Code ▾

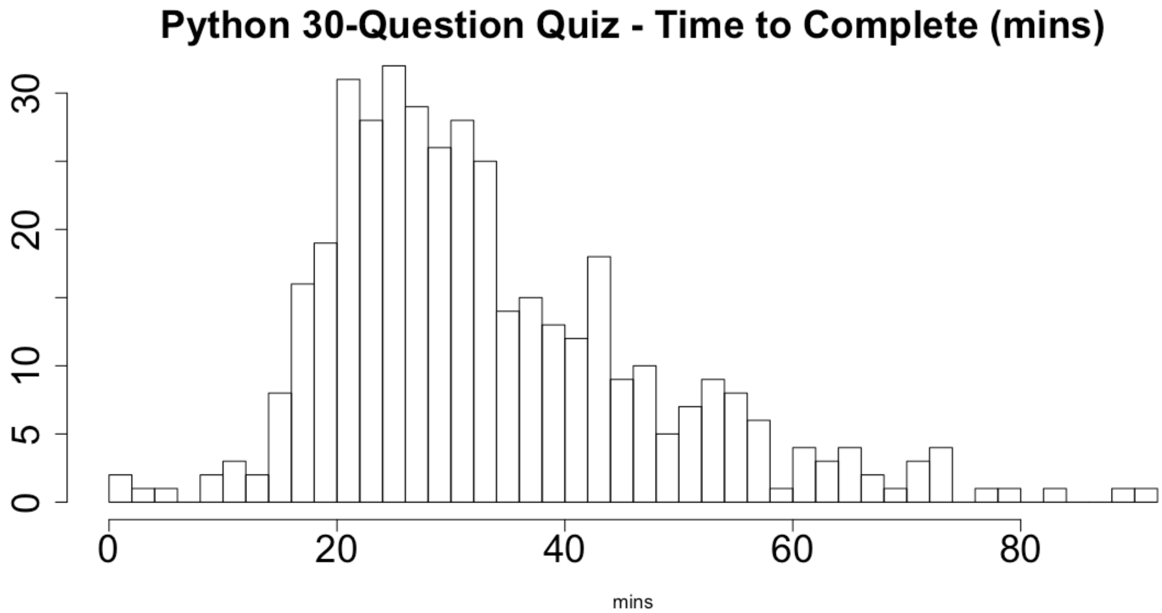
Python Assessment

Calibration, Scoring, & Impact Analyses

Number of Questions Attempted

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	18	19	30	NA
12	5	11	4	3	3	3	3	1	2	4	2	1	1	1	1	1	406	0

Time spent on Python Quiz



n	min	q1	avg	med	q3	max
<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
406	1.18	23.1675	33.45458	30.25	41.5725	90.72

1 row

Time spent on Python Quiz, by Claim

Time spent on Python 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 Python Language Quiz of those who attempted it is 0.875.

The proportion of candidates who have a valid attempted is 0.838 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 taskers. (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 consitency) - 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 Python Reliability is: 0.815
```

df_blueprint$question_id	flag	claim	name	target	n_candidates	item_mean	item_total_correlation	alpha_del
1296	<=	Claim 1	lang_python_01	`for loops`	389	0.900	0.335	(
1297	<=	Claim 1	lang_python_02	`lamda function`	389	0.856	0.445	(
1298		Claim 1	lang_python_03	`variable`	389	0.725	0.324	(
1299	<=	Claim 1	lang_python_04	`strings`	389	0.830	0.249	(
1300		Claim 1	lang_python_05	`String`	389	0.697	0.367	(
1301		Claim 1	lang_python_06	`zip`	389	0.779	0.258	(
1302		Claim 2	lang_python_07	`classes`	389	0.712	0.465	(
1303	<=	Claim 2	lang_python_08	`classes`	389	0.851	0.312	(
1304		Claim 2	lang_python_09	`iterators`	389	0.607	0.269	(
1305	<=	Claim 2	lang_python_10	NA	389	0.913	0.288	(
1306		Claim 2	lang_python_11	`string`	389	0.751	0.312	(
1307	<=	Claim 2	lang_python_12	`classes`	389	0.853	0.414	(
1308	<=	Claim 2	lang_python_13	`dictionary-comprehension`	389	0.856	0.411	(
1309		Claim 3	lang_python_14	`dictionary`	389	0.545	0.492	(
1310	<=	Claim 3	lang_python_15	`list-comprehension`	389	0.859	0.395	(
1311		Claim 3	lang_python_16	`read-json`	389	0.530	0.381	(
1312		Claim 3	lang_python_17	`static-method`	389	0.702	0.414	(
1313		Claim 3	lang_python_18	`write-file`	389	0.622	0.211	(
1314	<=	Claim 4	lang_python_19	`function args`	389	0.802	0.409	(
1315	<=	Claim 4	lang_python_20	`error-handing`	389	0.913	0.216	(
1316	<=	Claim 4	lang_python_21	`import-module`	389	0.866	0.397	(

df_blueprint$question_id	flag	claim	name	target	n_candidates	item_mean	item_total_correlation	alpha_del
1317		Claim 4	lang_python_22	`name-main`	389	0.769	0.268	(
1318	<=	Claim 4	lang_python_23	`name-space`	389	0.974	0.327	(
1319		Claim 4	lang_python_24	`requirements-txt`	389	0.728	0.458	(
1320		Claim 5	lang_python_25	`datetime`	389	0.452	0.259	(
1321	<=	Claim 5	lang_python_26	`flask`	389	0.879	0.286	(
1322		Claim 5	lang_python_27	`numpy`	389	0.483	0.249	(
1323		Claim 5	lang_python_28	`numpy`	389	0.645	0.283	(
1324	<=	Claim 5	lang_python_29	`requests`	389	0.620	0.187	(
1325		Claim 5	lang_python_30	`pandas`	389	0.373	0.251	(

IRT Analysis

Unidimensional Model

```
Iteration: 1, Log-Lik: -6666.943, Max-Change: 2.64229
Iteration: 2, Log-Lik: -5775.218, Max-Change: 0.29082
Iteration: 3, Log-Lik: -5730.990, Max-Change: 0.20993
Iteration: 4, Log-Lik: -5701.905, Max-Change: 0.19018
Iteration: 5, Log-Lik: -5680.953, Max-Change: 0.16530
Iteration: 6, Log-Lik: -5665.510, Max-Change: 0.14111
Iteration: 7, Log-Lik: -5654.032, Max-Change: 0.12041
Iteration: 8, Log-Lik: -5645.481, Max-Change: 0.10499
Iteration: 9, Log-Lik: -5639.119, Max-Change: 0.09050
Iteration: 10, Log-Lik: -5634.405, Max-Change: 0.07736
Iteration: 11, Log-Lik: -5630.930, Max-Change: 0.06576
Iteration: 12, Log-Lik: -5628.384, Max-Change: 0.05564
Iteration: 13, Log-Lik: -5622.014, Max-Change: 0.00982
Iteration: 14, Log-Lik: -5621.975, Max-Change: 0.00598
Iteration: 15, Log-Lik: -5621.951, Max-Change: 0.00406
Iteration: 16, Log-Lik: -5621.907, Max-Change: 0.00206
Iteration: 17, Log-Lik: -5621.904, Max-Change: 0.00166
Iteration: 18, Log-Lik: -5621.903, Max-Change: 0.00134
Iteration: 19, Log-Lik: -5621.900, Max-Change: 0.00041
Iteration: 20, Log-Lik: -5621.900, Max-Change: 0.00027
Iteration: 21, Log-Lik: -5621.900, Max-Change: 0.00021
Iteration: 22, Log-Lik: -5621.900, Max-Change: 0.00008

Calculating information matrix...

Call:
mirt(data = py_resp_wide[mask_analytic_sample, -c(1:2)], model = model,
    SE = TRUE)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 22 EM iterations.
mirt version: 1.32.1
M-step optimizer: nlminb
EM acceleration: Ramsay
Number of rectangular quadrature: 61
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  19.5315

Log-posterior = -5621.898
Estimated parameters: 60
DIC = 11363.8
G2 (1073741763) = 6488.2, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

IRT Item Parameters

	a	b	g	u
1314	1.255	-1.297	0	1
1303	1.012	-1.893	0	1
1311	1.088	-0.021	0	1
1296	1.106	-2.219	0	1
1308	1.402	-1.542	0	1
1302	1.426	-0.744	0	1
1297	1.620	-1.413	0	1
1310	1.259	-1.674	0	1
1307	1.387	-1.534	0	1
1306	0.810	-1.416	0	1
1316	1.294	-1.702	0	1
1300	0.980	-0.891	0	1
1325	0.607	1.032	0	1

	a	b	g	u
1319	1.334	-0.852	0	1
1298	0.950	-1.081	0	1
1317	0.703	-1.750	0	1
1324	0.448	-1.017	0	1
1321	0.861	-2.448	0	1
1323	0.620	-0.928	0	1
1313	0.547	-0.854	0	1
1318	1.255	-3.106	0	1
1309	1.785	-0.042	0	1
1299	0.697	-2.348	0	1
1322	0.574	0.236	0	1
1304	0.702	-0.568	0	1
1315	0.699	-3.429	0	1
1301	0.714	-1.814	0	1
1320	0.617	0.446	0	1
1312	1.193	-0.791	0	1
1305	0.968	-2.613	0	1

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

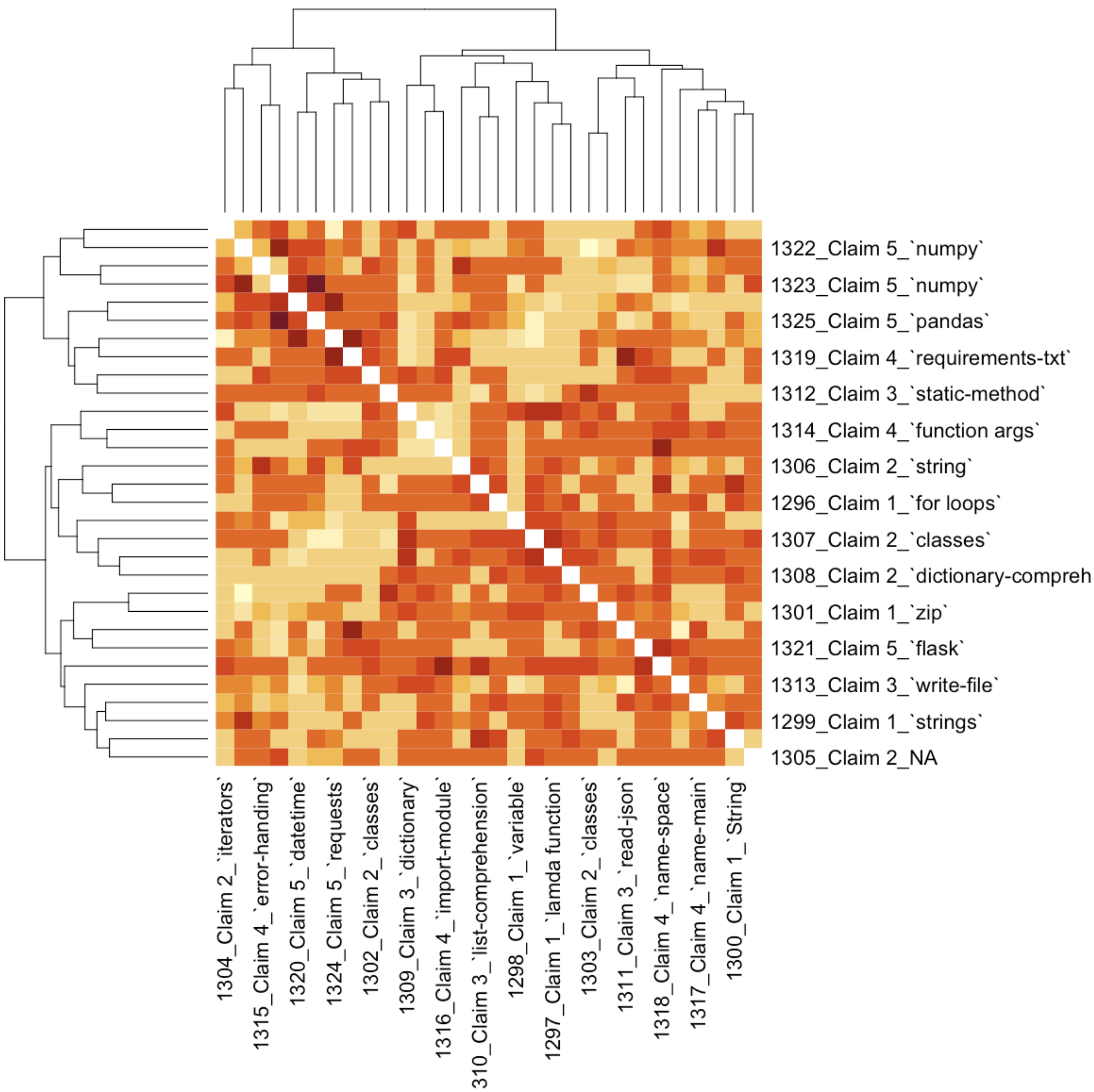
claim	name	target	item	S_X2	df.S_X2	RMSEA.S_X2	p.S_X2	flag
Claim 4	lang_python_19	`function args`	1314	12.469	16	0.000	0.711	
Claim 2	lang_python_08	`classes`	1303	21.259	17	0.025	0.215	
Claim 3	lang_python_16	`read-json`	1311	18.999	14	0.030	0.165	
Claim 1	lang_python_01	`for loops`	1296	10.124	16	0.000	0.860	
Claim 2	lang_python_13	`dictionary-comprehension`	1308	12.141	15	0.000	0.668	
Claim 2	lang_python_07	`classes`	1302	12.187	15	0.000	0.665	
Claim 1	lang_python_02	`lamda function`	1297	14.605	15	0.000	0.480	
Claim 3	lang_python_15	`list-comprehension`	1310	18.852	15	0.026	0.220	
Claim 2	lang_python_12	`classes`	1307	31.498	15	0.053	0.008	<=
Claim 2	lang_python_11	`string`	1306	9.343	17	0.000	0.929	
Claim 4	lang_python_21	`import-module`	1316	14.703	15	0.000	0.473	
Claim 1	lang_python_05	`String`	1300	18.774	17	0.016	0.342	
Claim 5	lang_python_30	`pandas`	1325	21.868	16	0.031	0.147	
Claim 4	lang_python_24	`requirements-txt`	1319	11.348	15	0.000	0.728	
Claim 1	lang_python_03	`variable`	1298	21.760	17	0.027	0.194	
Claim 4	lang_python_22	`name-main`	1317	20.987	18	0.021	0.280	
Claim 5	lang_python_29	`requests`	1324	22.347	19	0.021	0.267	
Claim 5	lang_python_26	`flask`	1321	9.823	17	0.000	0.911	
Claim 5	lang_python_28	`numpy`	1323	23.426	17	0.031	0.136	
Claim 3	lang_python_18	`write-file`	1313	27.796	17	0.040	0.047	<=
Claim 4	lang_python_23	`name-space`	1318	9.235	6	0.037	0.161	
Claim 3	lang_python_14	`dictionary`	1309	11.257	12	0.000	0.507	
Claim 1	lang_python_04	`strings`	1299	6.918	18	0.000	0.991	

claim	name	target	item	S_X2	df.S_X2	RMSEA.S_X2	p.S_X2	flag
Claim 5	lang_python_27	`numpy`	1322	26.147	17	0.037	0.072	
Claim 2	lang_python_09	`iterators`	1304	15.889	17	0.000	0.532	
Claim 4	lang_python_20	`error-handing`	1315	20.123	16	0.026	0.215	
Claim 1	lang_python_06	`zip`	1301	19.058	18	0.012	0.388	
Claim 5	lang_python_25	`datetime`	1320	22.801	16	0.033	0.119	
Claim 3	lang_python_17	`static-method`	1312	22.723	16	0.033	0.121	
Claim 2	lang_python_10	NA	1305	9.999	16	0.000	0.867	

