	15	0.000	0.999	
Claim 5	lang_javascript_24	Popular Frameworks	1349	15.779	15	0.014	0.397	
Claim 4	lang_javascript_11	Common libraries	1336	10.723	14	0.000	0.708	
Claim 4	lang_javascript_19	Common libraries	1344	18.964	16	0.027	0.271	
Claim 4	lang_javascript_06	Packages-Modules	1331	16.171	13	0.031	0.240	
Claim 4	lang_javascript_20	Common libraries	1345	10.672	14	0.000	0.712	
Claim 2	lang_javascript_29	Getters and setters	1354	10.989	15	0.000	0.753	
Claim 6	lang_javascript_30	Git Hooks	1355	12.335	14	0.000	0.579	
Claim 1	lang_javascript_14	Functions	1339	15.942	12	0.036	0.194	
Claim 3	lang_javascript_12	Browser-stroage	1337	7.813	12	0.000	0.800	

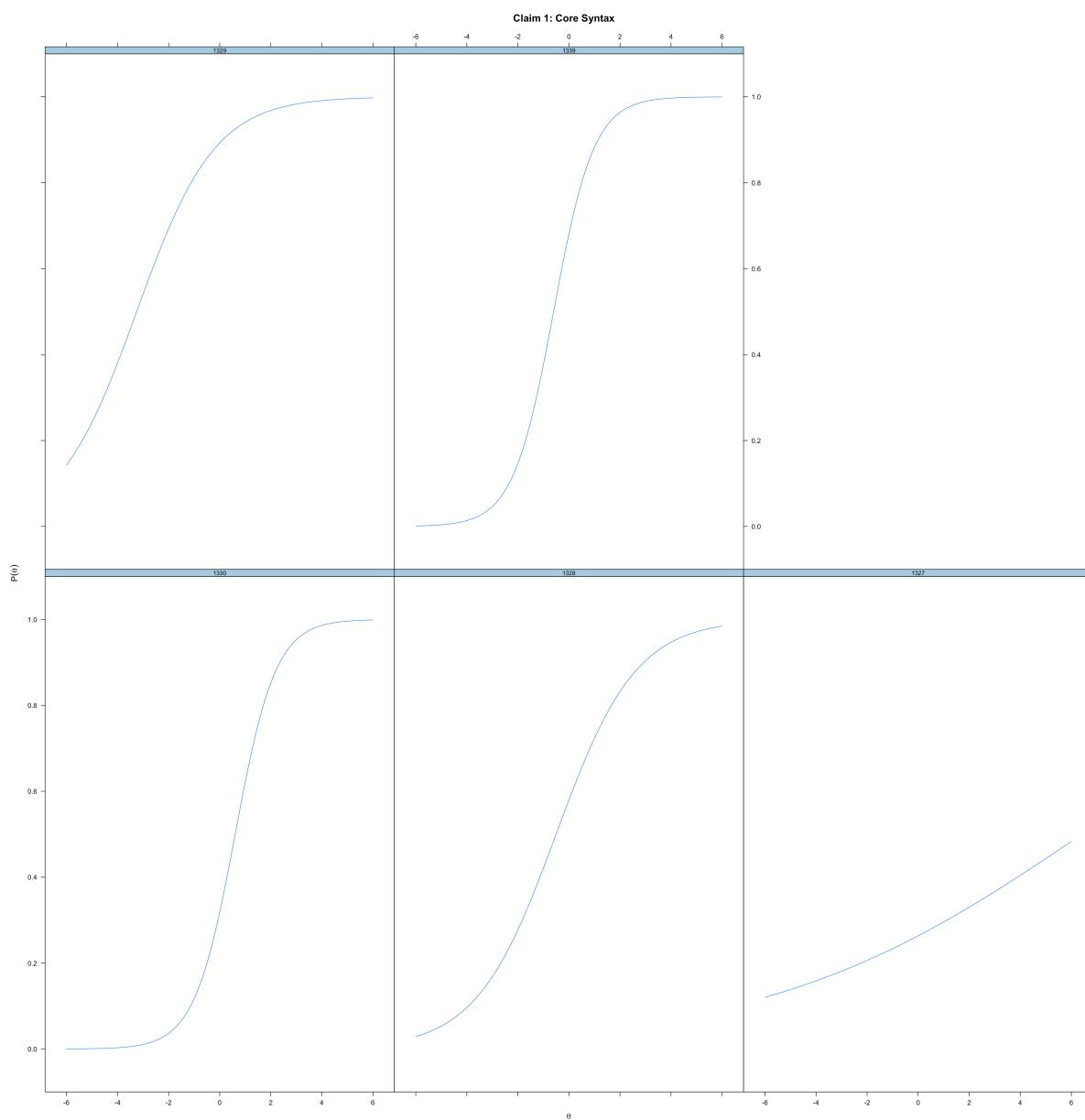
Items are flagged with '<=' to be reviewed or removed if *p-value* < 0.05. This indicates poor fit of 2PL model.

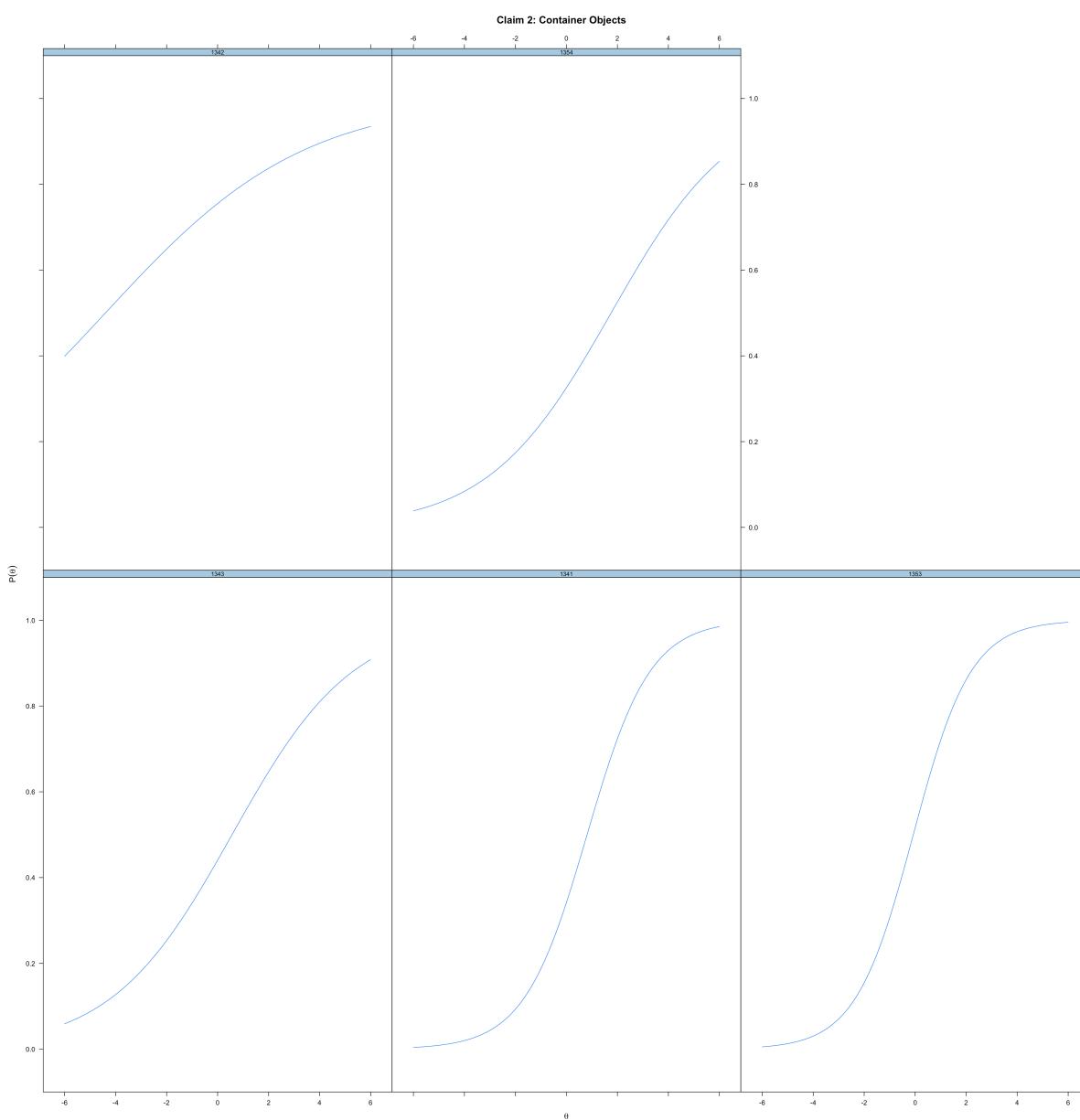
IRT Plots

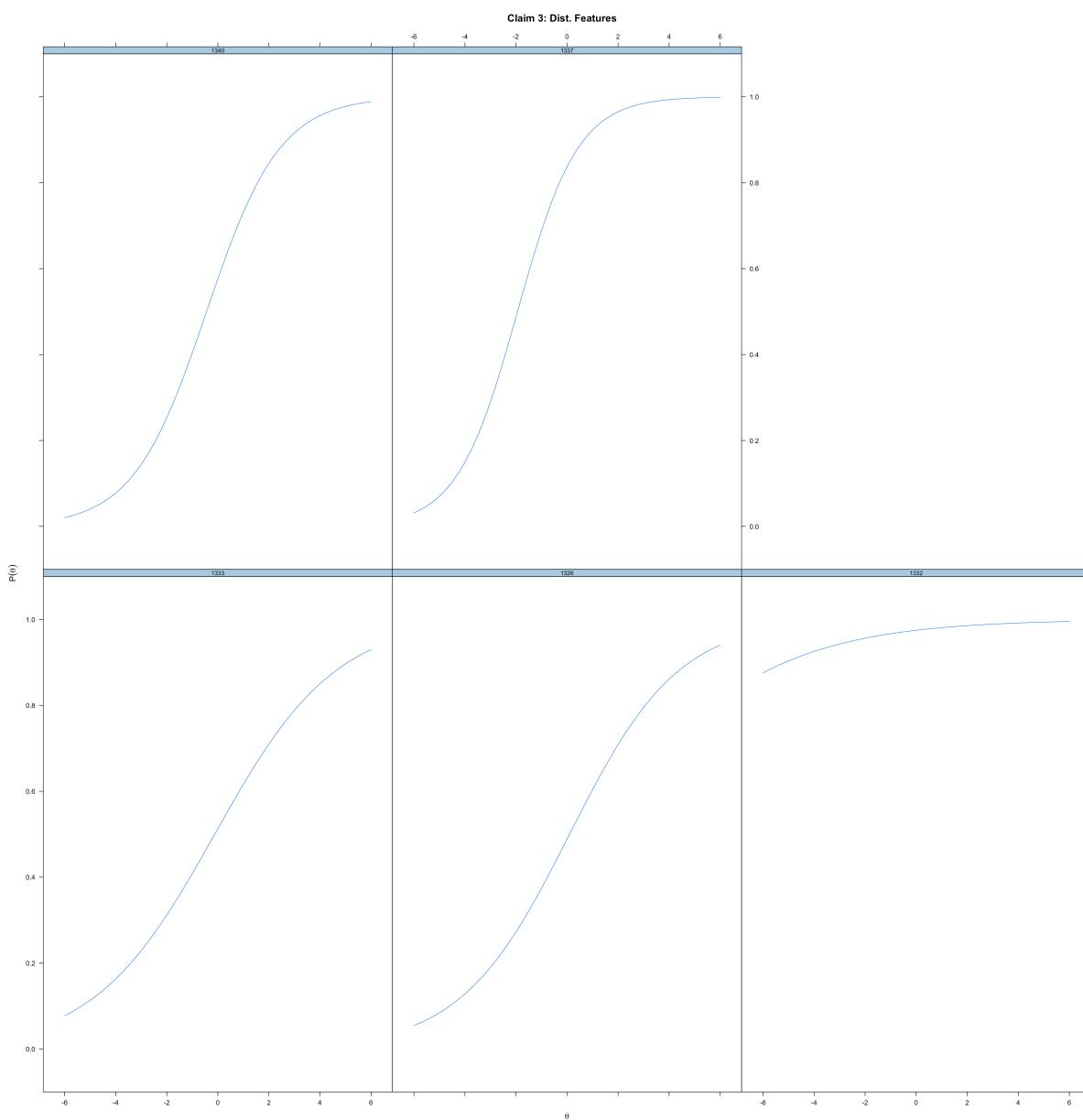
Item Local-Dependence Plots - Residual Dependencies given a unidimensional model:

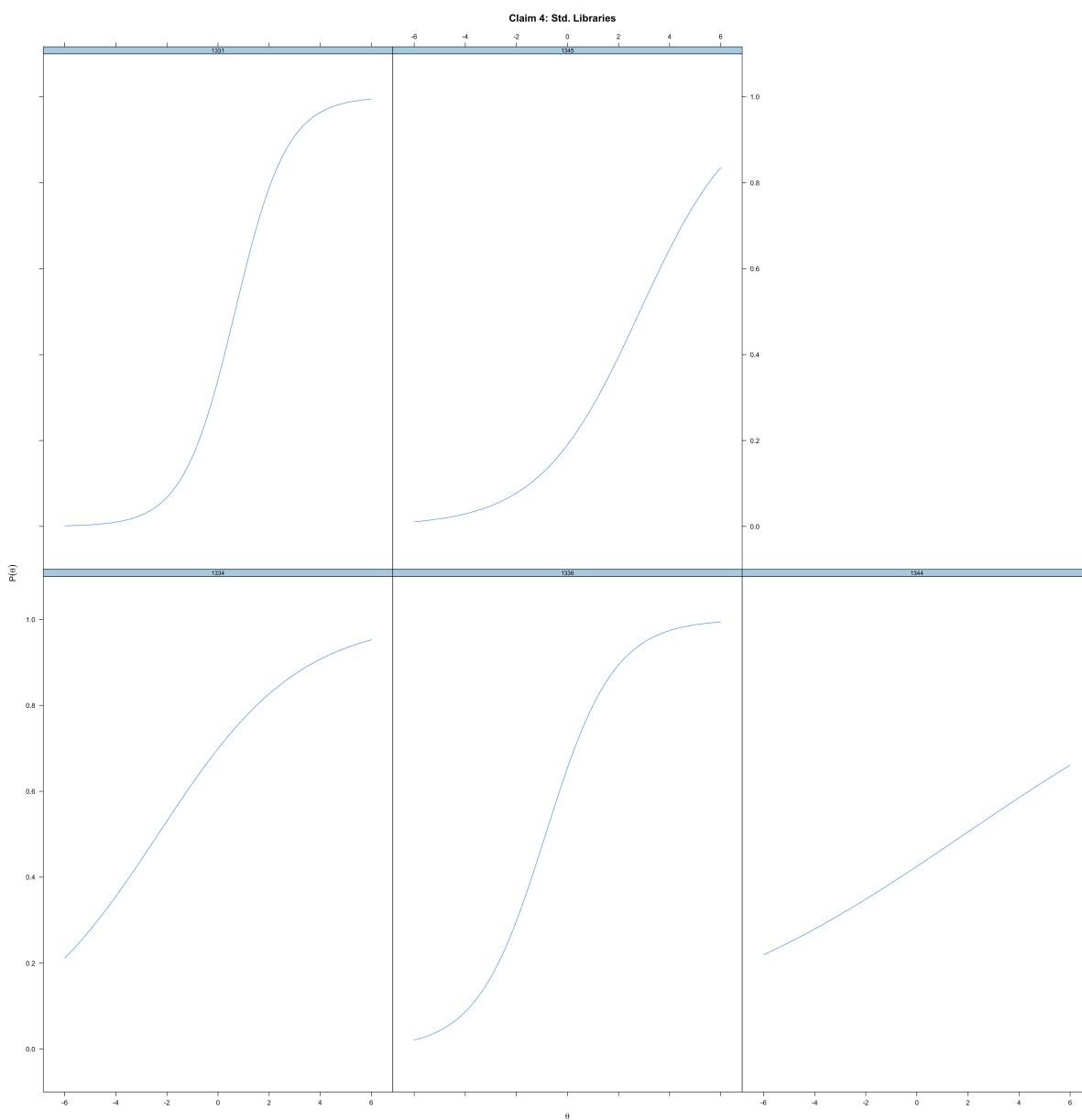


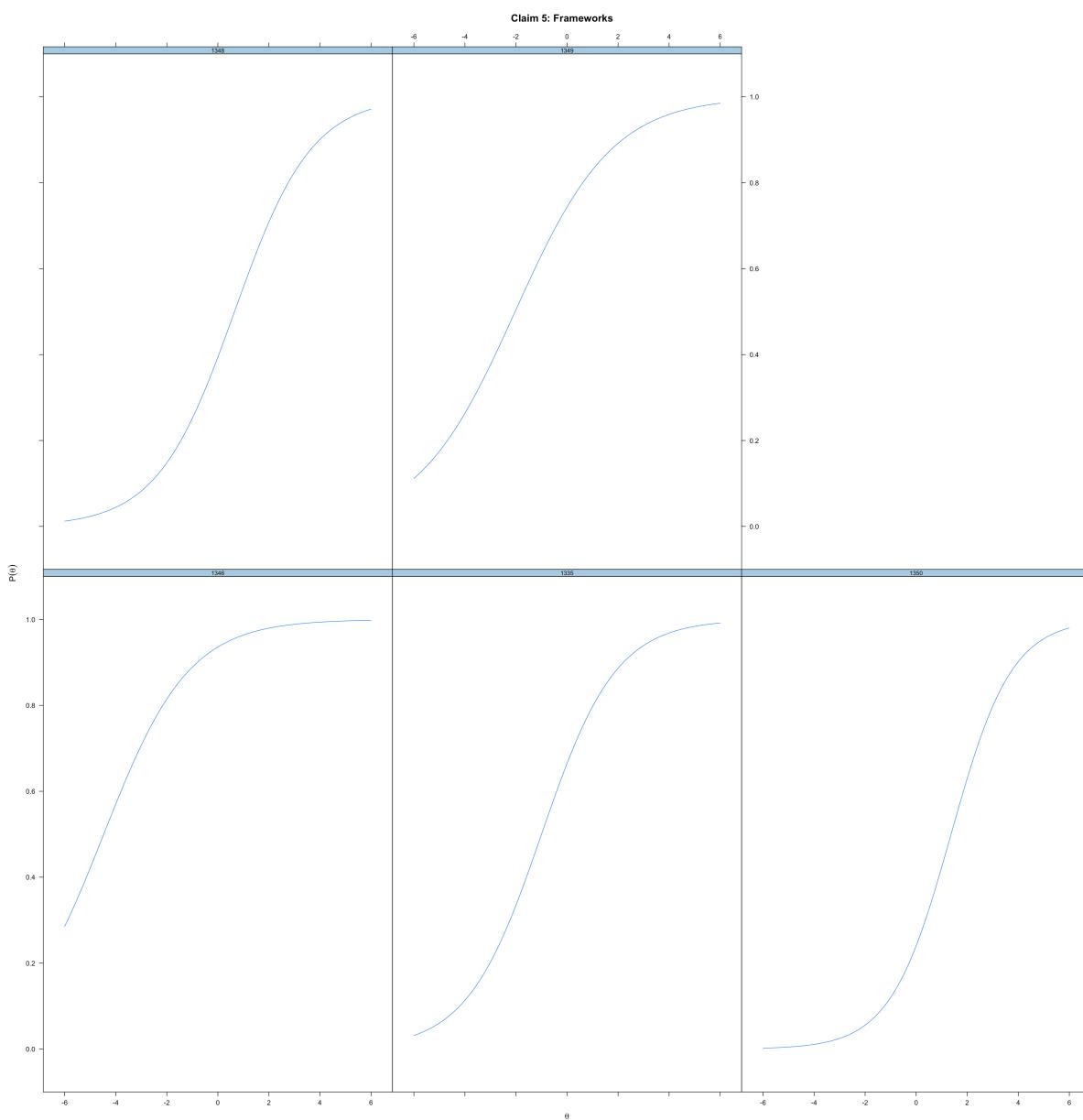
Item Characteristic Curves (Tracelines), by Claims