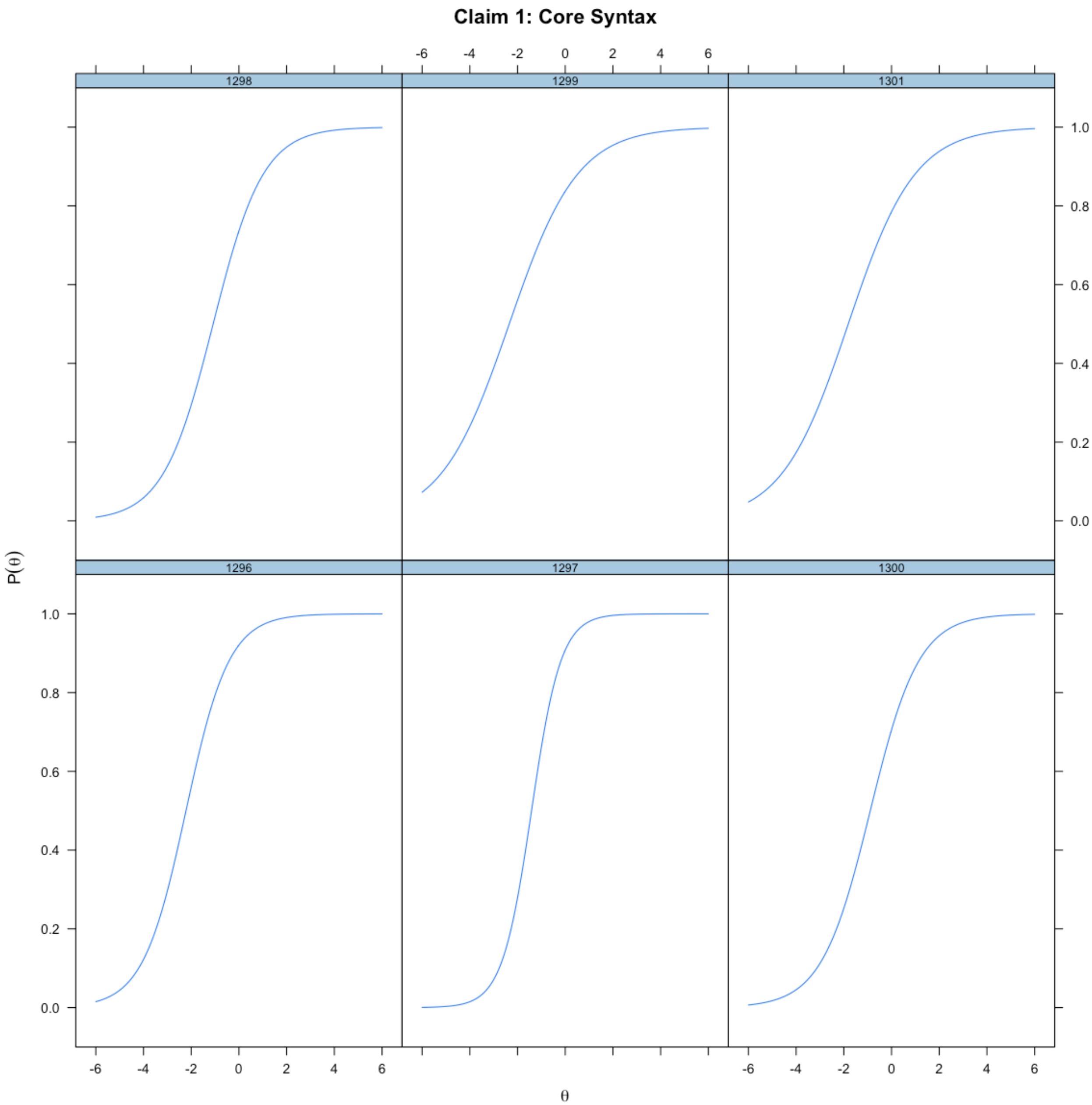
Items are flagged with ‘<=’ to be reviewed or removed if *p-value* < 0.05. This indicates poor fit of 2PL model.

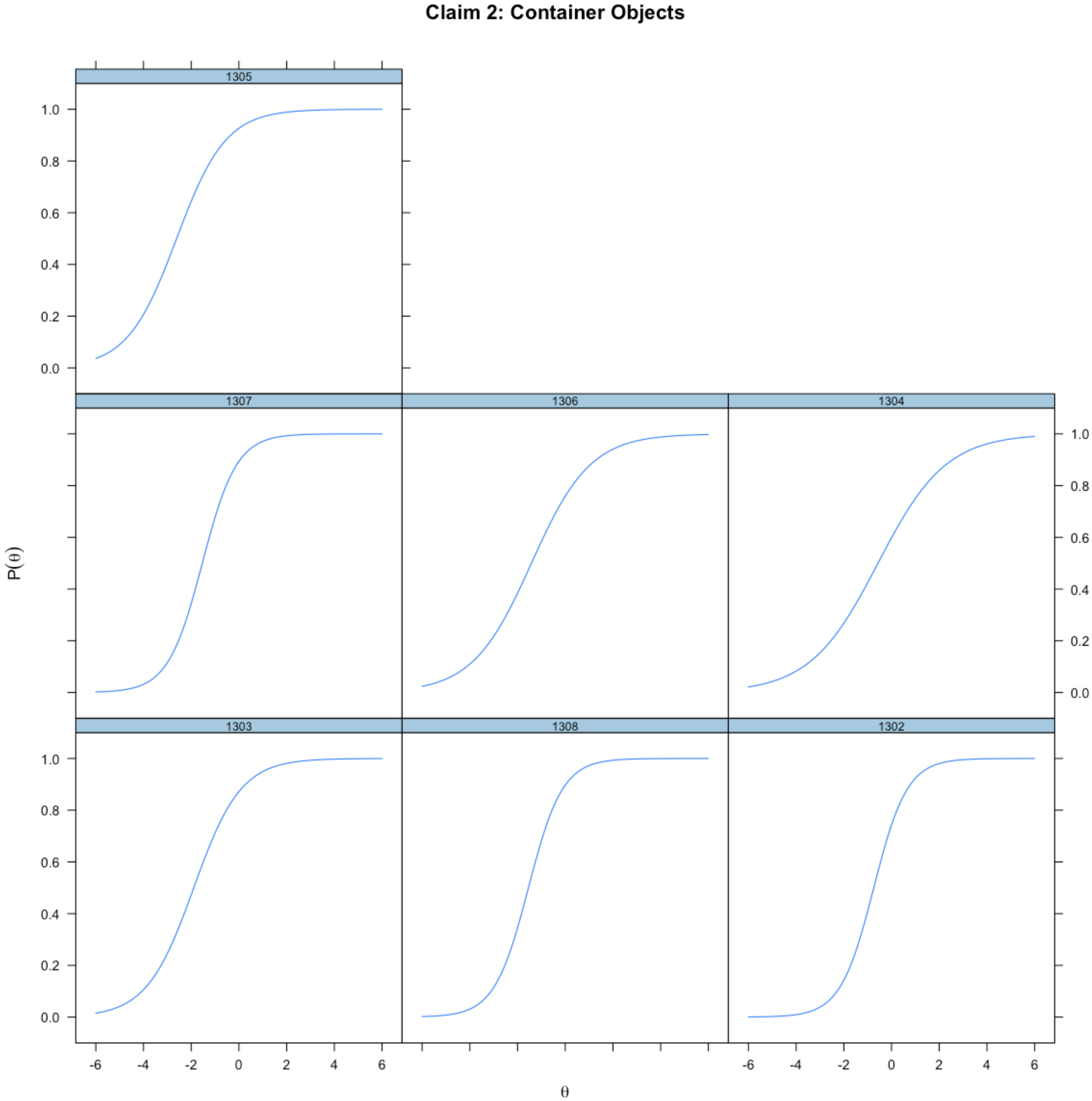
IRT Plots

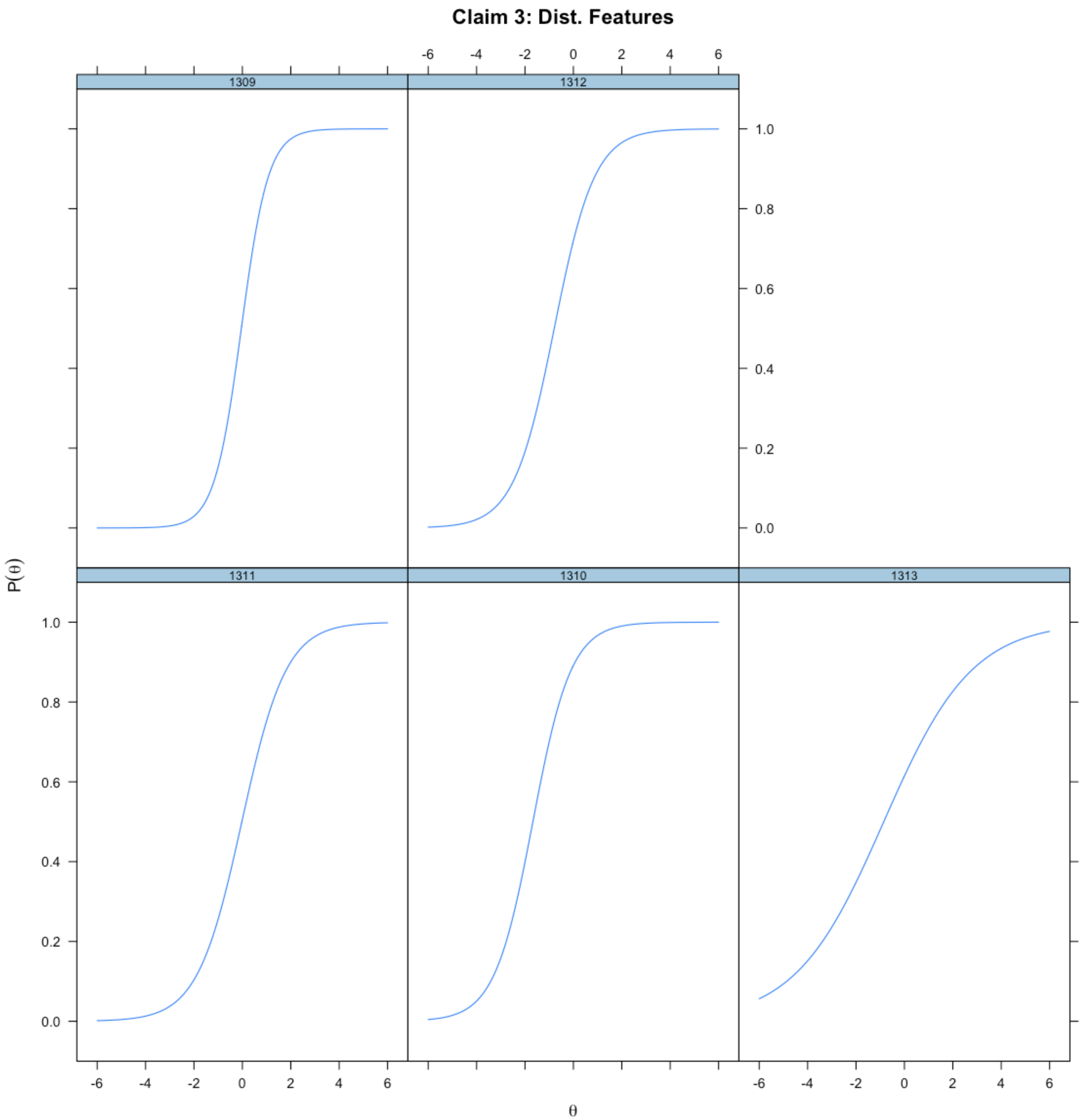
Item Local-Dependence Plots - Residual Dependencies given a unidimensional model:

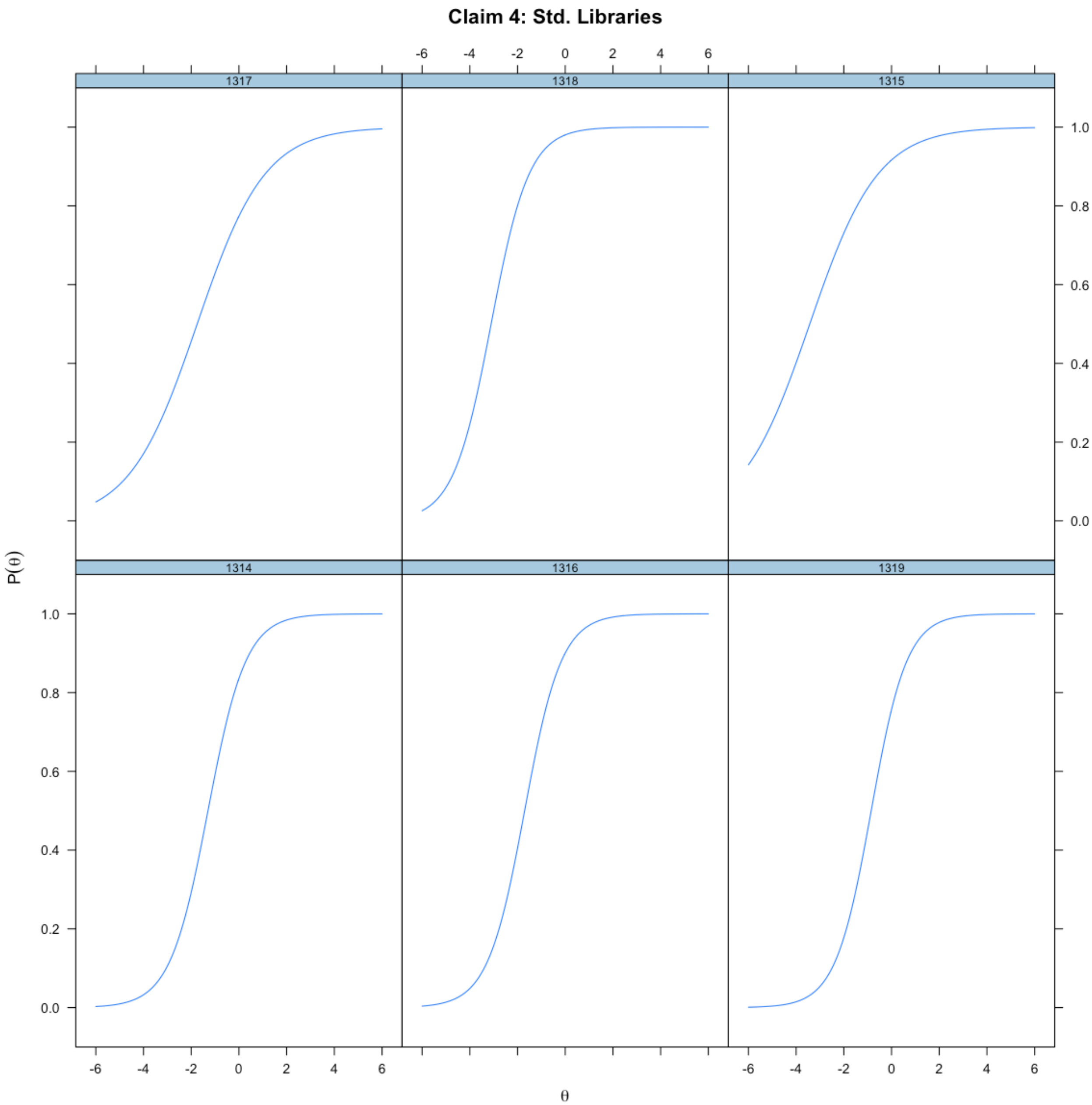


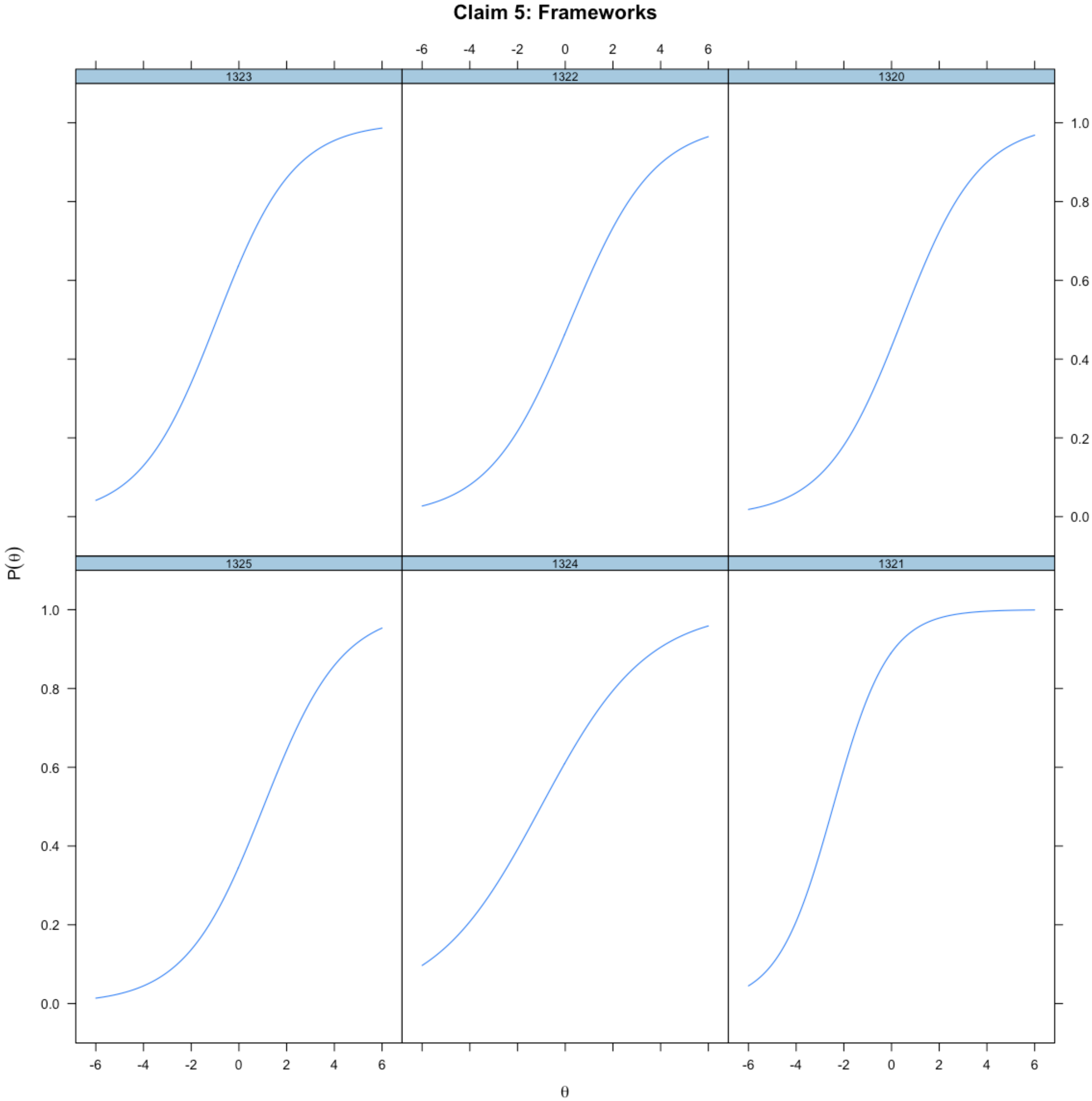
Item Characteristic Curves (Tracelines), by Claims


Claim 1: Core Syntax
1298
1299
1301
1296
1297
1300
P(θ)
θ
-6
-4
-2
0
2
4
6
0.0
0.2
0.4
0.6
0.8
1.0

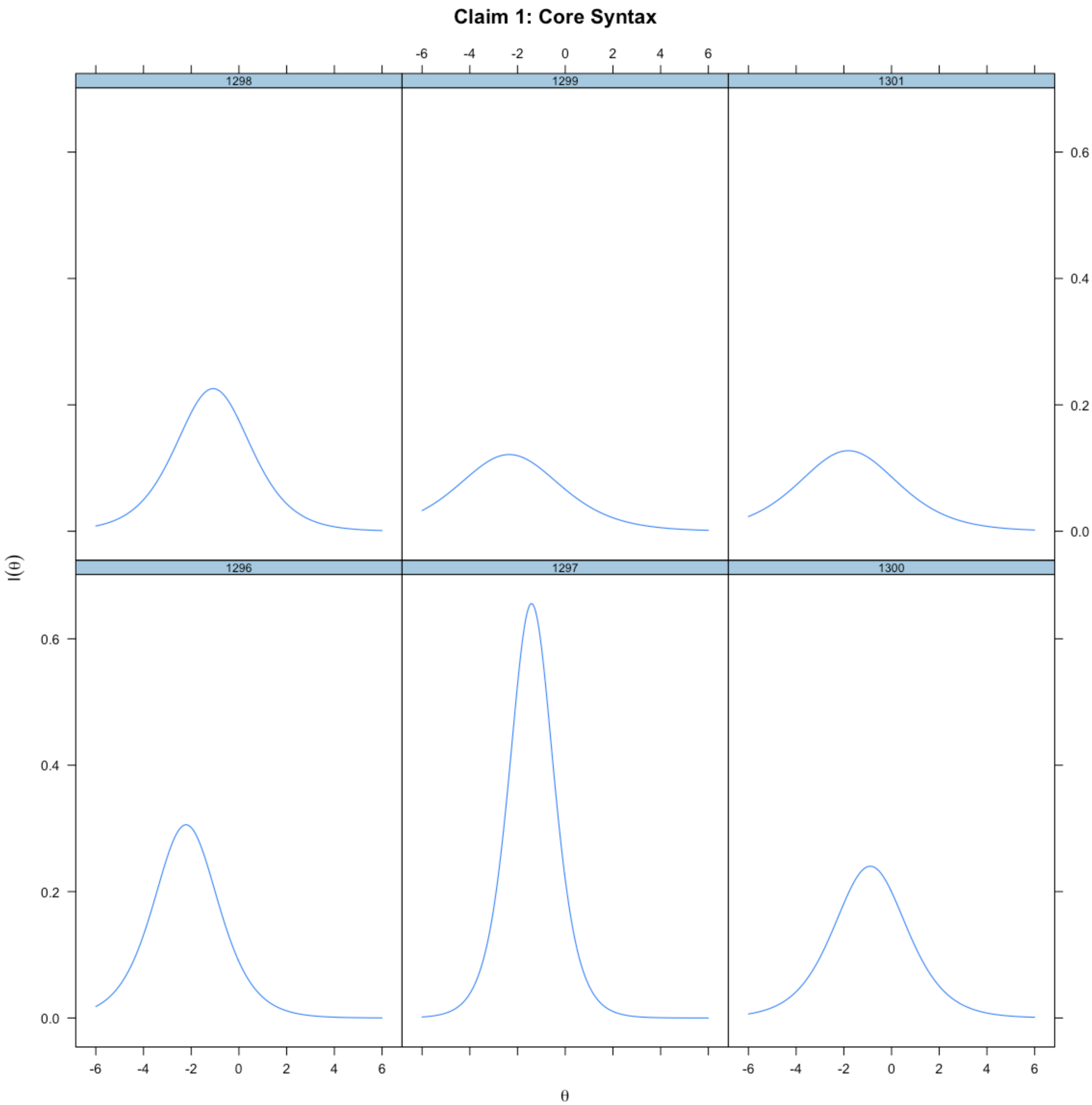

Claim 2: Container Objects
1305
1307
1306
1304
1303
1308
1302
1.0
0.8
0.6
0.4
0.2
0.0
-6
-4
-2
0
2
4
6
P(θ)
θ

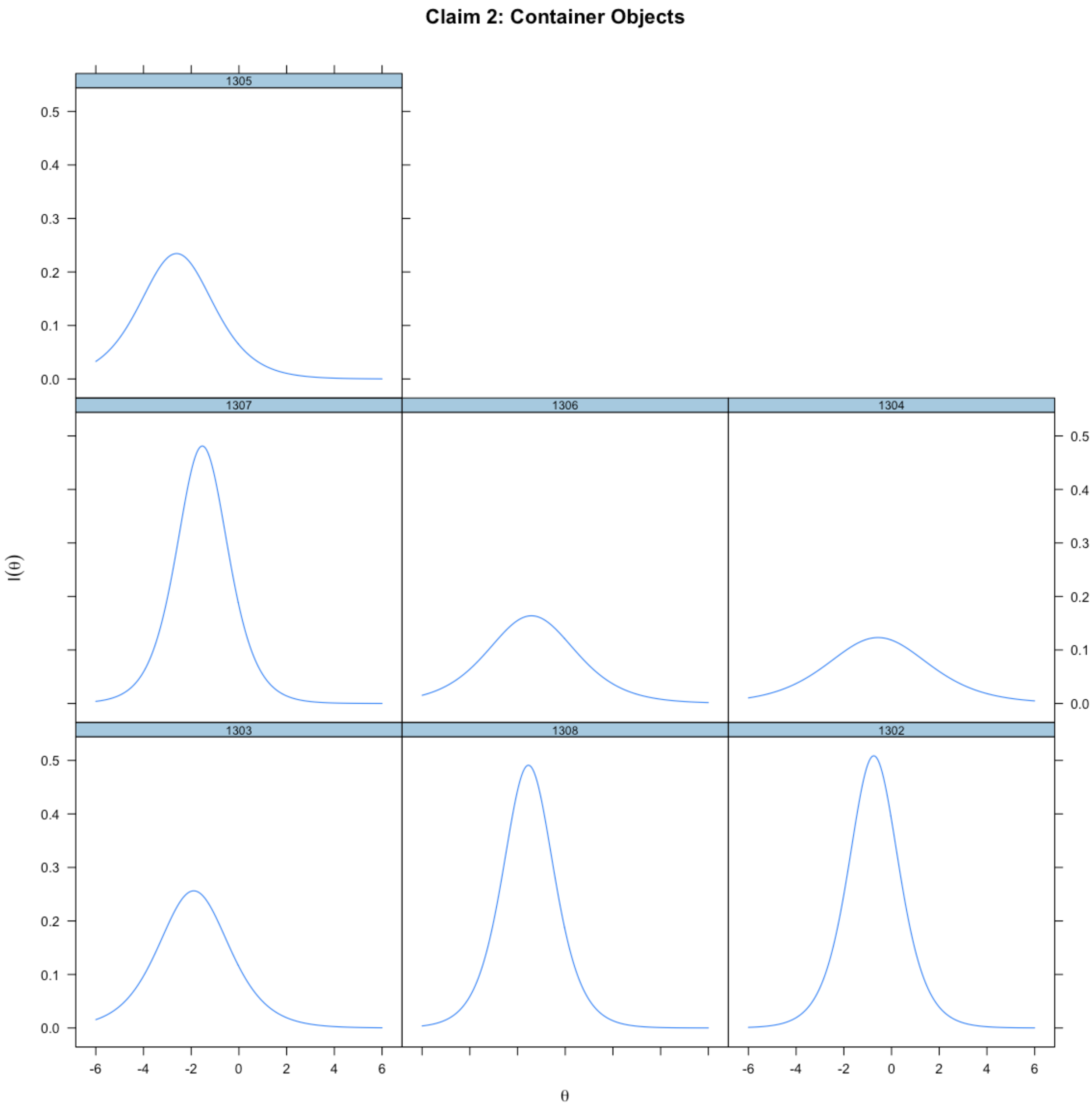

Claim 3: Dist. Features
1309
1312
1311
1310
1313
P(θ)
θ
-6
-4
-2
0
2
4
6
0.0
0.2
0.4
0.6
0.8
1.0