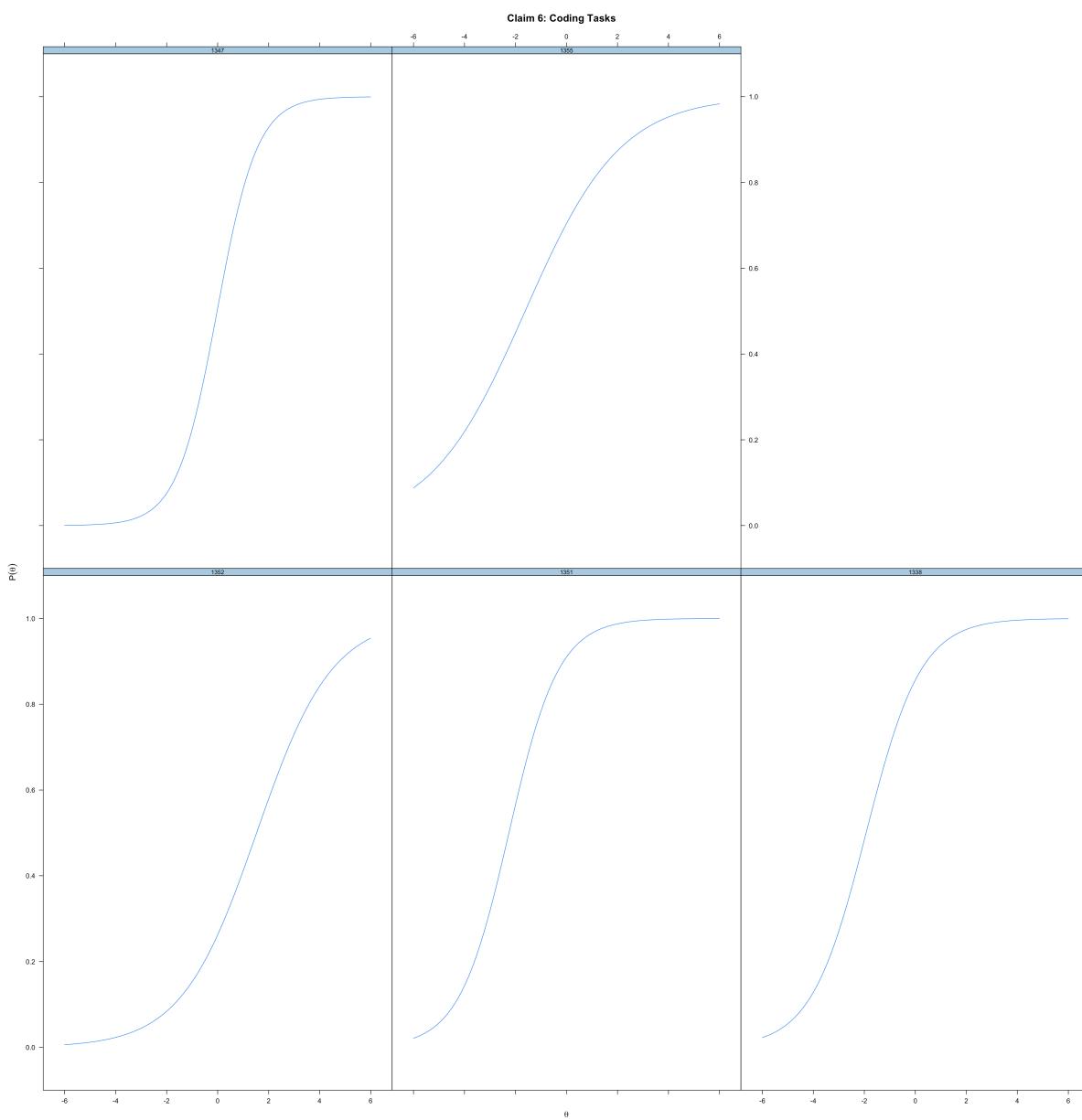




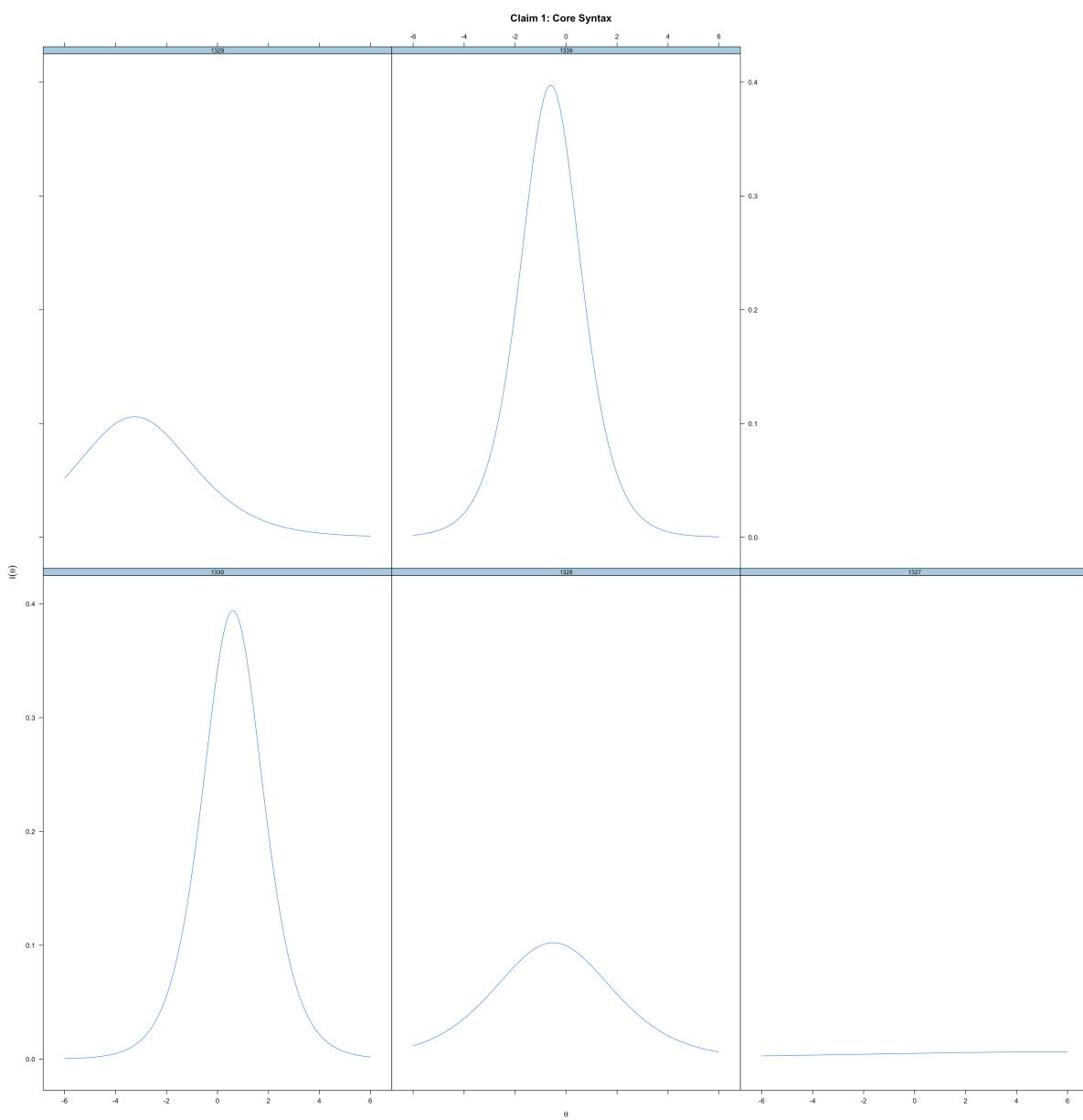


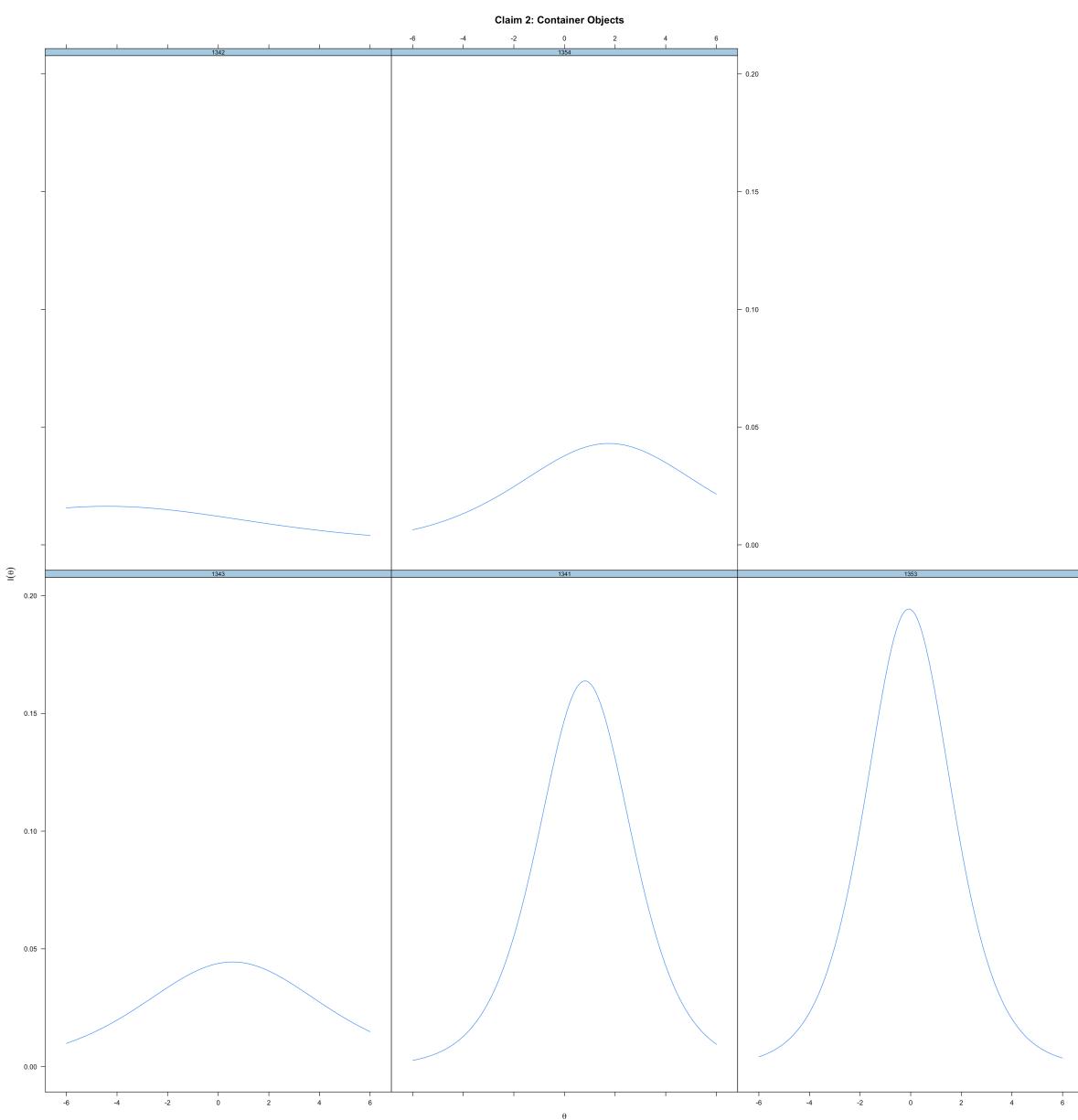


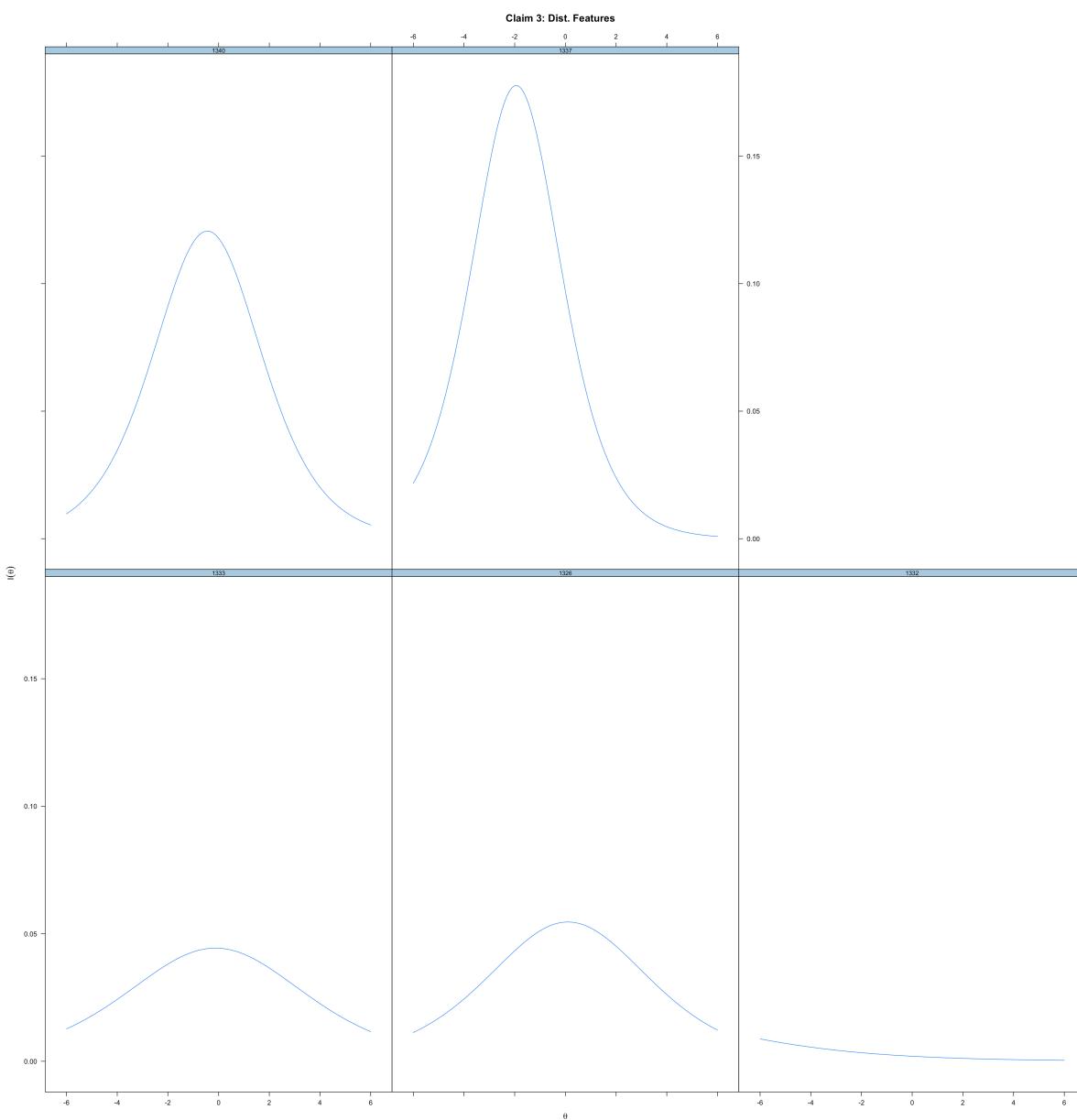


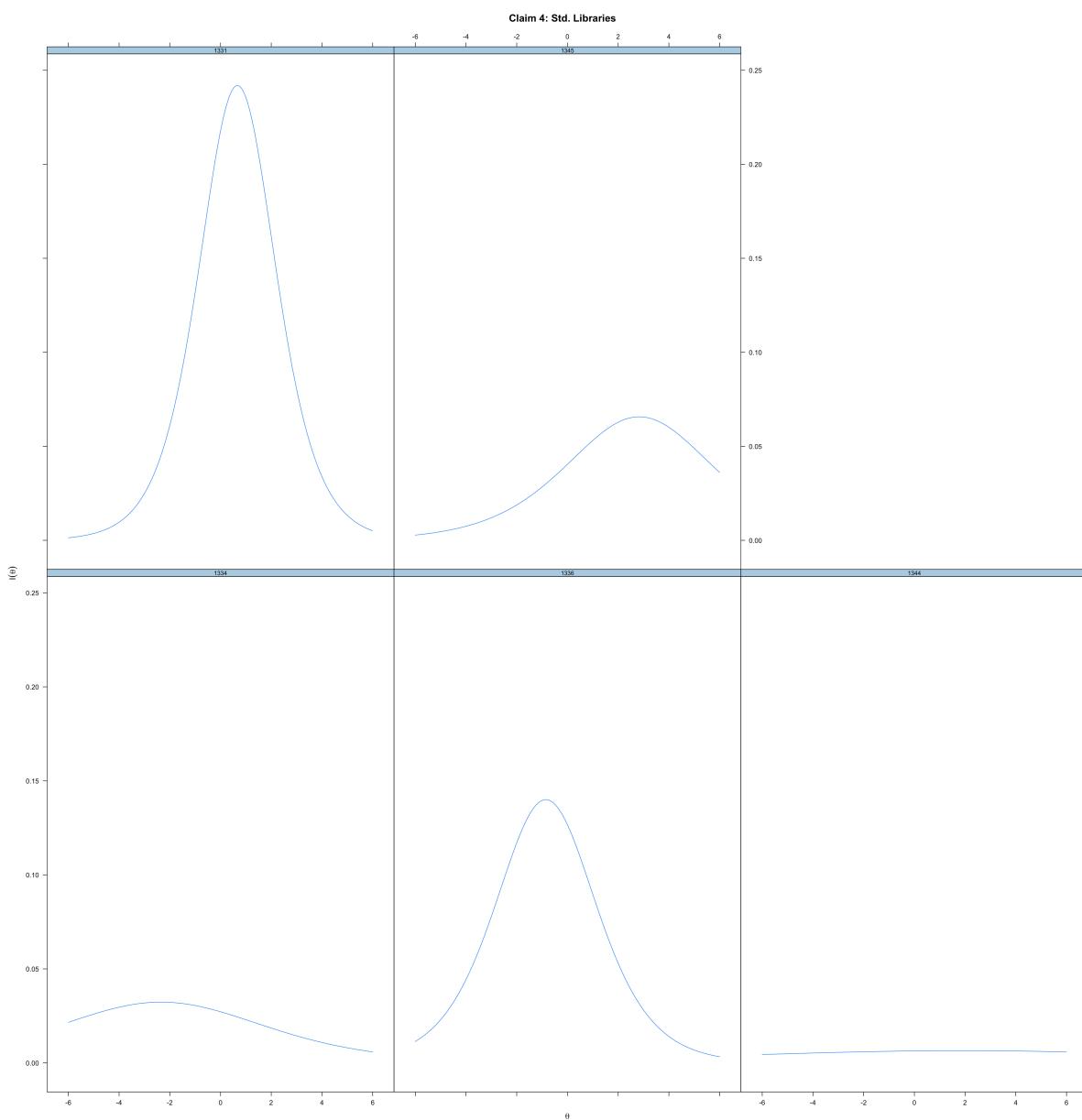


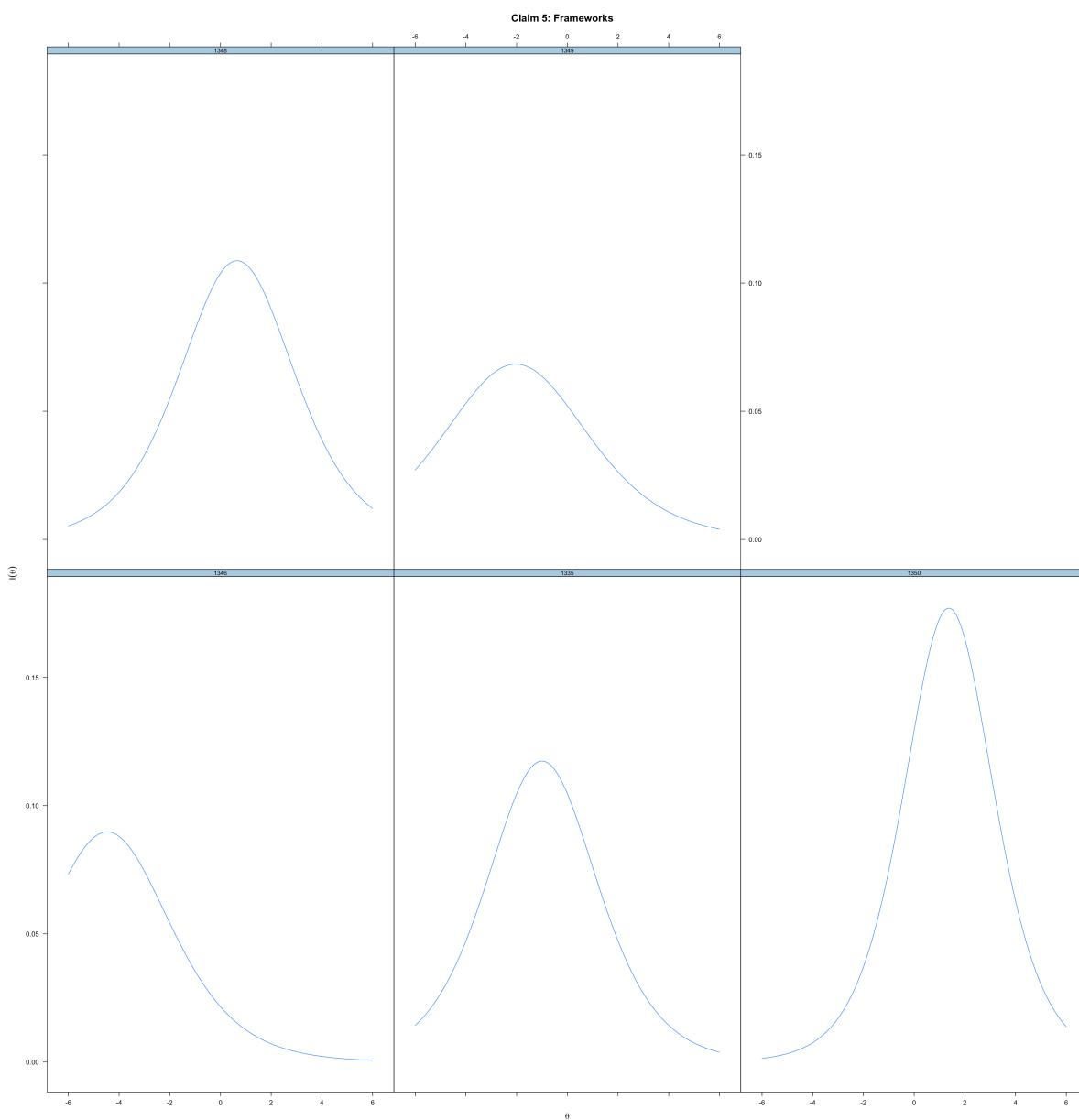
Item Information Curves by Claims

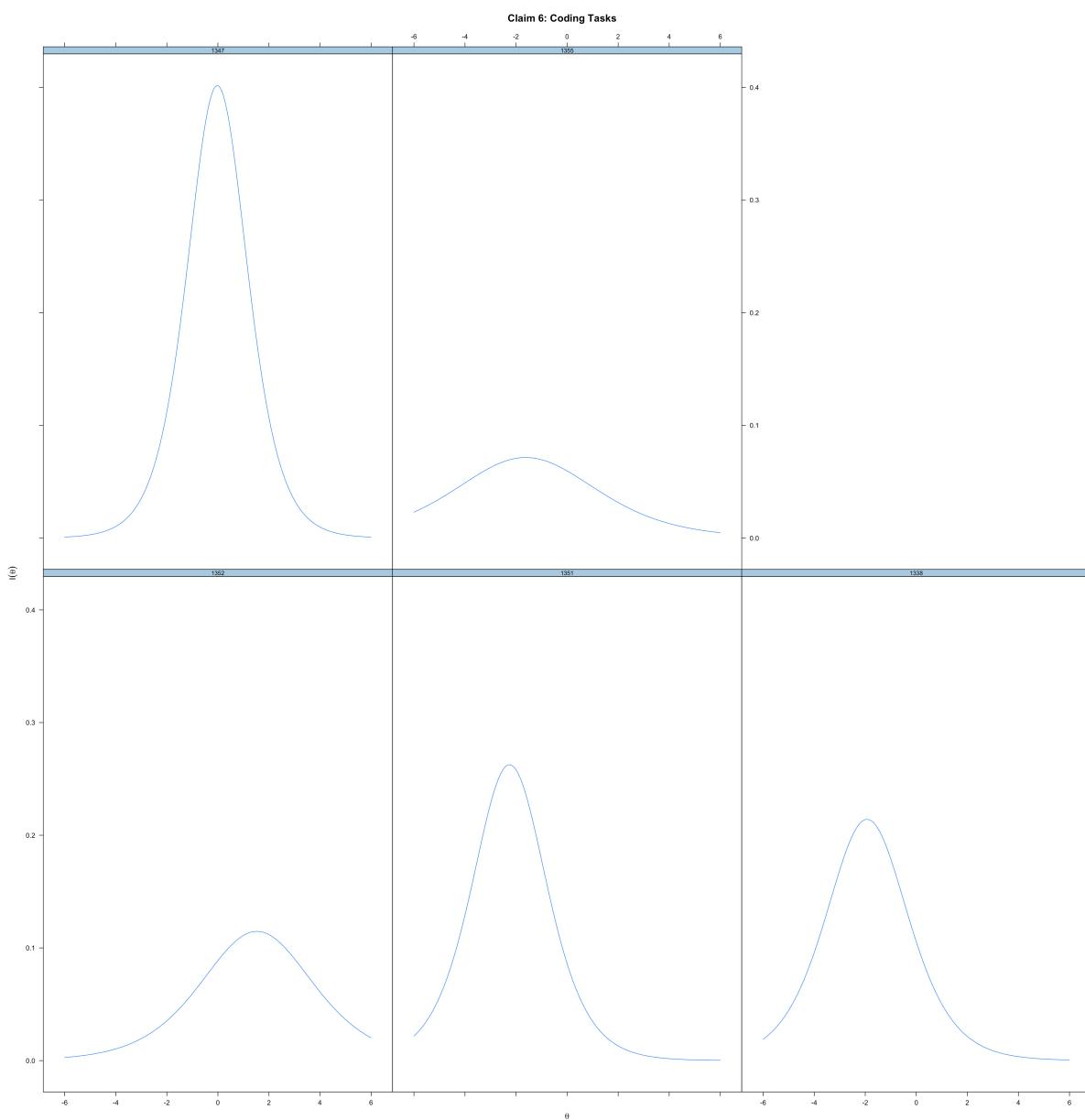




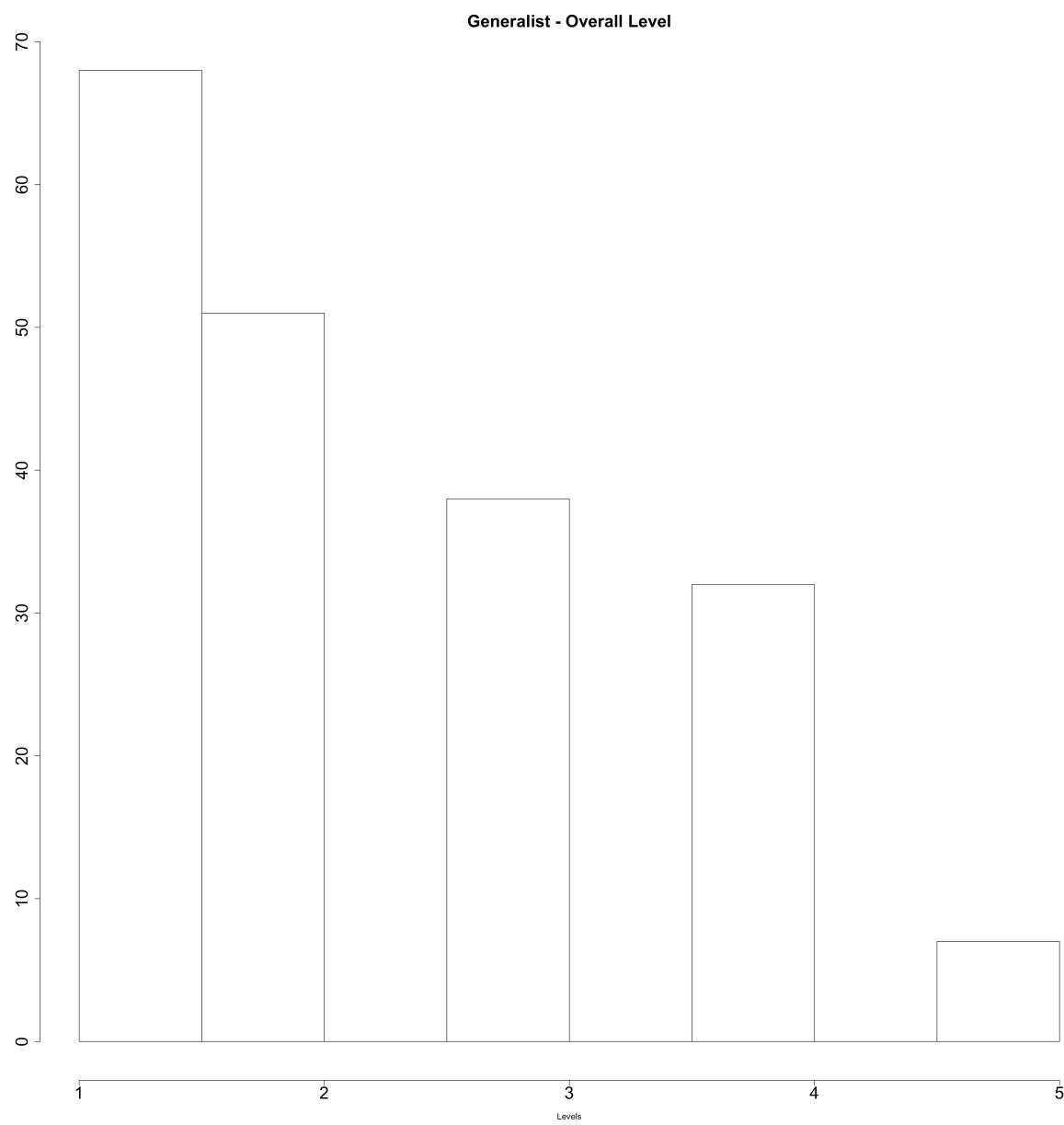


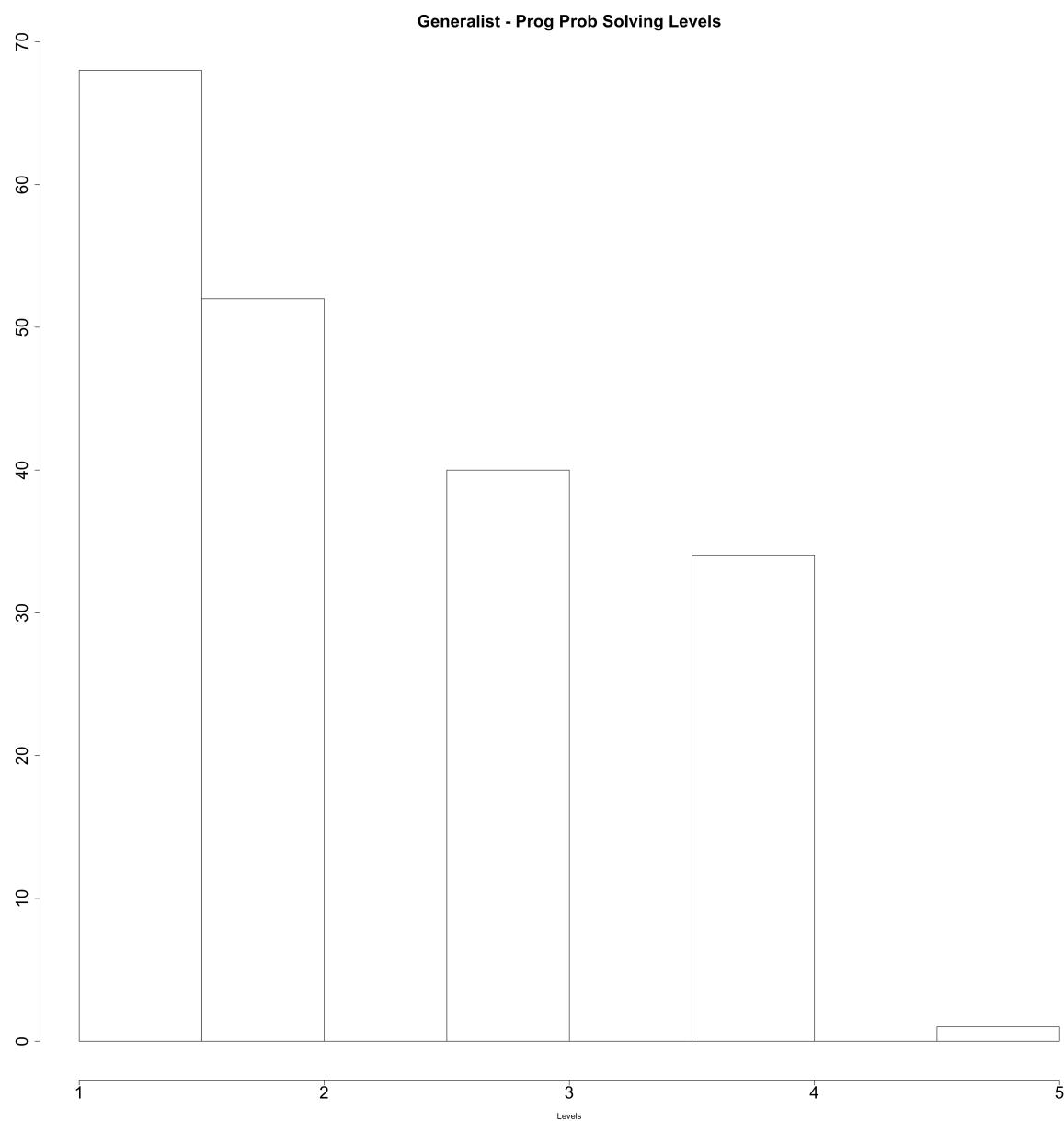






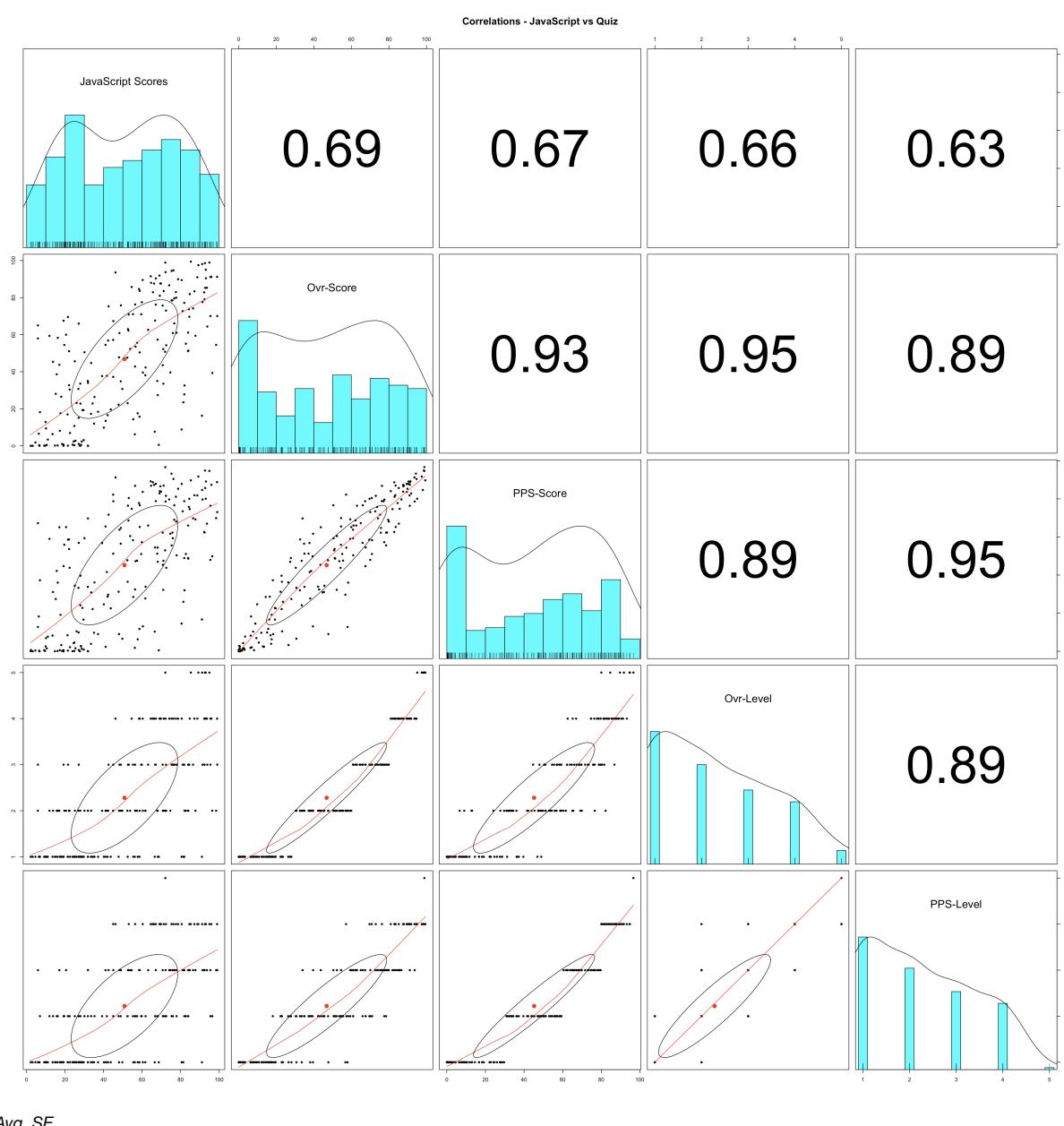
Calibration Sample Characteristics



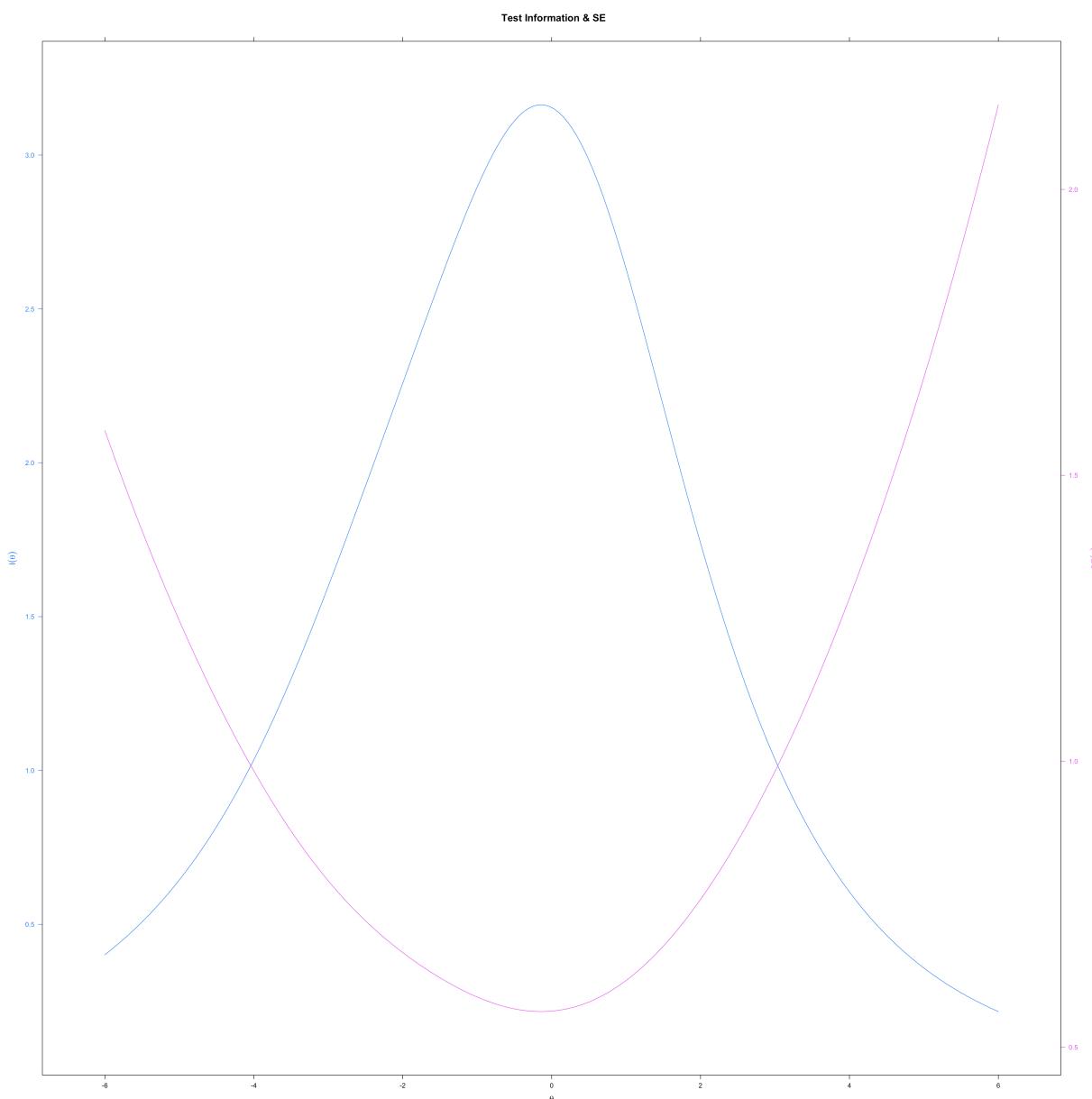


Scoring*

Distributions & Correlations



Avg. SE



Equating/Linking Scores

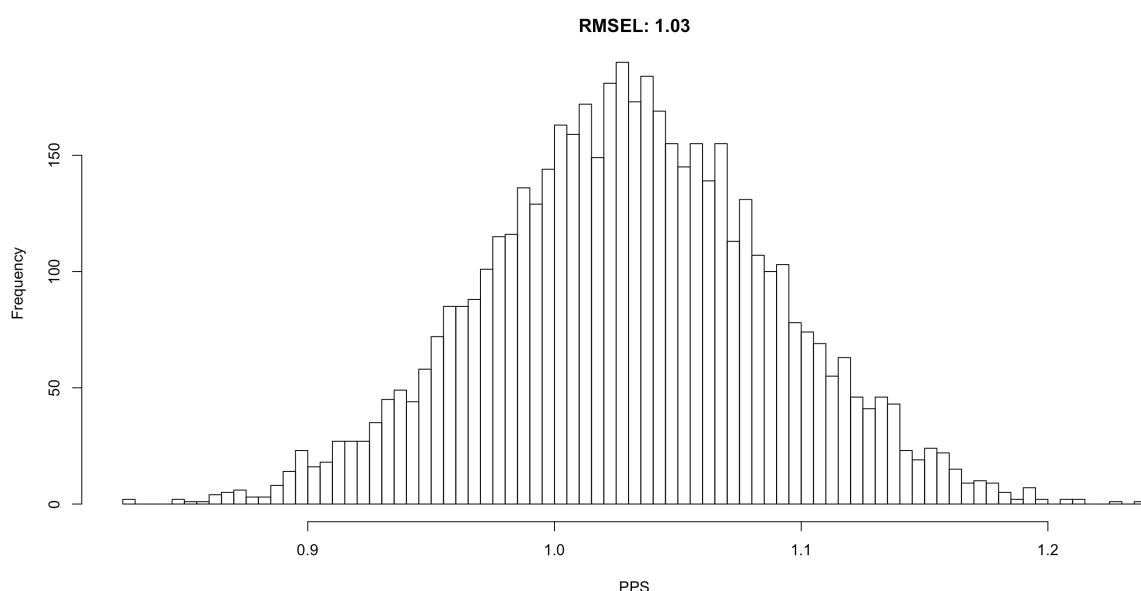
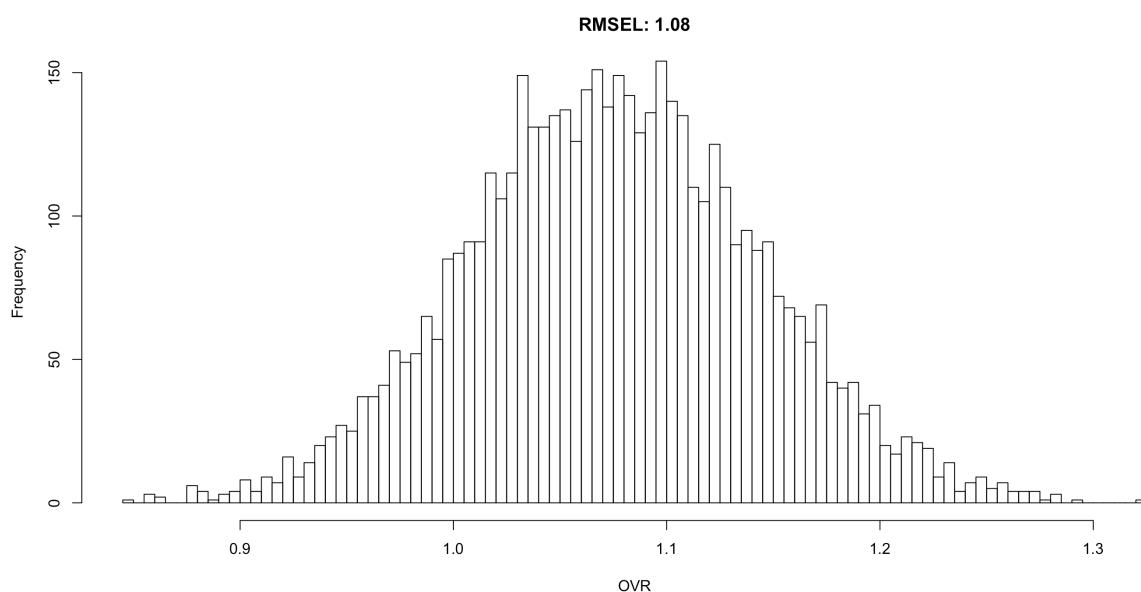
In order to get language assessment SkillEstimates on the same scale as the other quizzes, it requires being able to produce item parameters or IRT-ability estimates on the same scale as the other assessments.

The equating/linking design available is a *Single Group Design* where the same candidates have taken both a previous core quiz and a language assessment. Therefore, there are limited methodologies available to link the two scales together (plus with a limited sample size).

Three approaches explored are:

- Mean Method
- Linear Method
- Equipercentile Method

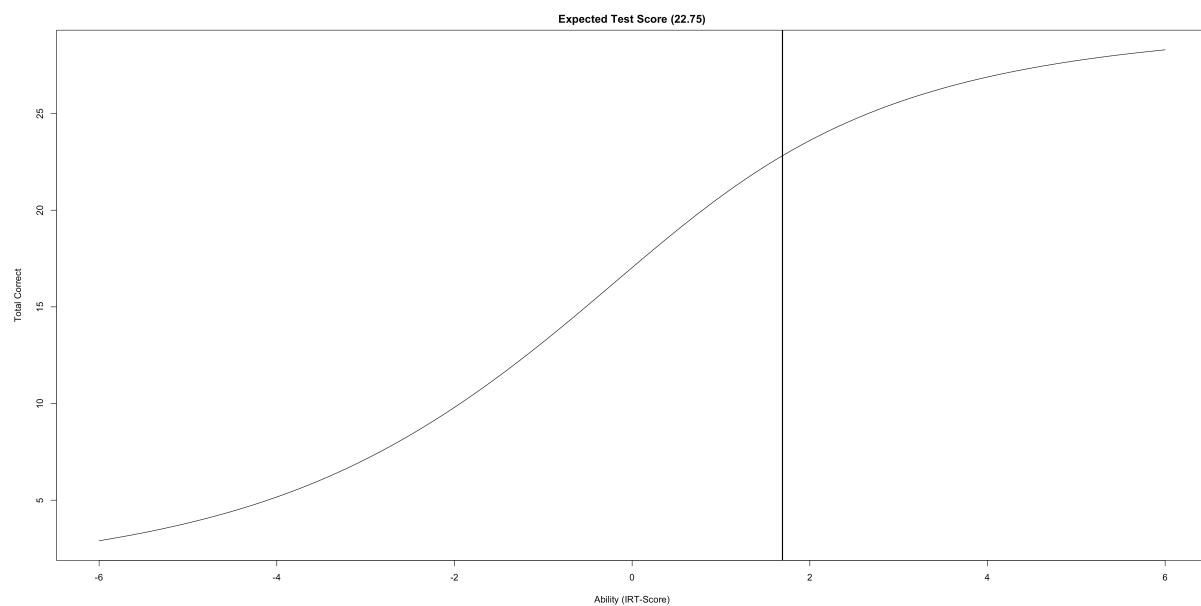
Mean Method



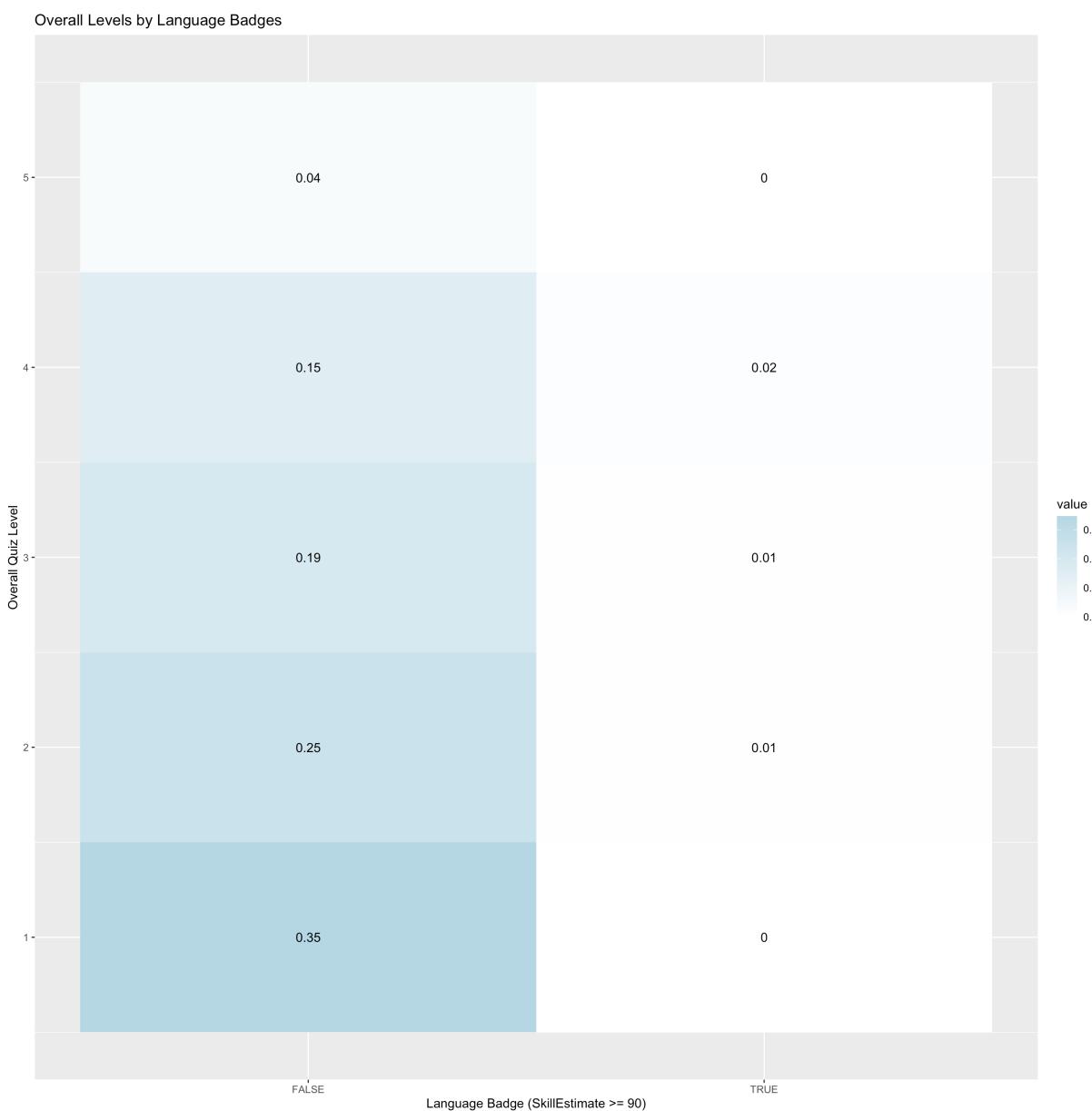
\$OVR		
	avg <dbl>	sd <dbl>
	1.075358	0.06831261
1 row		
\$PPS		
	avg <dbl>	sd <dbl>
	1.030332	0.05851488
1 row		
NA		

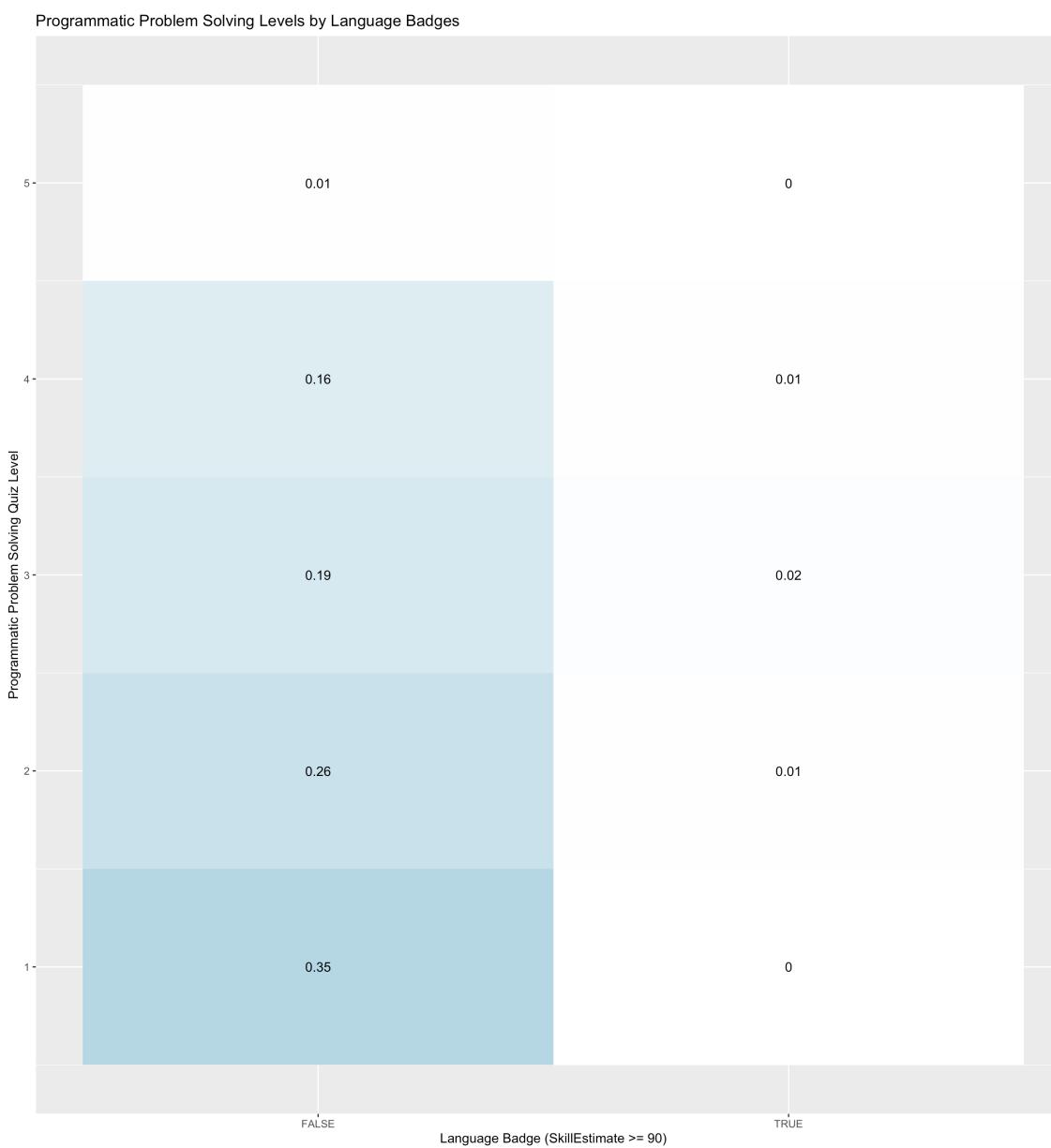
Result: Programmatic Problem Solving Scales score produce smaller linking error