Claim 4: Std. Libraries
1317
1318
1315
1314
1316
1319
P(θ)
θ
-6
-4
-2
0
2
4
6
0.0
0.2
0.4
0.6
0.8
1.0

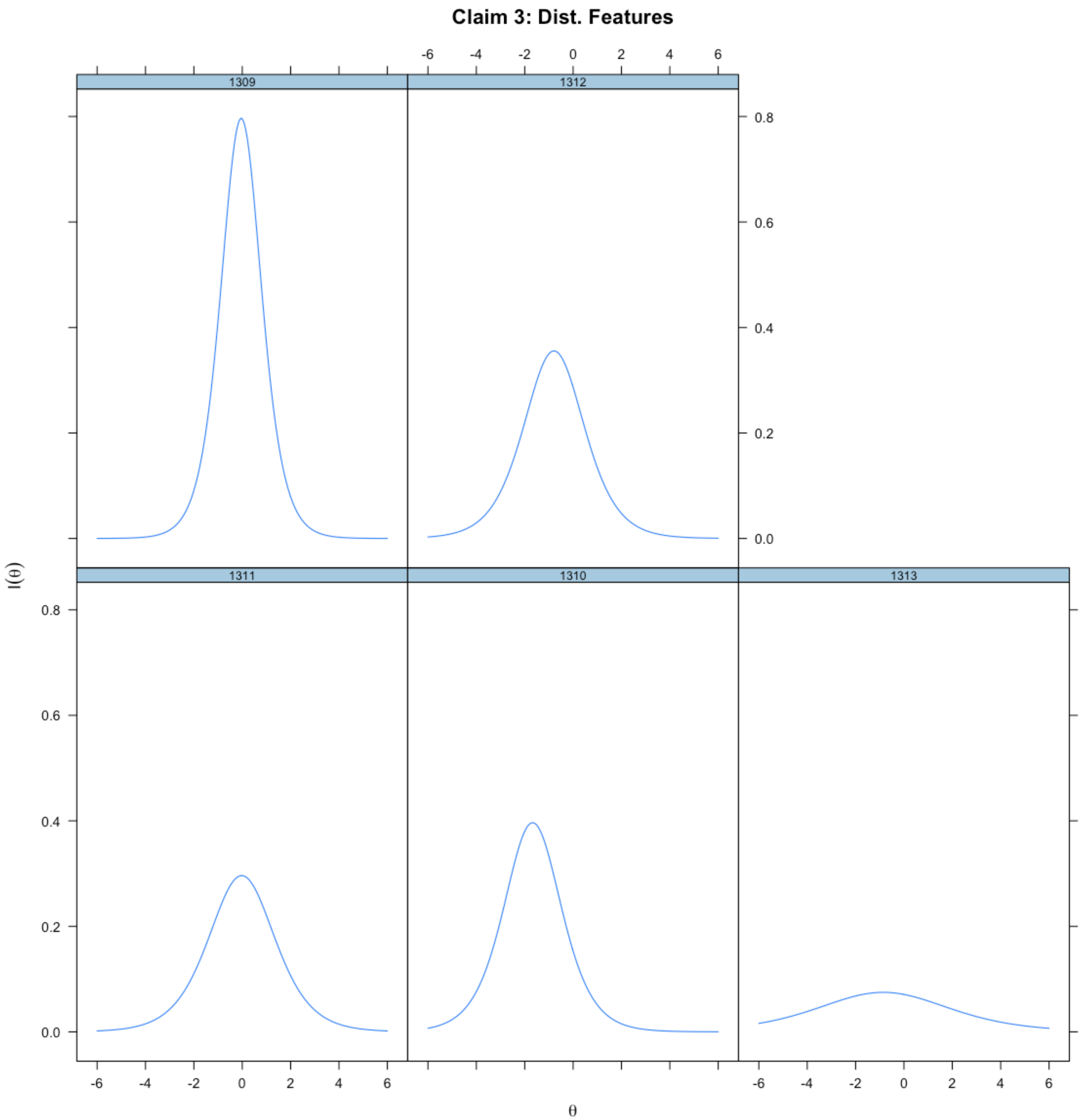


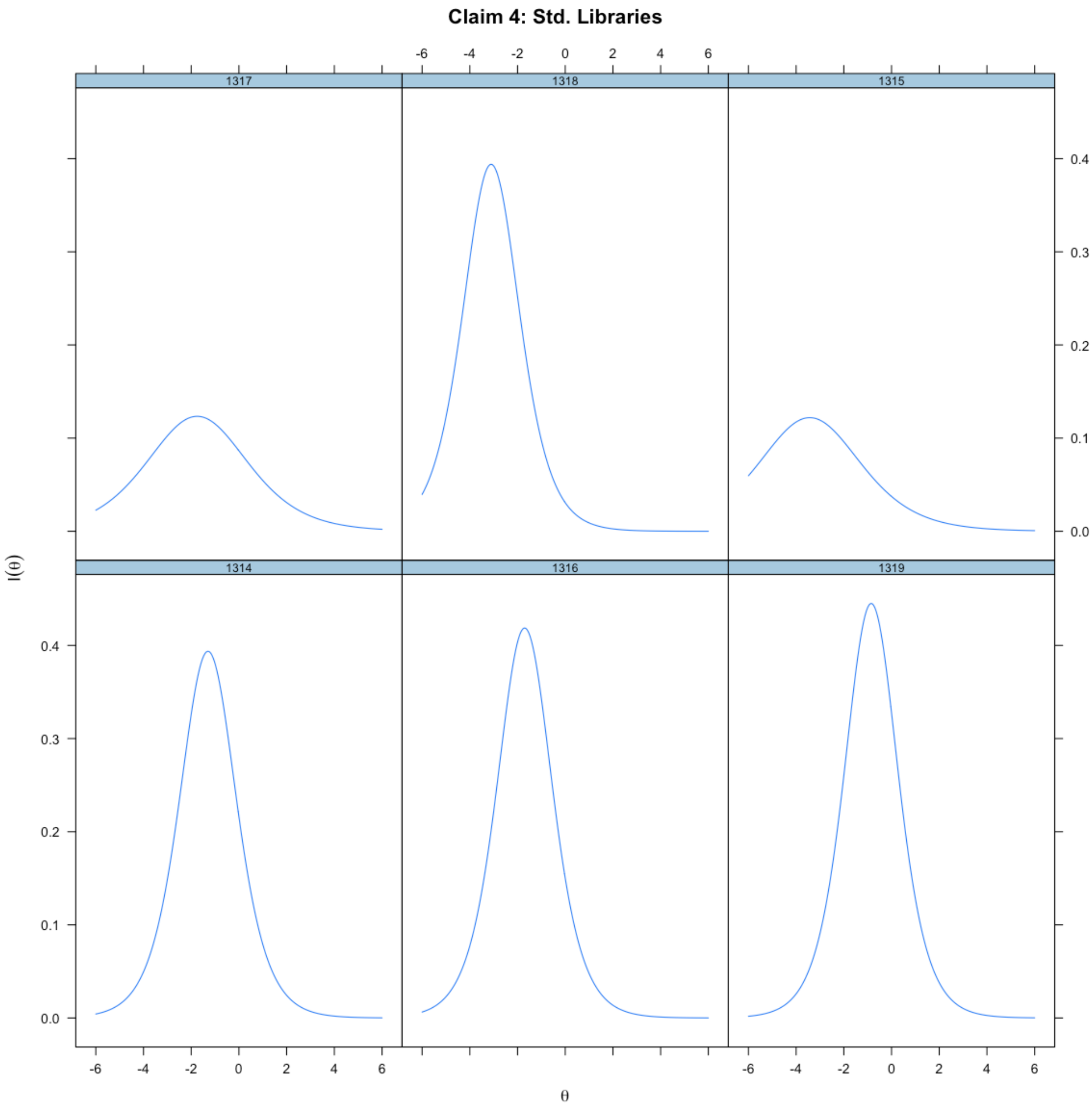
Item Information Curves by Claims

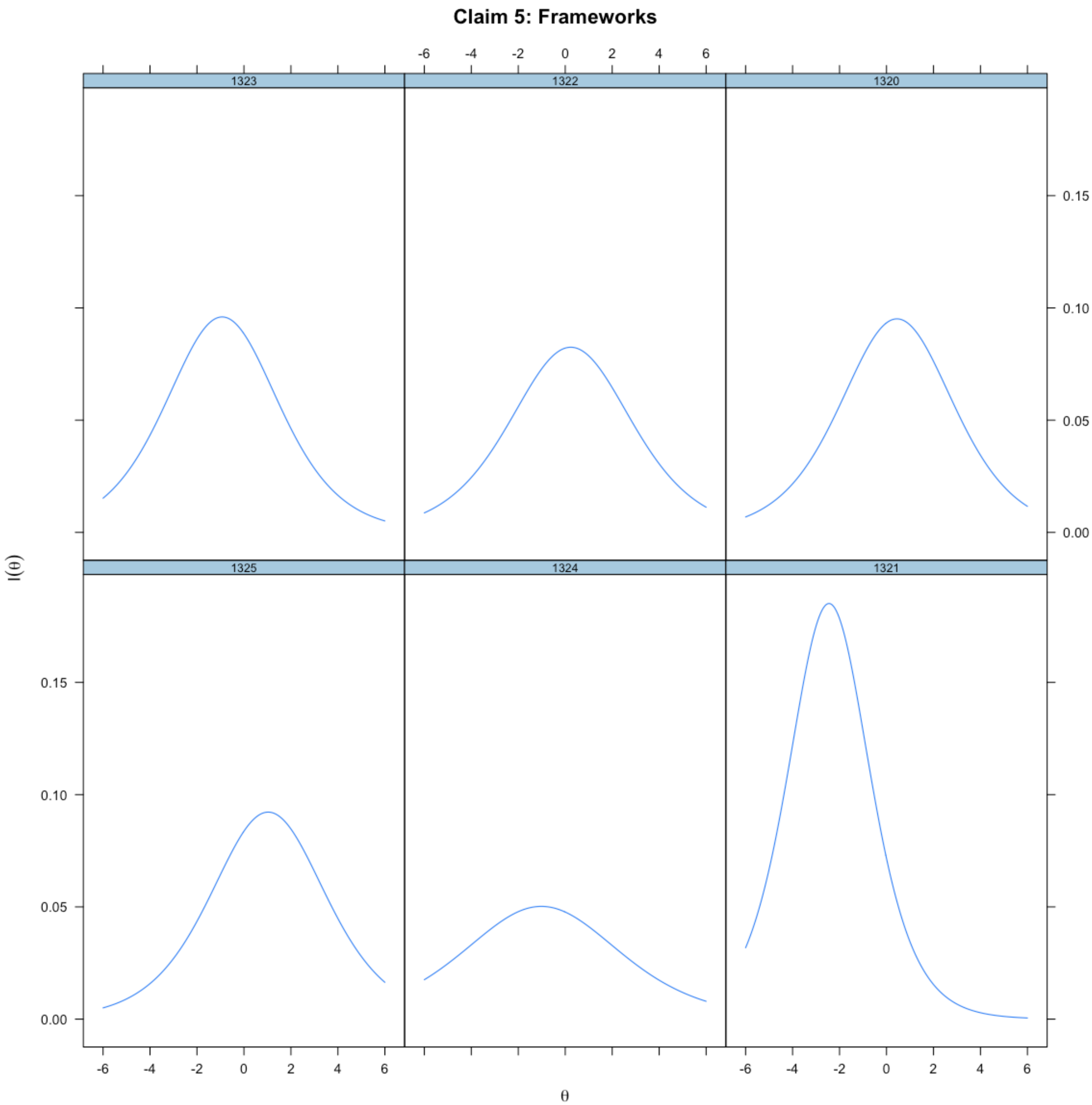

Claim 1: Core Syntax
1298
1299
1301
1296
1297
1300
I(θ)
θ
-6
-4
-2
0
2
4
6
0.0
0.2
0.4
0.6



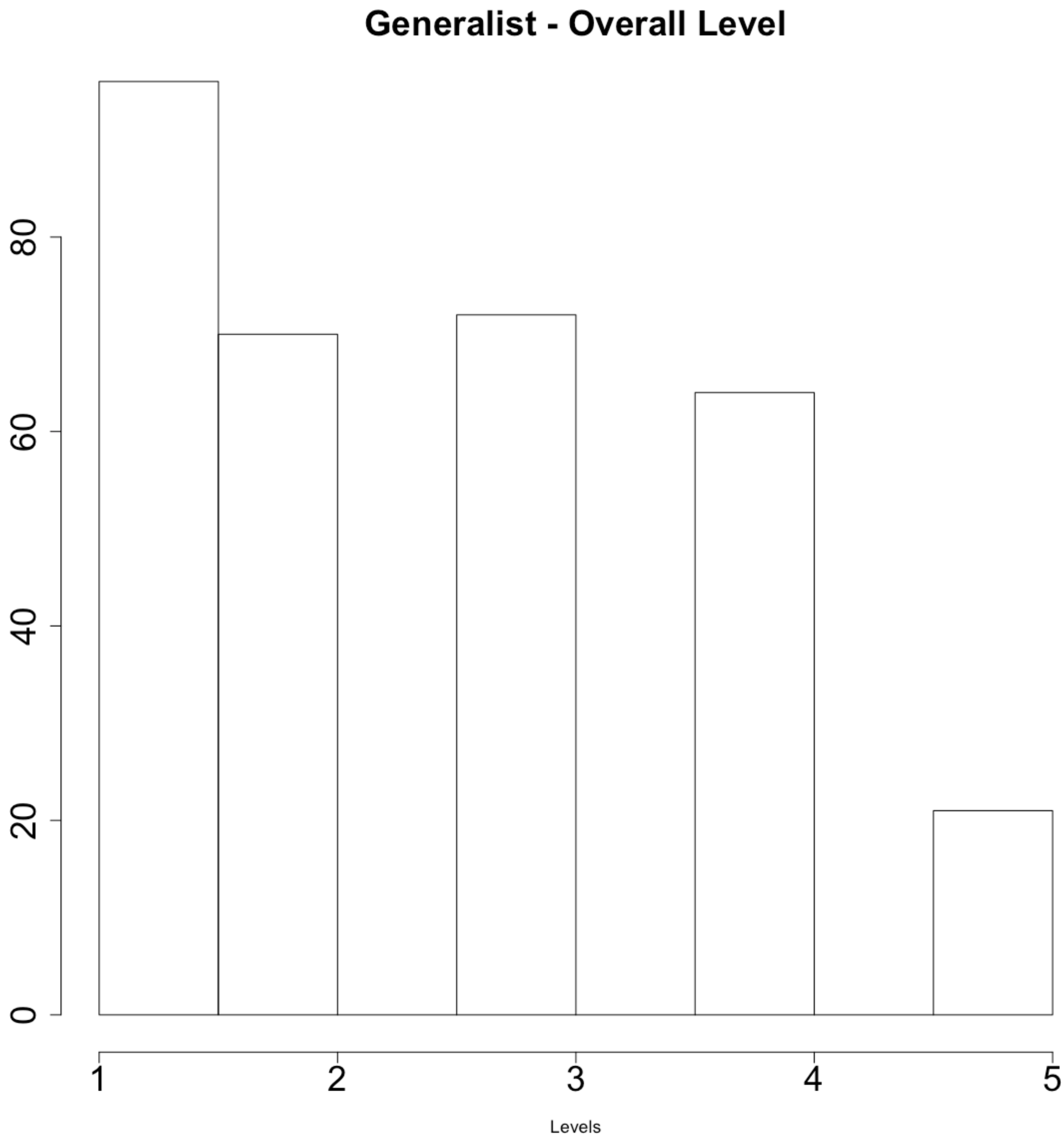
Claim 2: Container Objects
1305
1307
1306
1304
1303
1308
1302
I(θ)
θ
0.0
0.1
0.2
0.3
0.4
0.5
-6
-4
-2
0
2
4
6

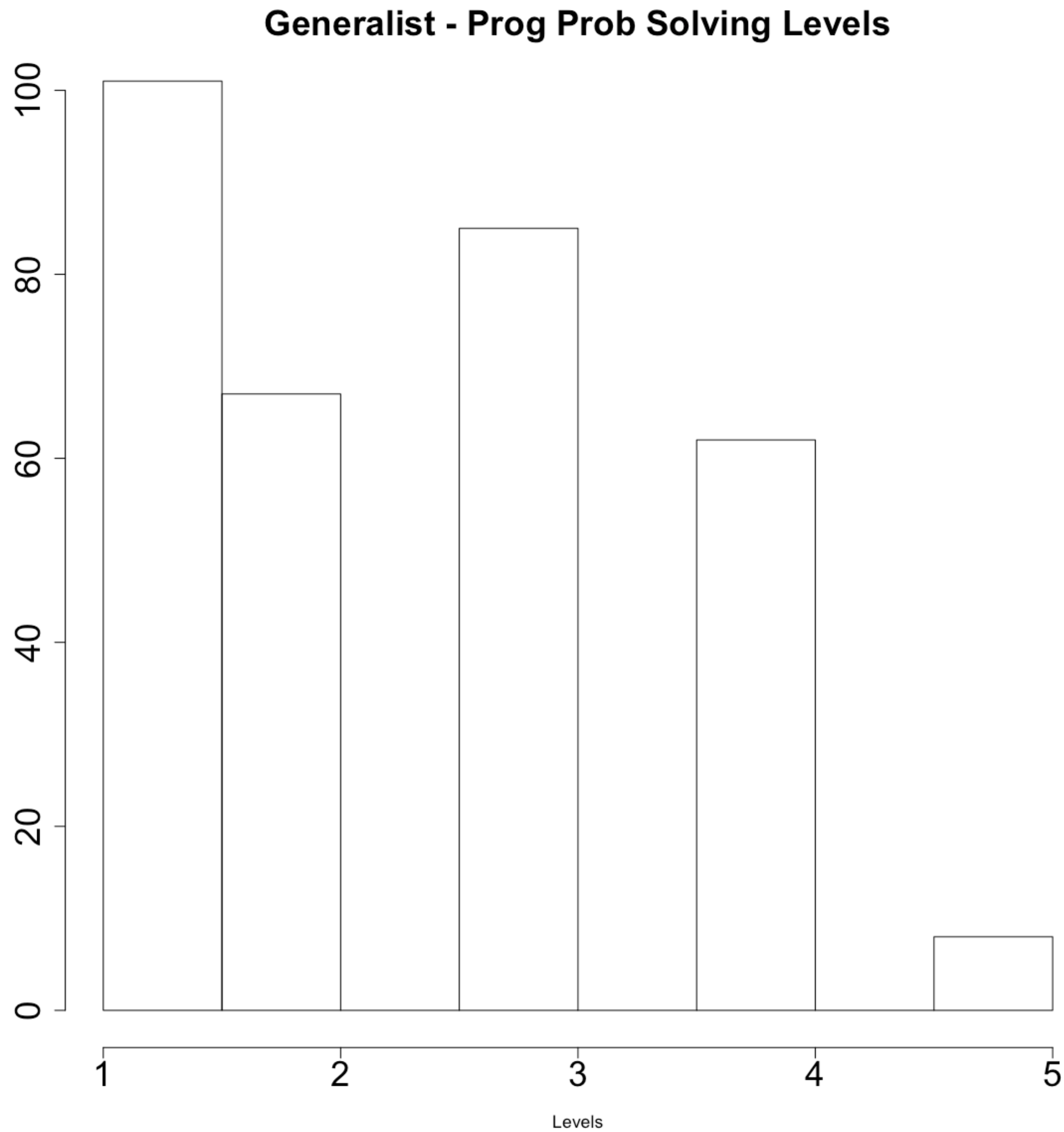

Claim 3: Dist. Features
1309
1312
1311
1310
1313
I(θ)
θ
0.0
0.2
0.4
0.6
0.8
-6
-4
-2
0
2
4
6


Claim 4: Std. Libraries
1317
1318
1315
1314
1316
1319
I(θ)
θ
-6
-4
-2
0
2
4
6
0.0
0.1
0.2
0.3
0.4



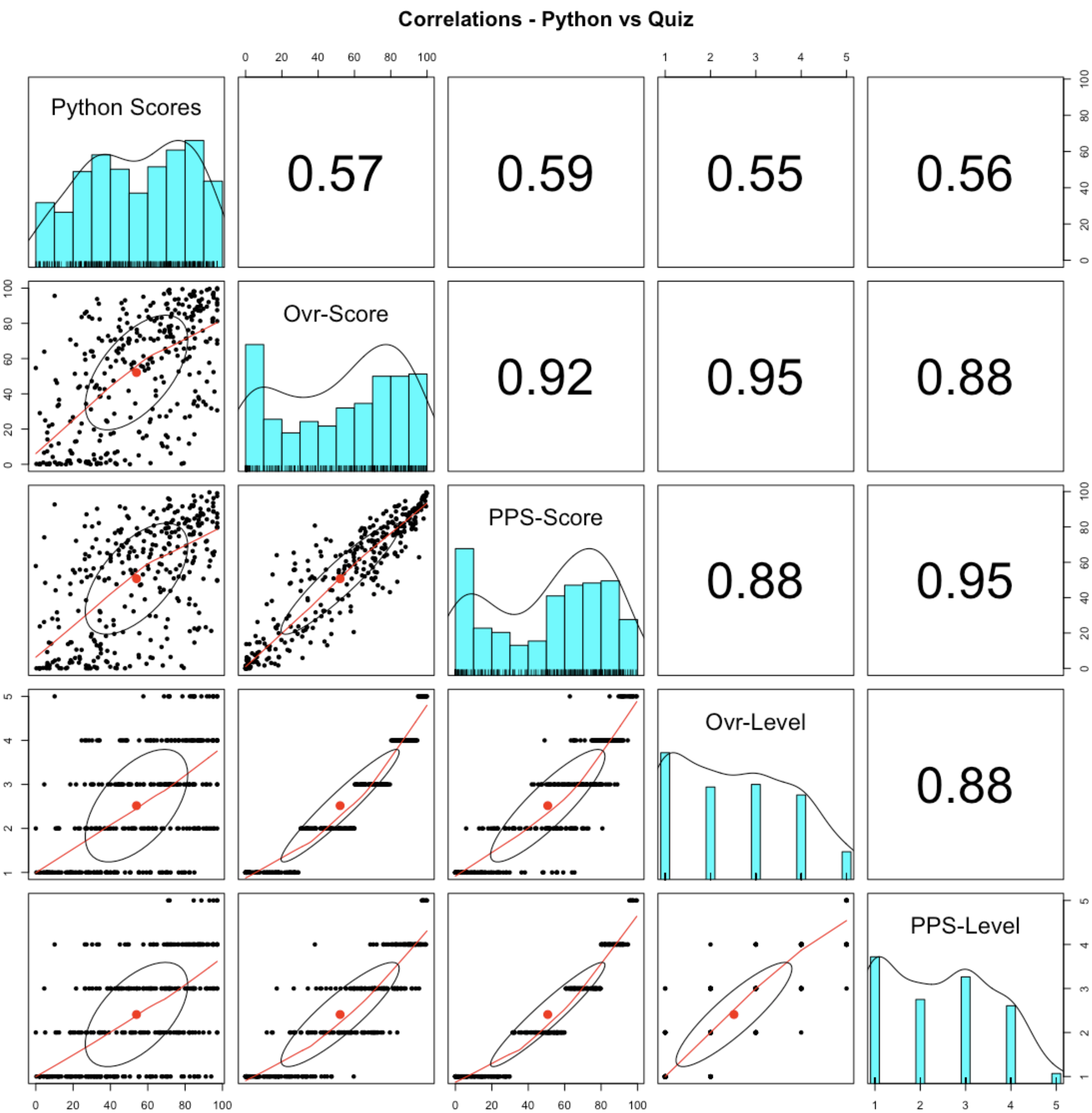
Calibration Sample Characteristics


Generalist - Overall Level
0
20
40
60
80
1
2
3
4
5
Levels

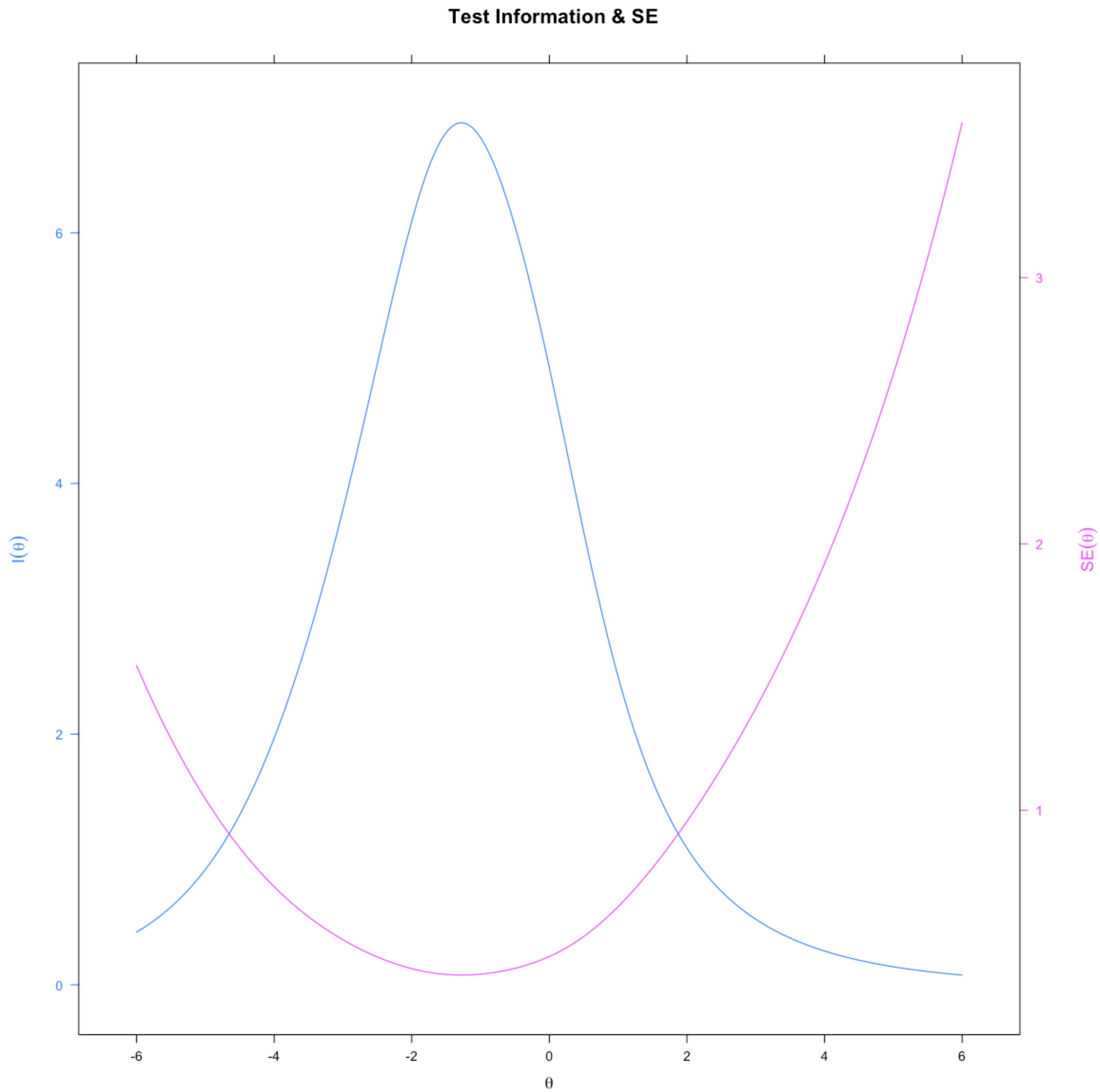


Scoring

Distributions & Correlations


Correlations - Python vs Quiz
Python Scores
Ovr-Score
PPS-Score
Ovr-Level
PPS-Level
0.57
0.59
0.55
0.56
0.92
0.95
0.88
0.88
0.95
0.88

Avg. SE



Equating/Linking Scores

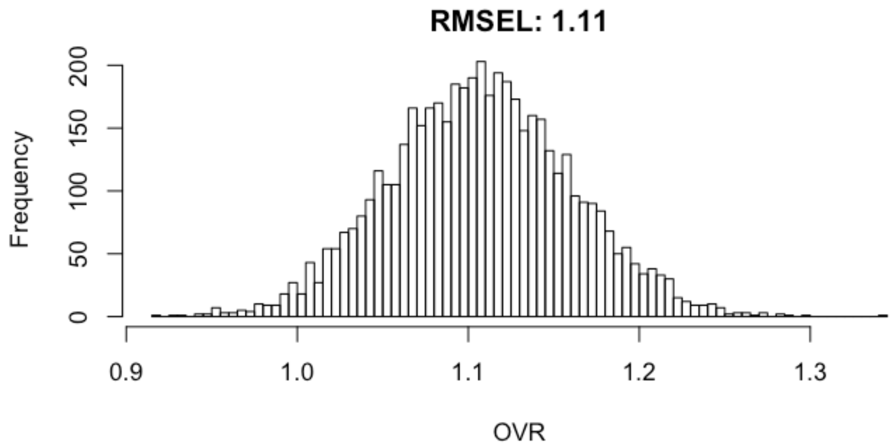
In order to get langauge assessment SkillEstimates on the same scale as the the other quizzes, it requires being able to produce item parameters or IRT-ability estimates on the same scale as the other assessments.