Badges - Test Characteristic Curve

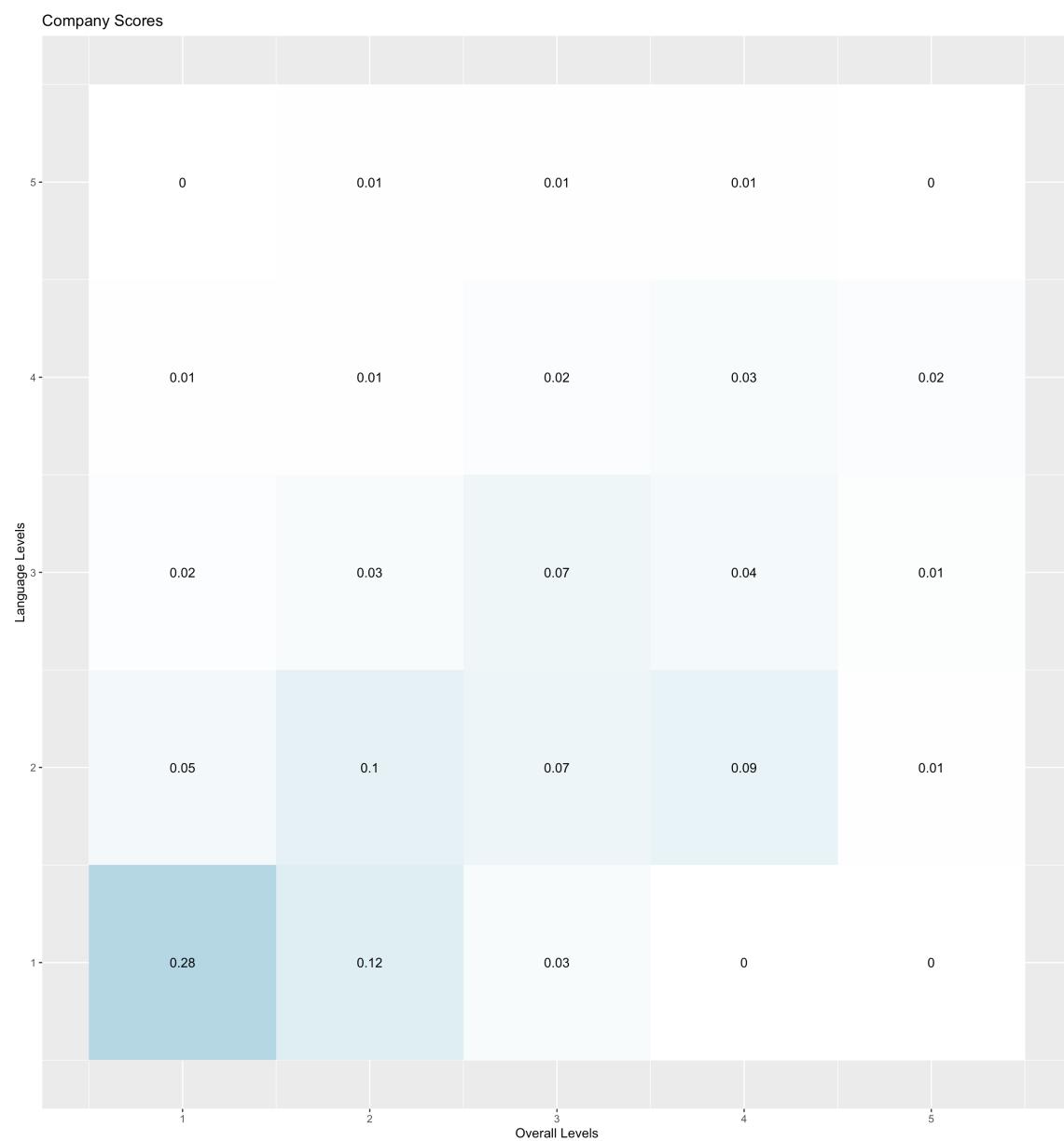


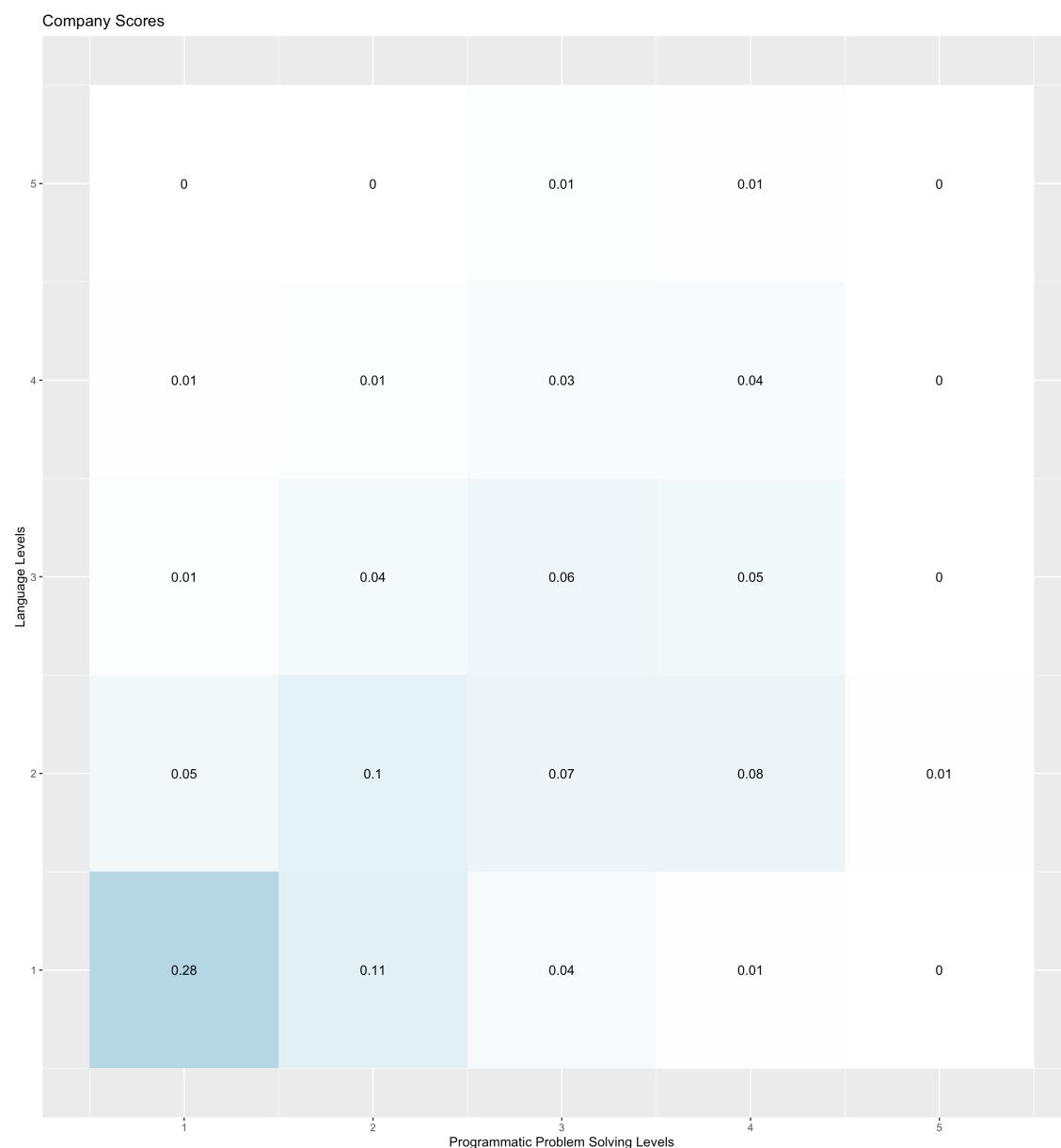
Badges - Quiz Level, by Badge



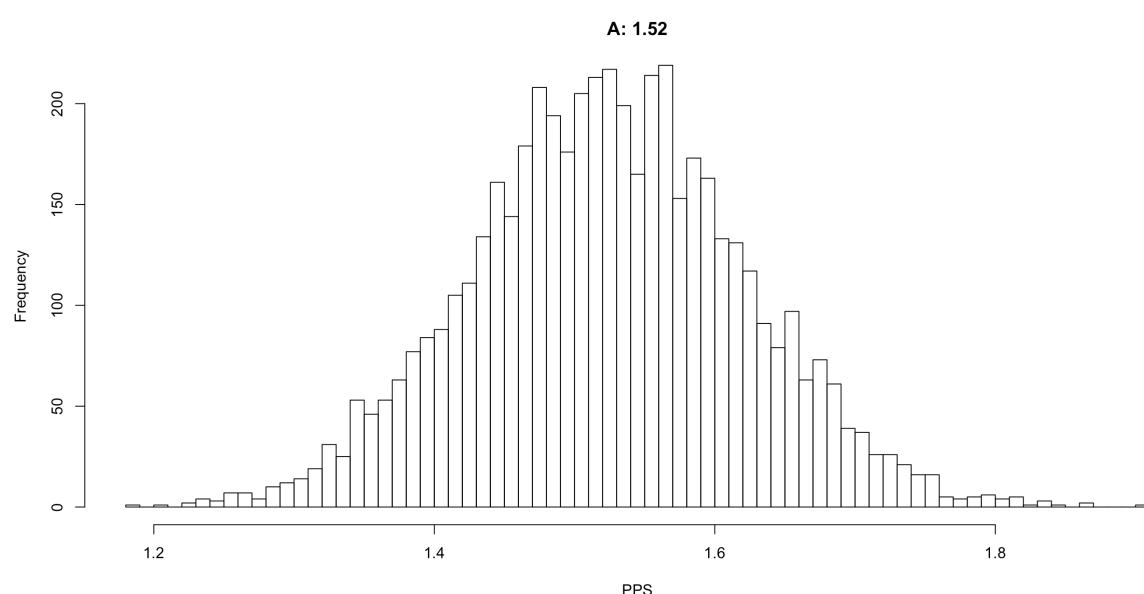
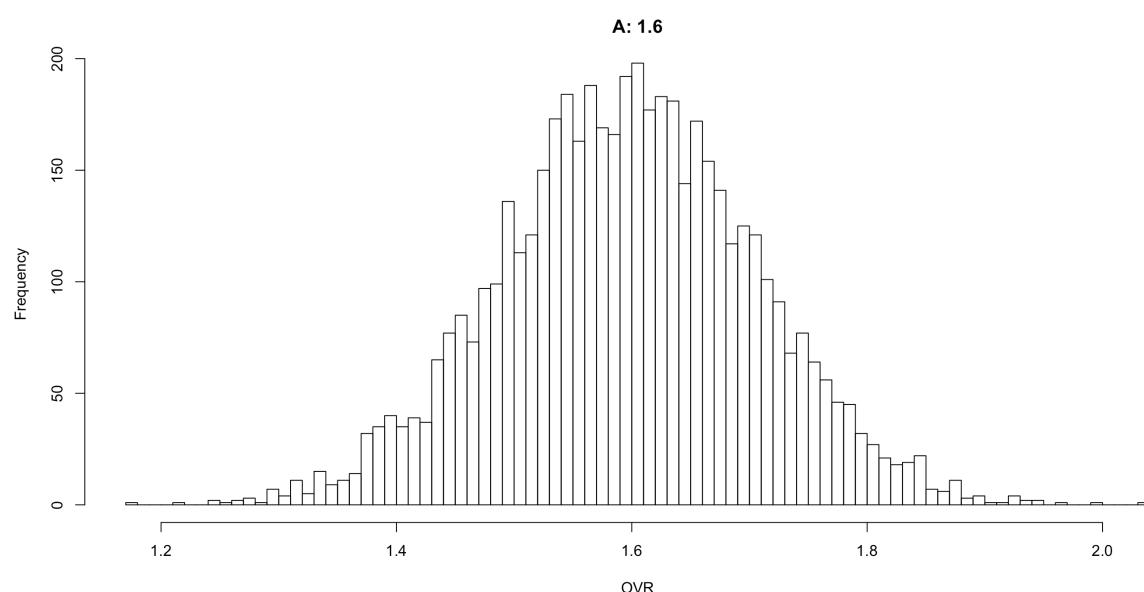