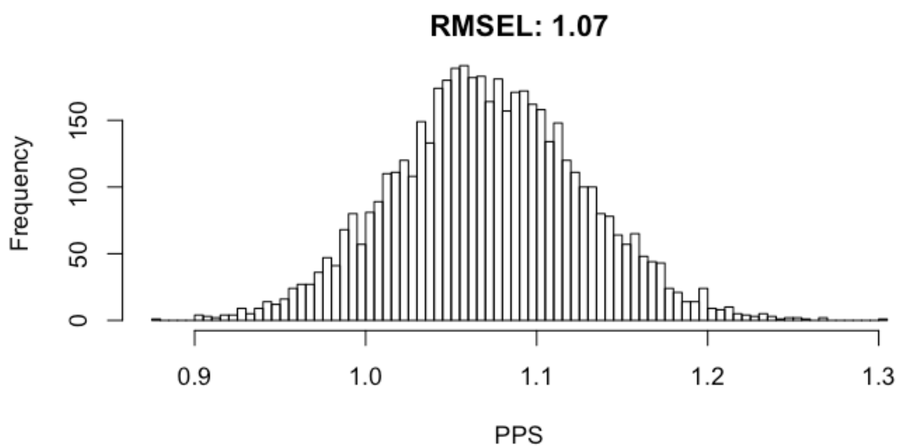
The equating/linking design available is a *Single Group* Design where the same candidates have taken both a previous core quiz and a language assessement. Therefore, there are limited methodolgies available to link the two scales together (plus with a limited sample size).

Three approaches explored are:

- Mean Method
- Linear Method
- Equipercentile Method

Mean Method






```
$OVR
```

avg <dbl>	sd <dbl>
1.106903	0.05316797

1 row

```
$PPS
```

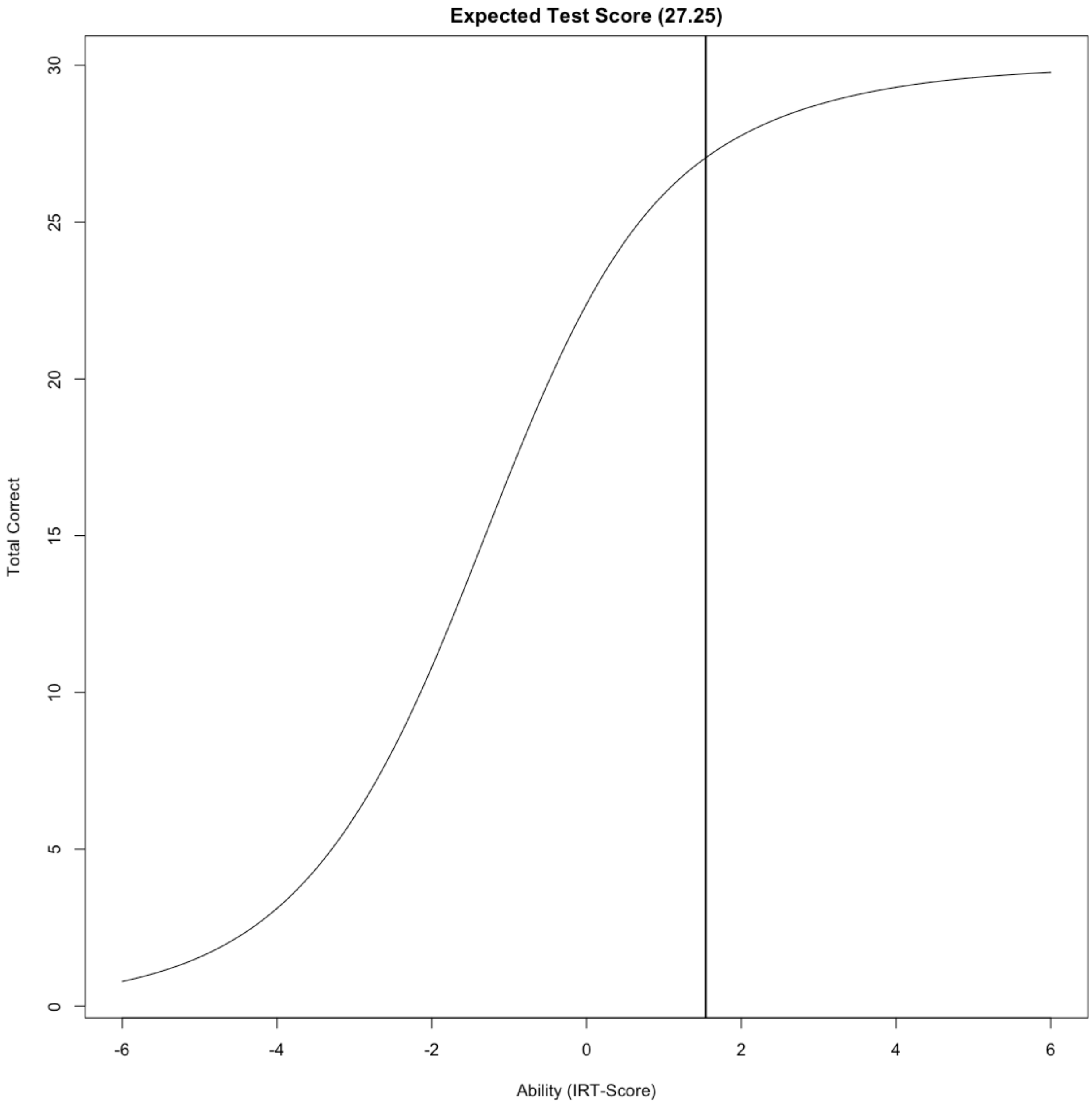
avg <dbl>	sd <dbl>
1.071559	0.05558107

1 row

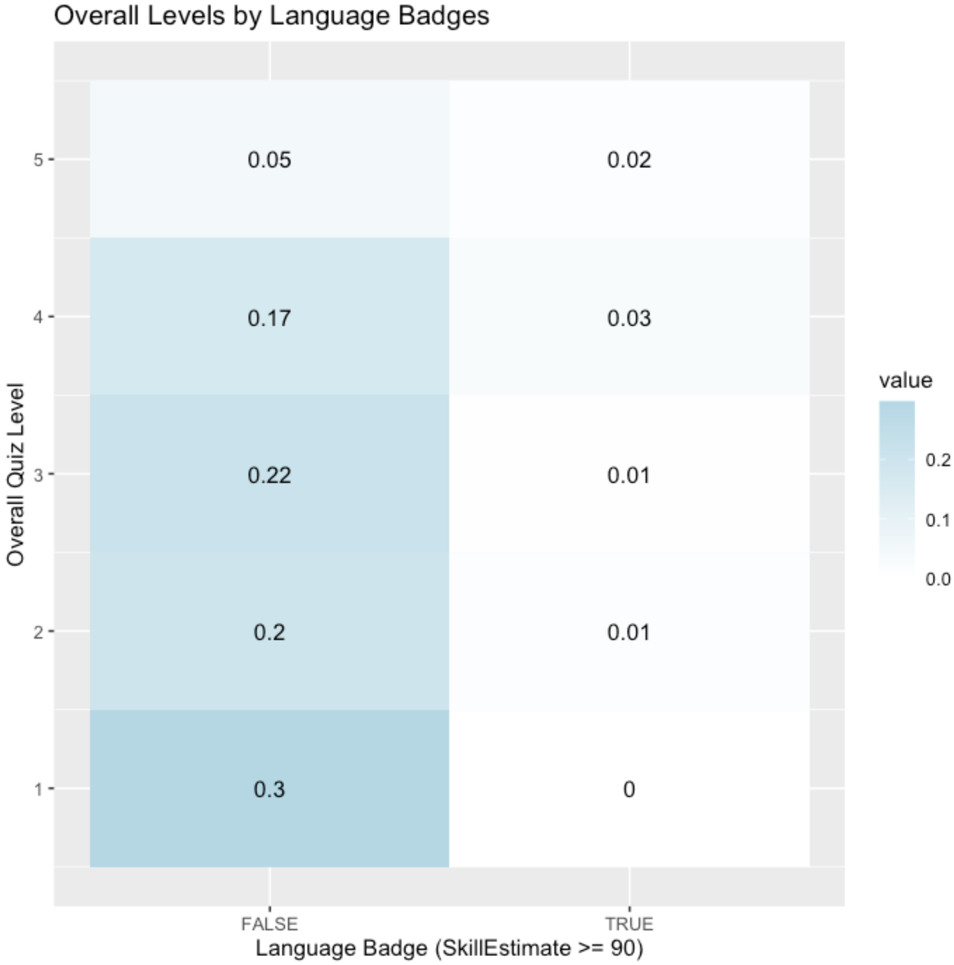
```
NA
```

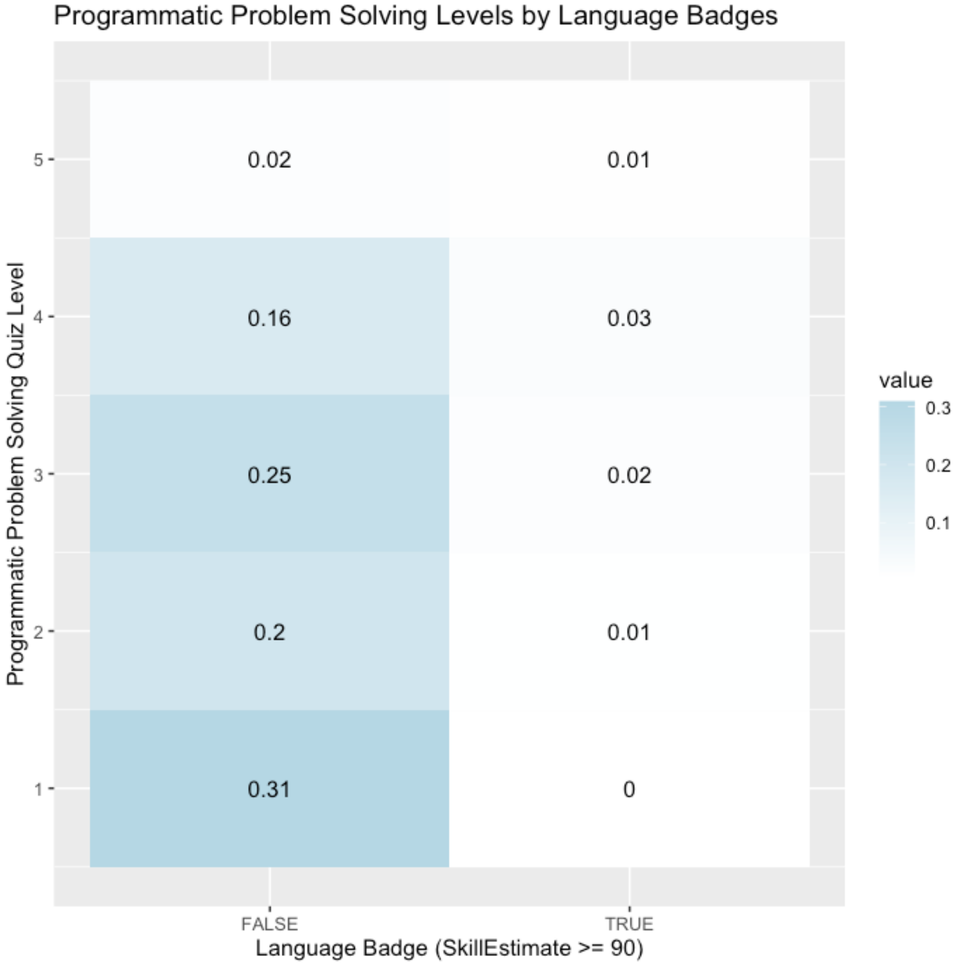
Result: Programmatic Problem Solving Scales score produce smaller linking error

Badges - Test Characteristic Curve

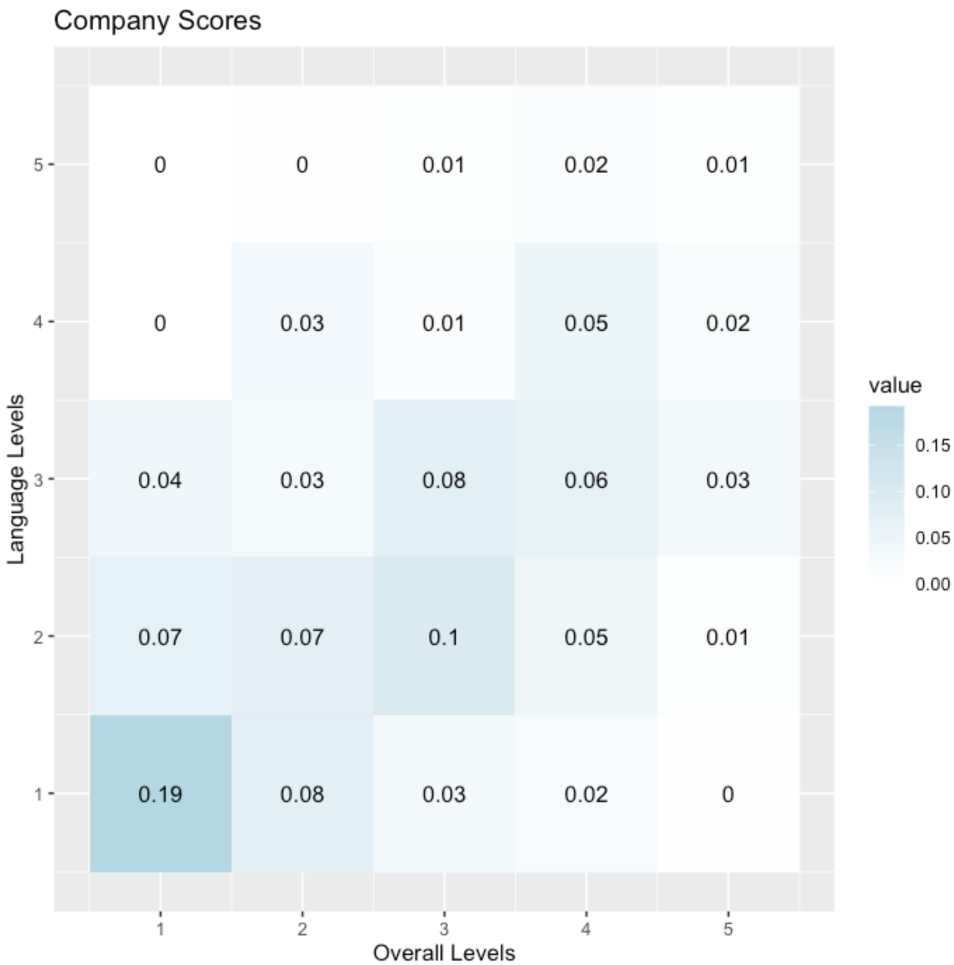


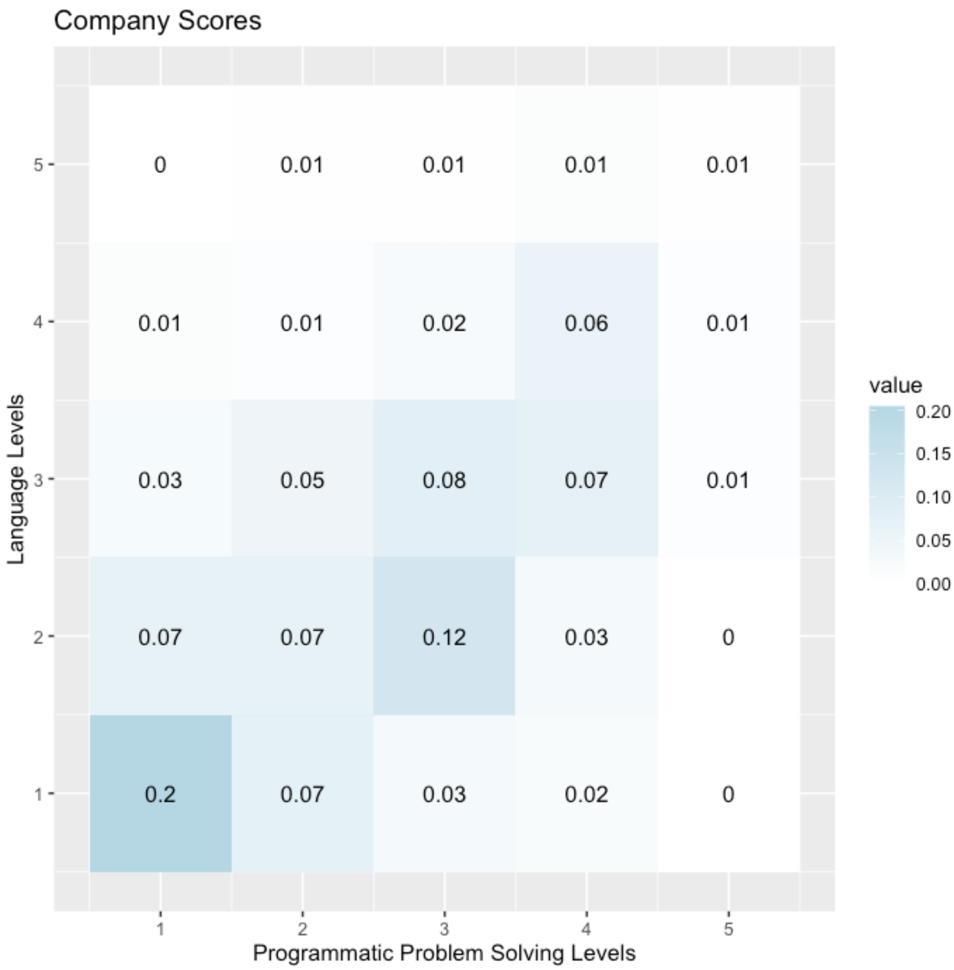
Badges - Quiz Level, by Badge


Overall Levels by Language Badges
0.05
0.02
0.17
0.03
0.22
0.01
0.2
0.01
0.3
0
Overall Quiz Level
1
2
3
4
5
FALSE
TRUE
Language Badge (SkillEstimate >= 90)
value
0.2
0.1
0.0

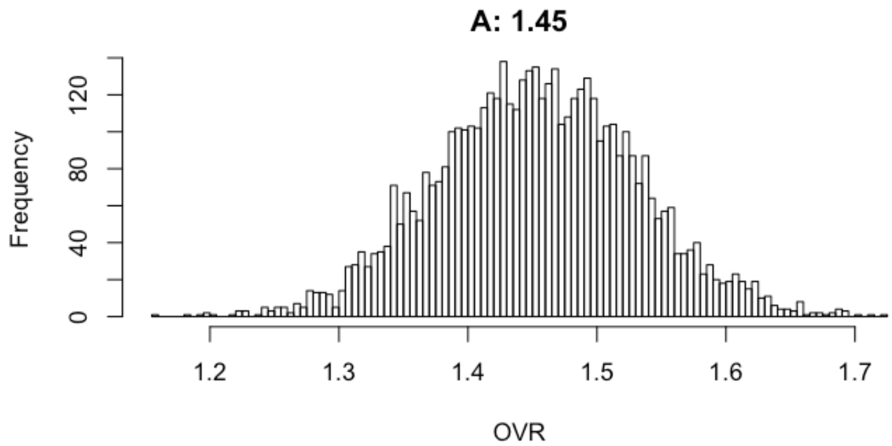


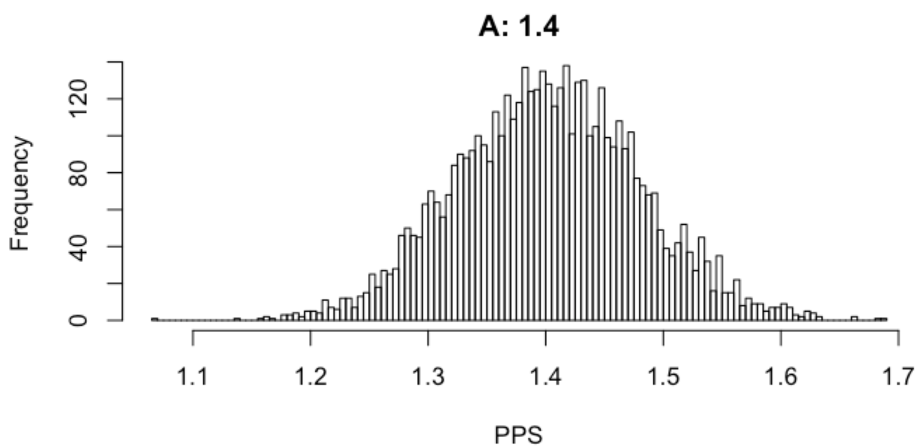
Language Levels, Quiz Levels

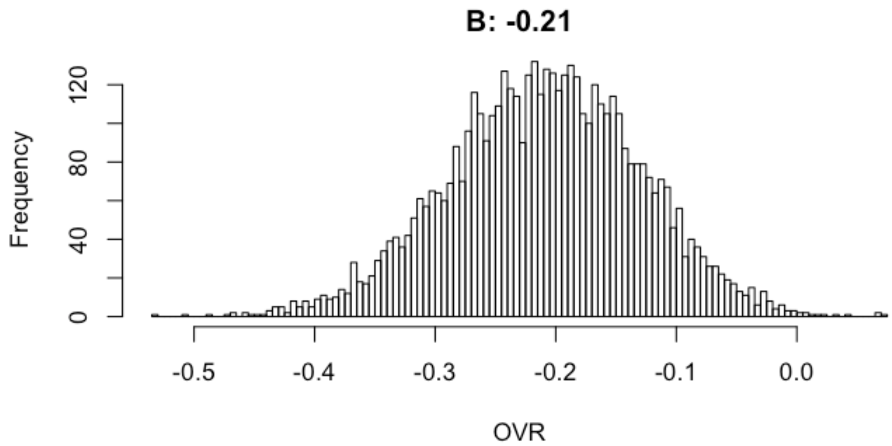

Company Scores
Language Levels
Overall Levels
value
0.15
0.10
0.05
0.00
0 0 0.01 0.02 0.01
0 0.03 0.01 0.05 0.02
0.04 0.03 0.08 0.06 0.03
0.07 0.07 0.1 0.05 0.01
0.19 0.08 0.03 0.02 0

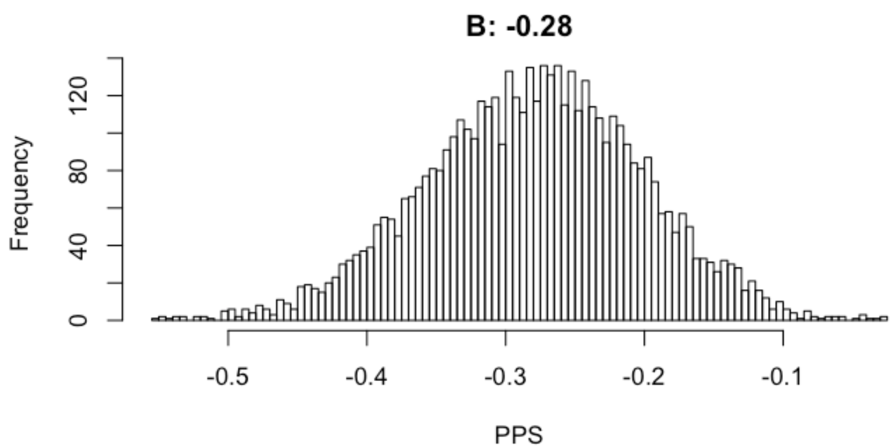


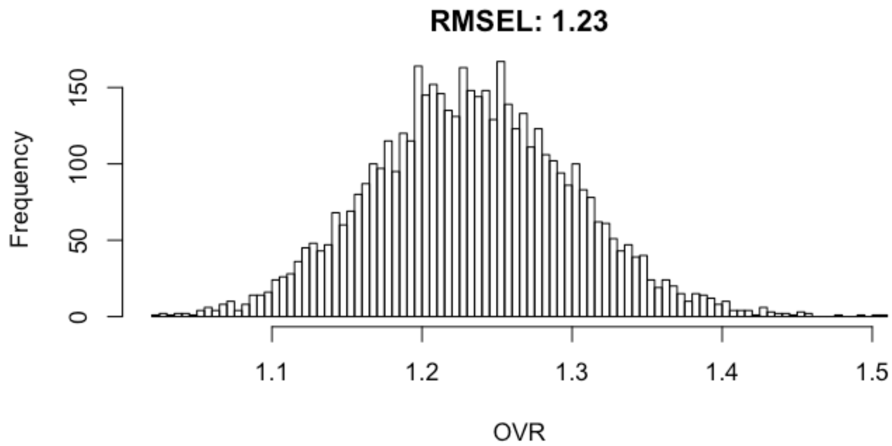
Linear Method

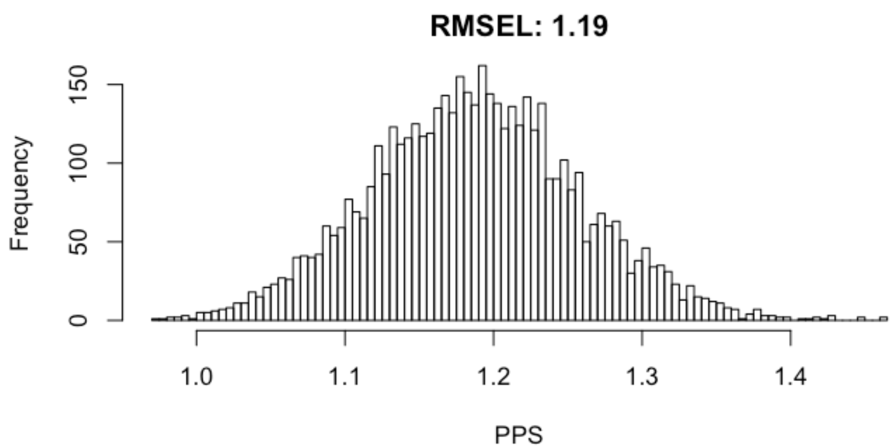

A: 1.45
Frequency
0
40
80
120
1.2
1.3
1.4
1.5
1.6
1.7
OVR


A: 1.4
Frequency
0
40
80
120
1.1
1.2
1.3
1.4
1.5
1.6
1.7
PPS


B: -0.21
Frequency
0
40
80
120
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
OVR