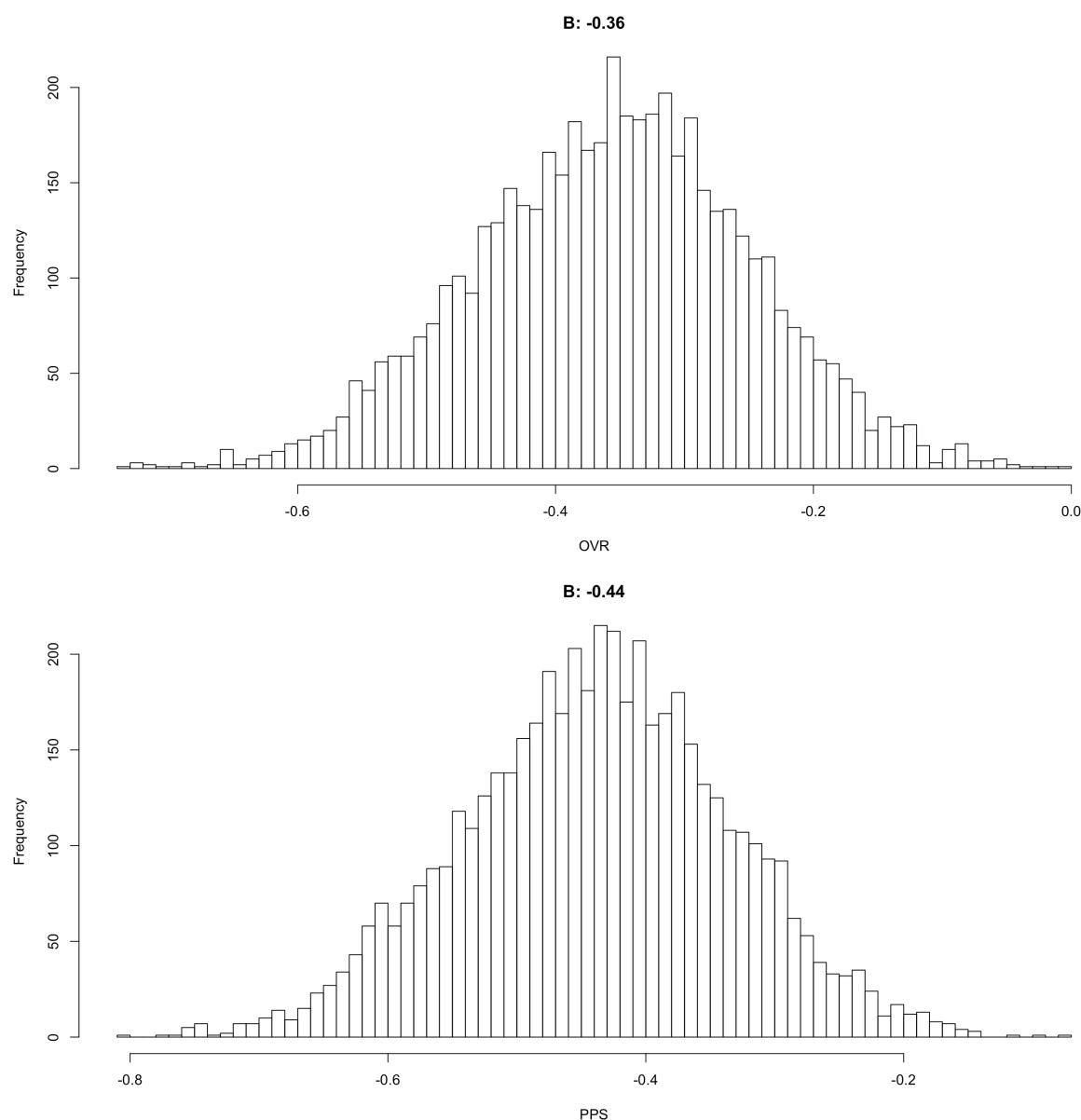
Language Levels, Quiz Levels

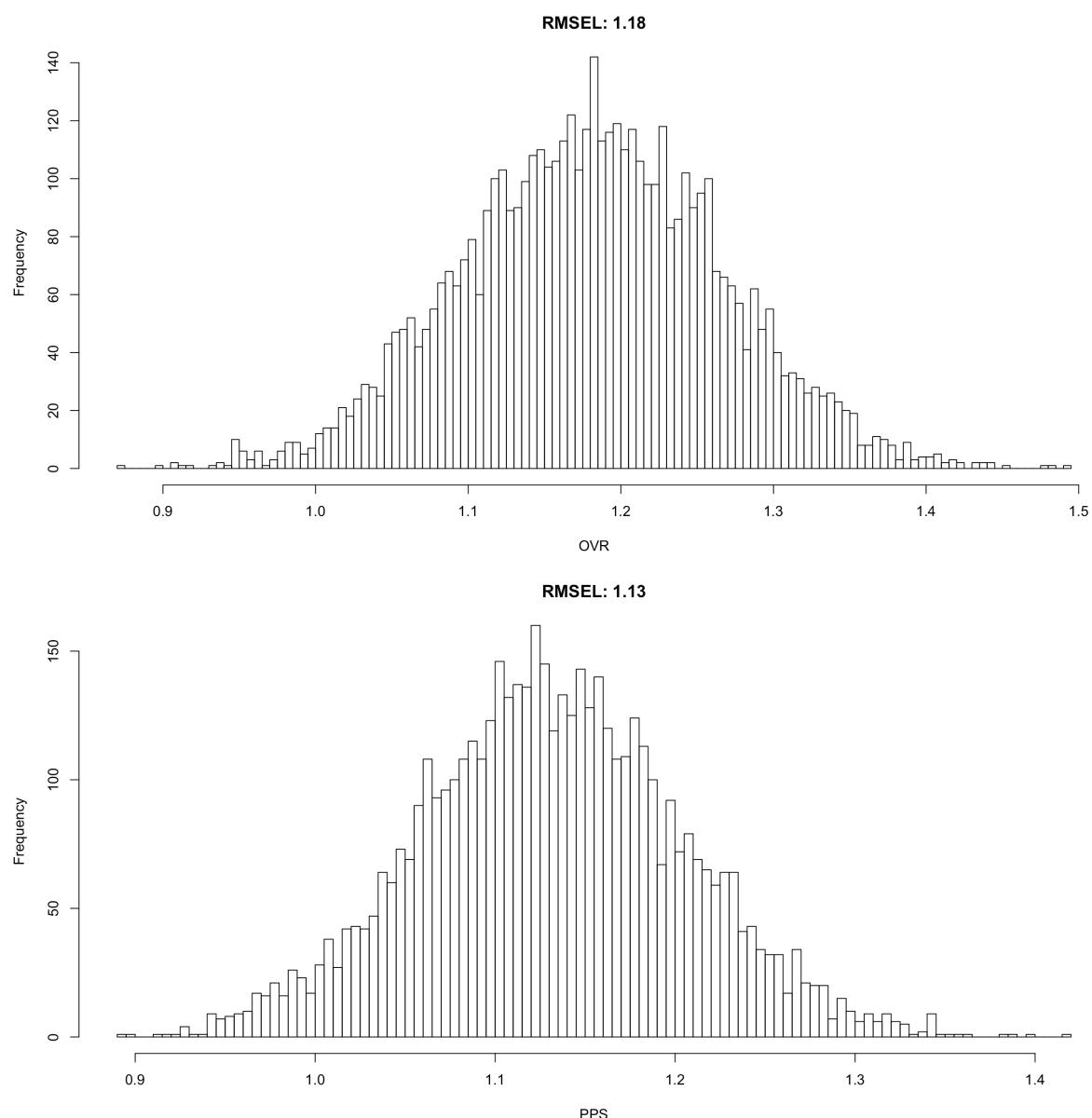




Linear Method

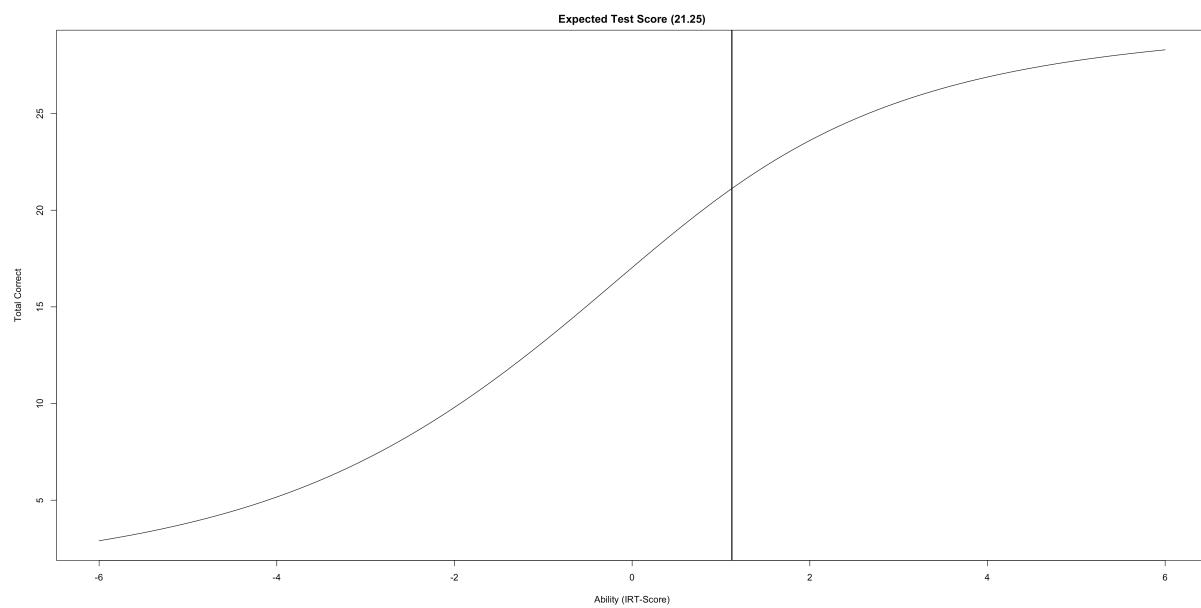




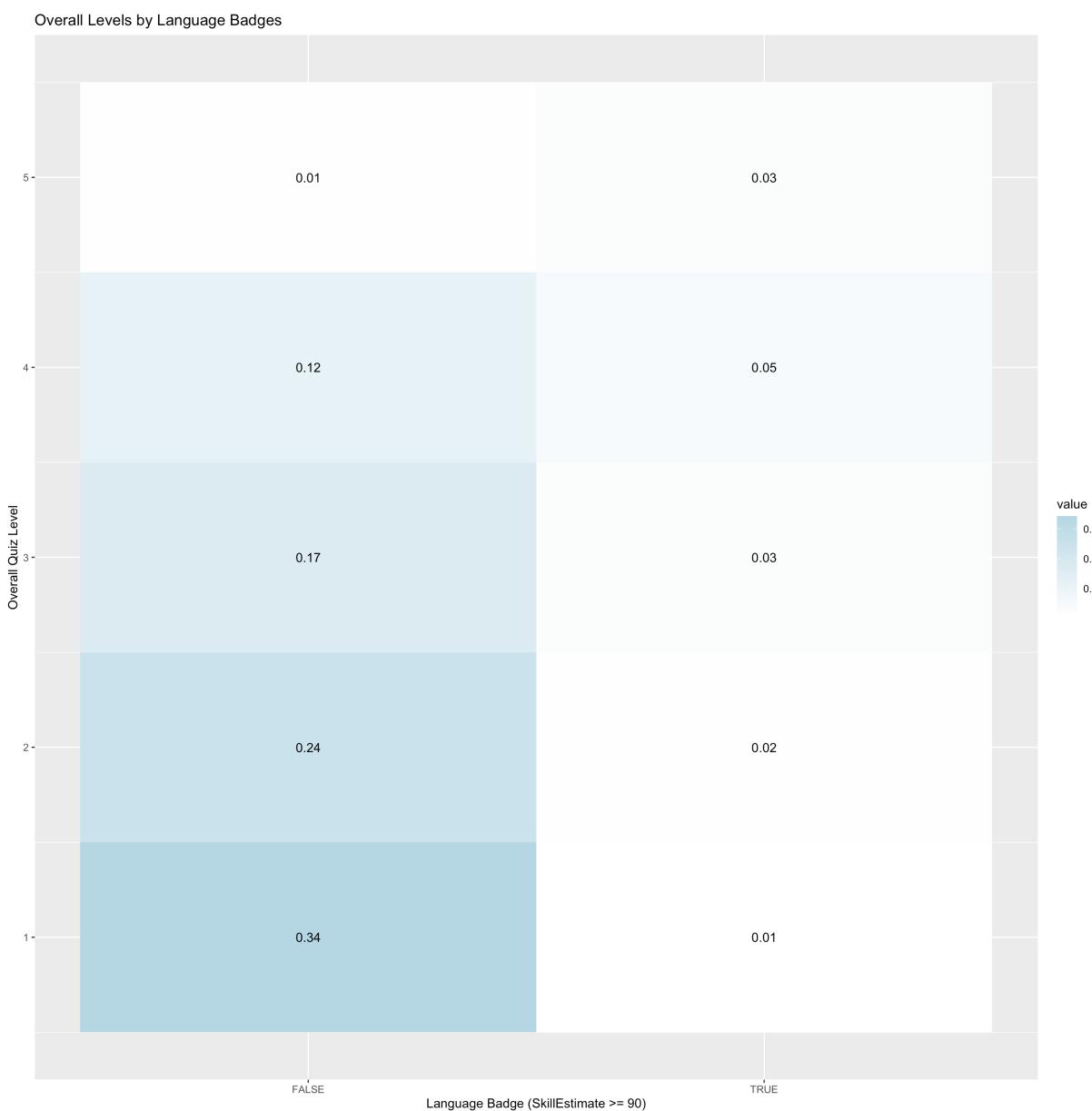


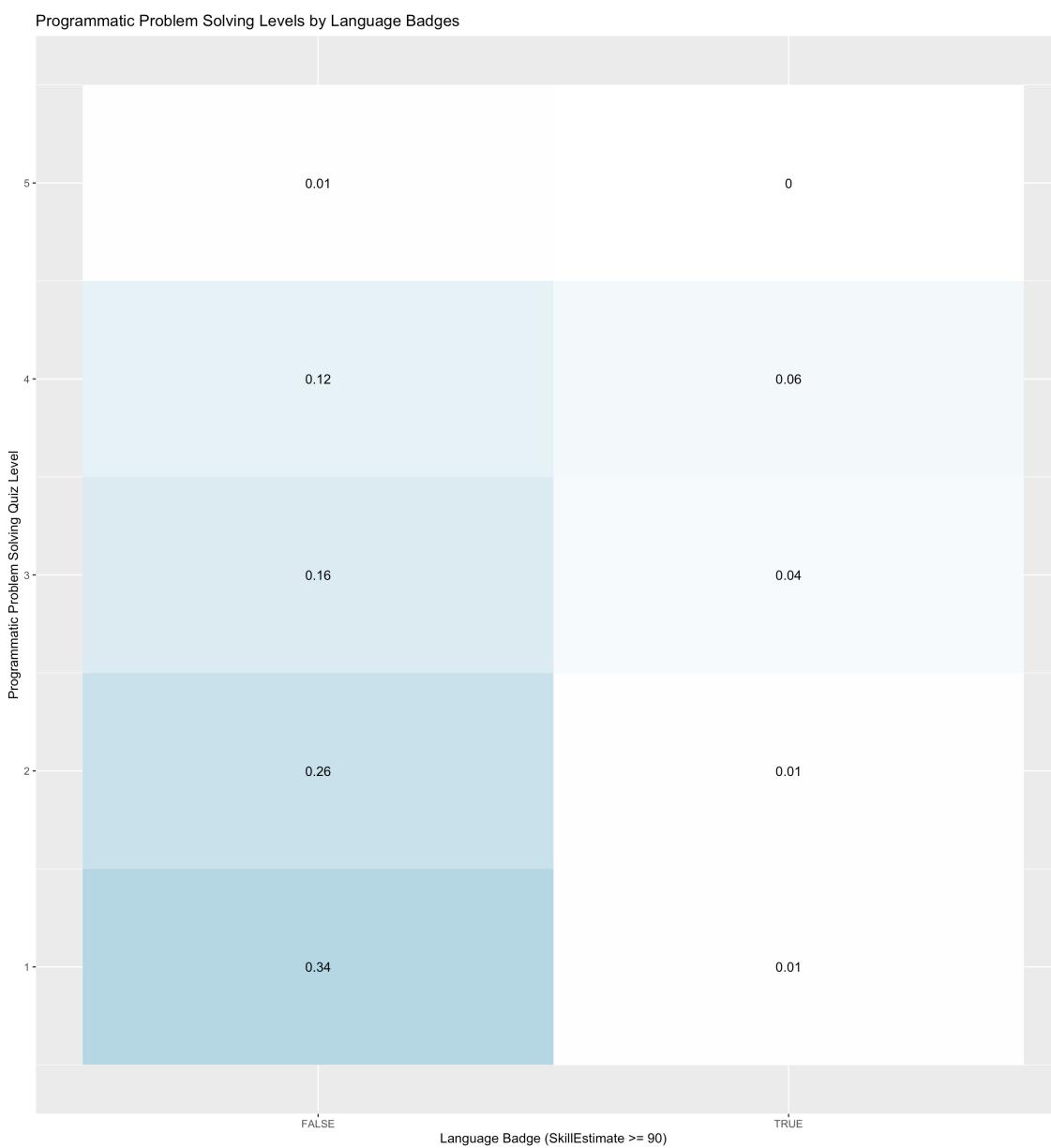
Result: Programmatic Problem Solving Scales score produce smaller linking error

Badges - Test Characteristic Curve



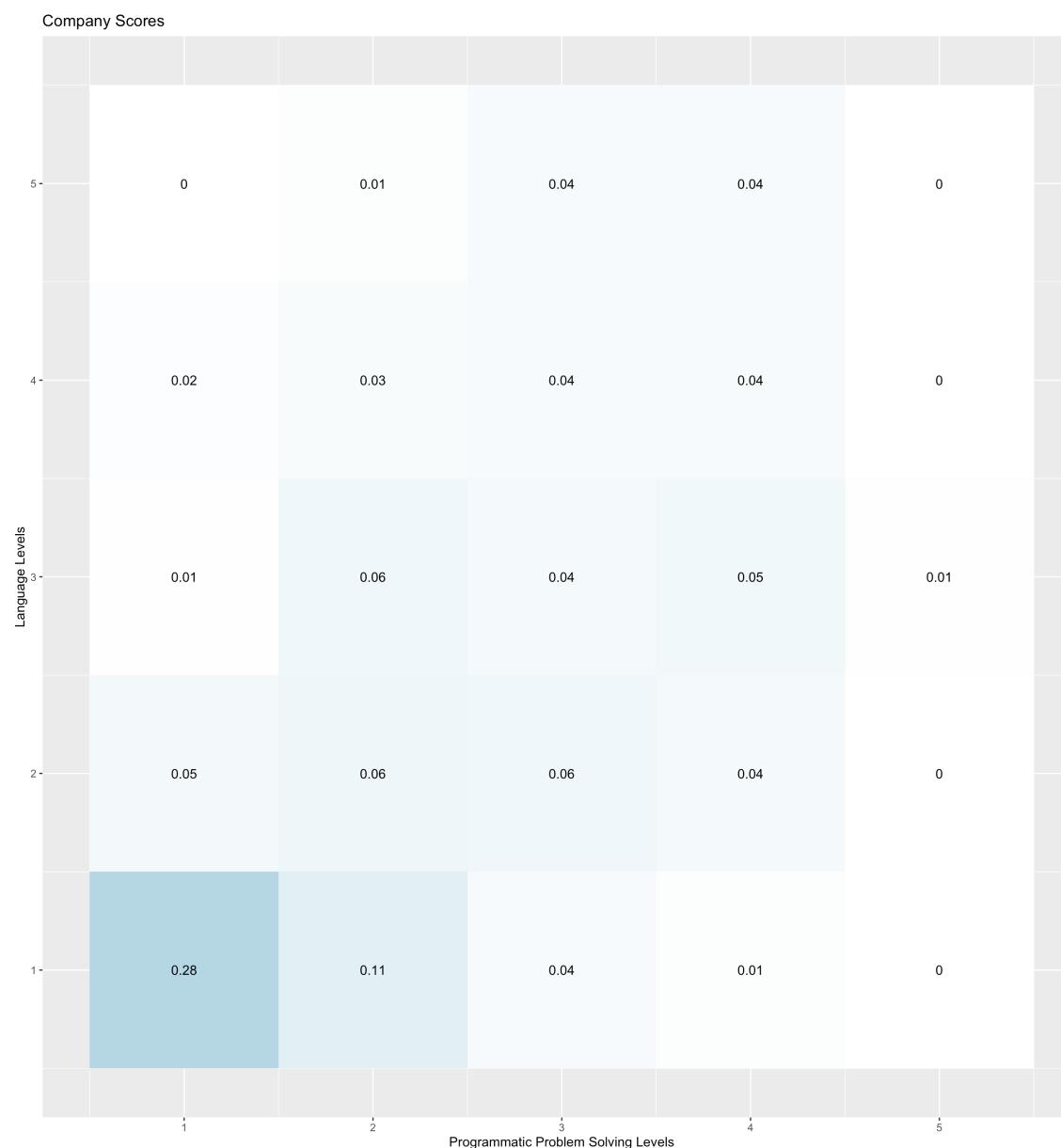
Badges - Quiz Level, by Badge





Language Levels, Quiz Levels





Percentile Method

Summary