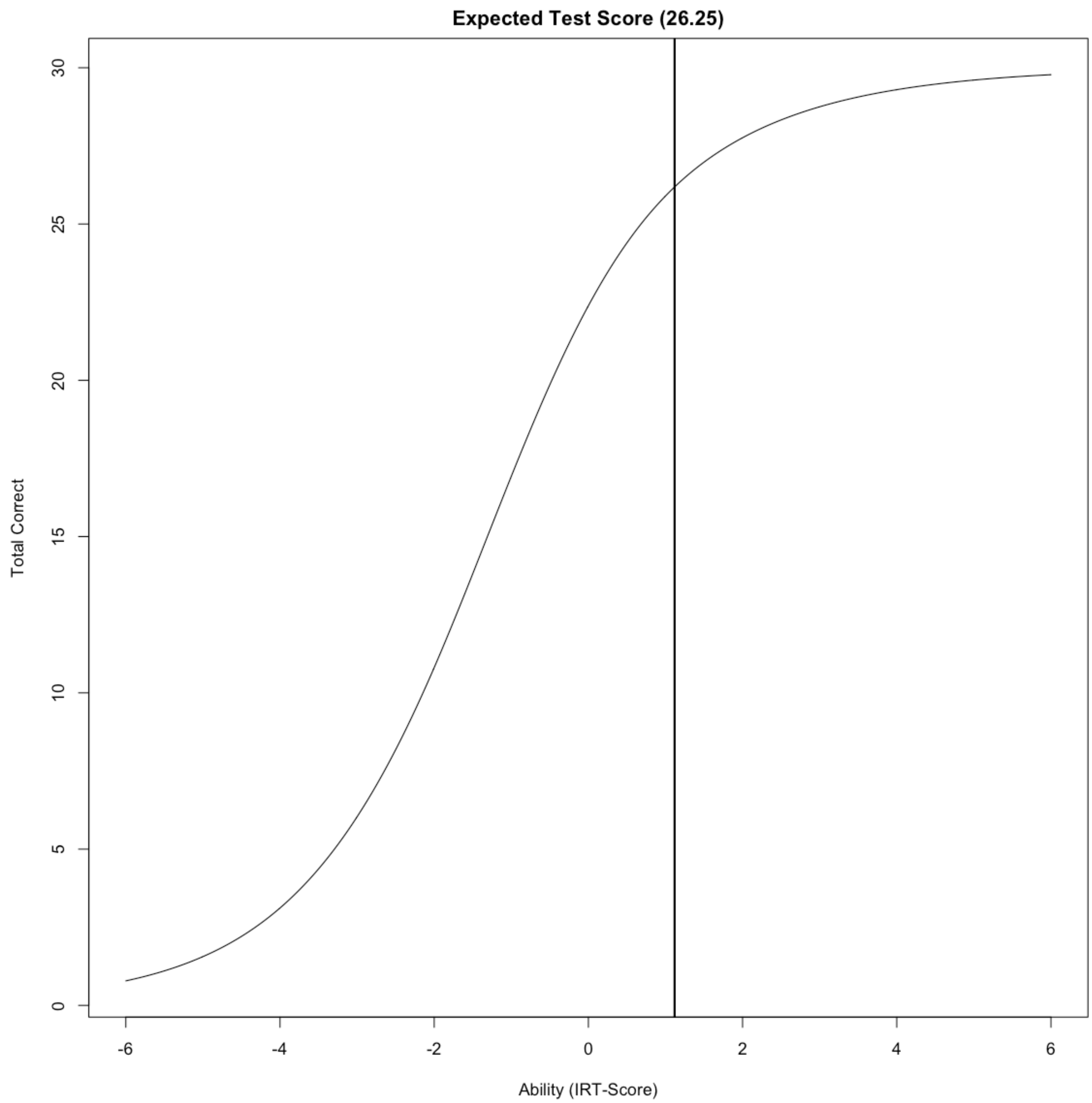

B: -0.28
Frequency
0
40
80
120
-0.5
-0.4
-0.3
-0.2
-0.1
PPS



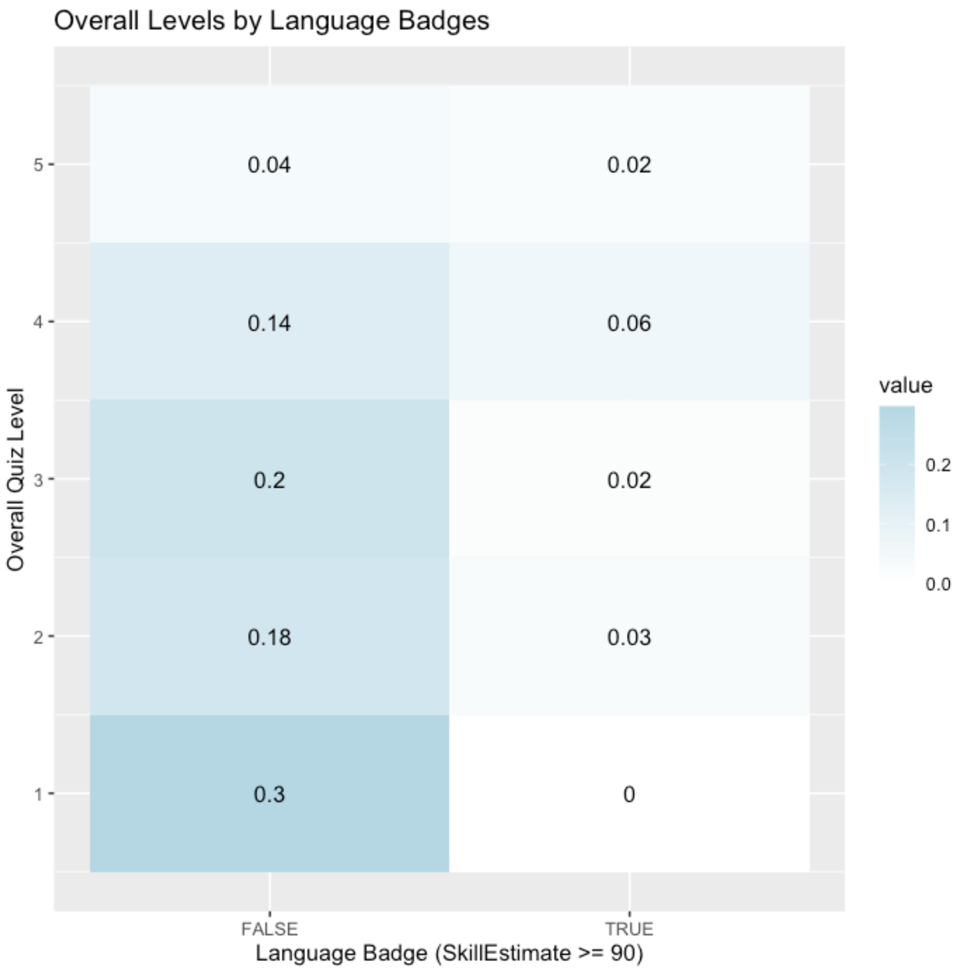


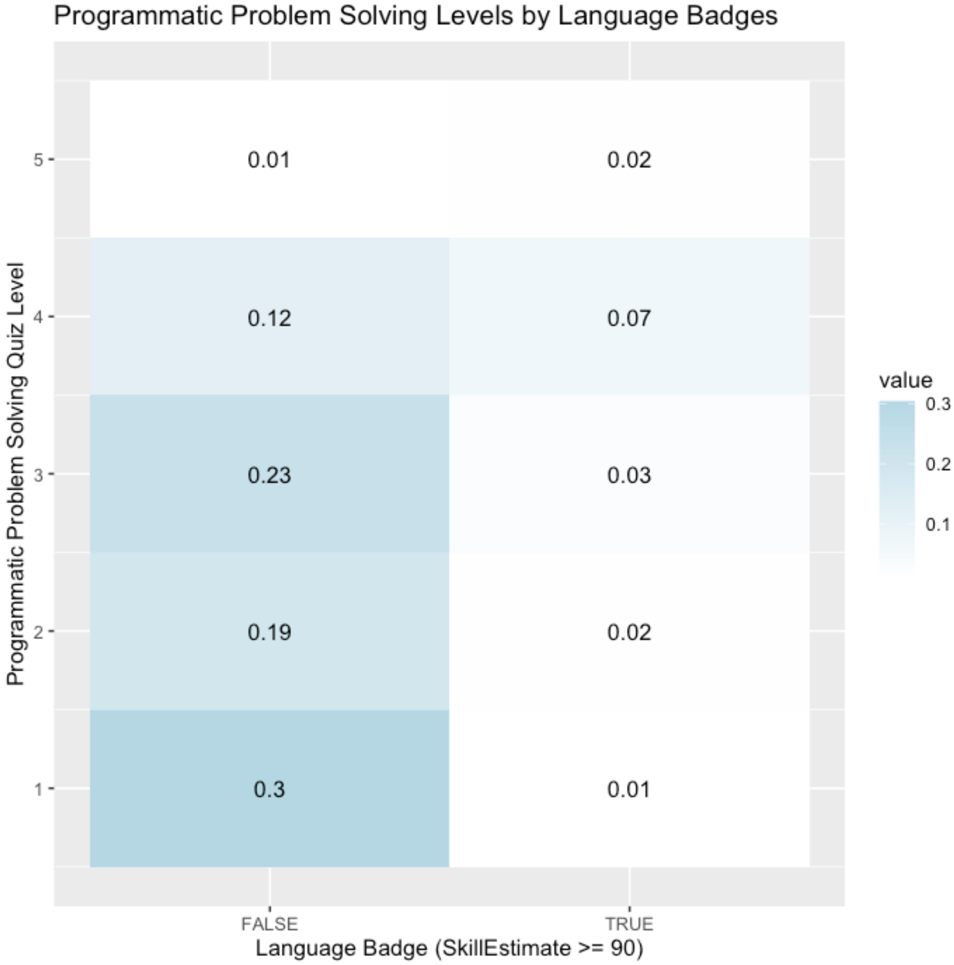
Result: Programmatic Problem Solving Scales score produce smaller linking error

Badges - Test Characteristic Curve

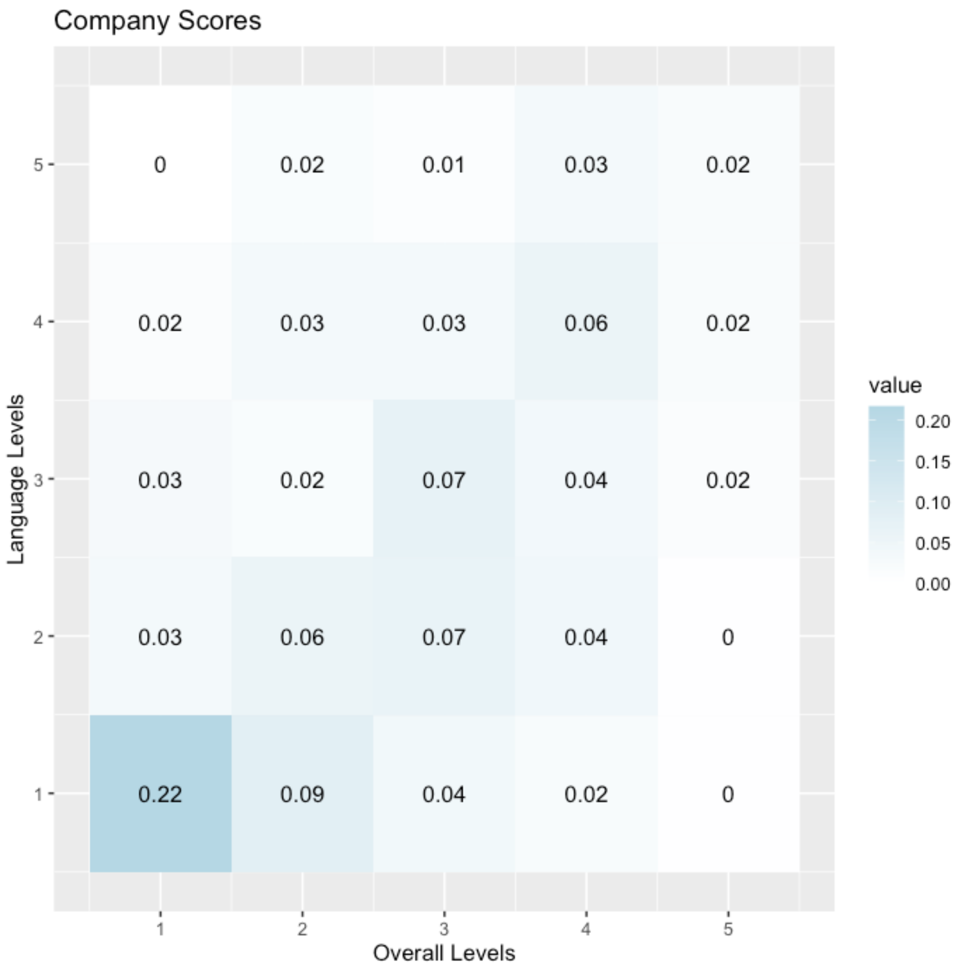


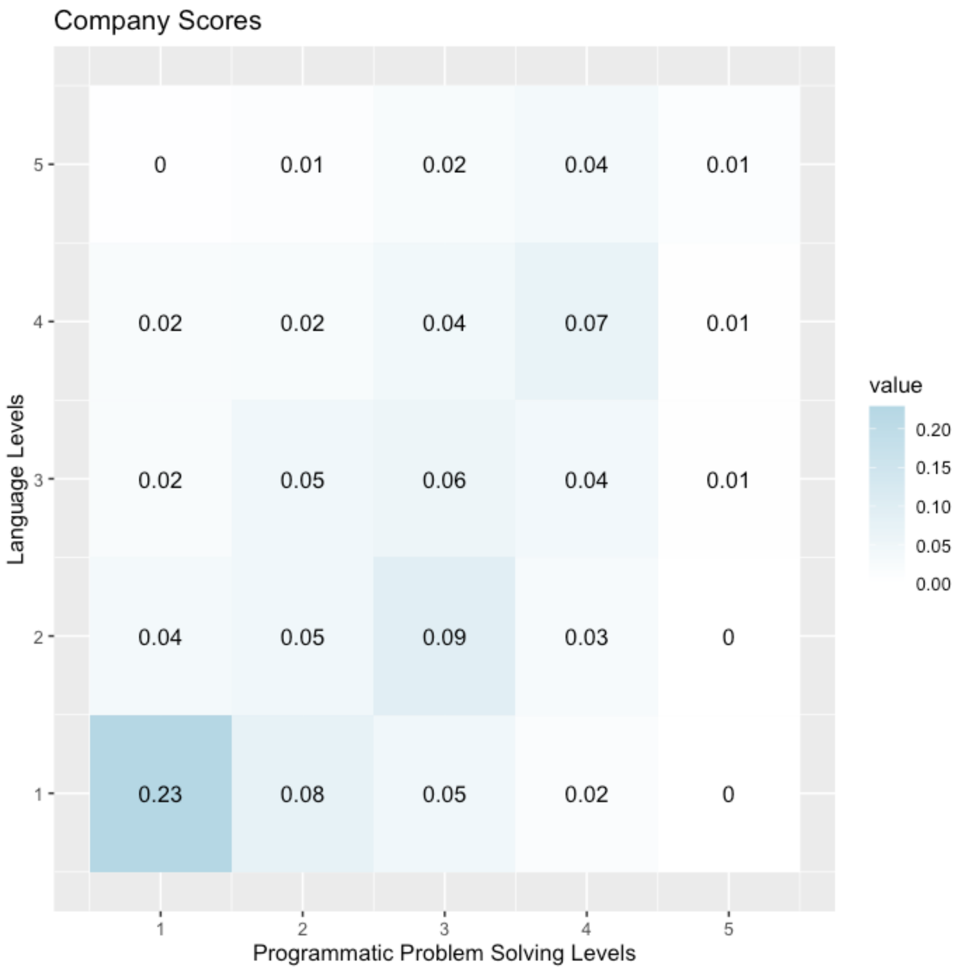
Badges - Quiz Level, by Badge


Overall Levels by Language Badges
0.04
0.02
0.14
0.06
0.2
0.02
0.18
0.03
0.3
0
5
4
3
2
1
Overall Quiz Level
FALSE
TRUE
Language Badge (SkillEstimate >= 90)
value
0.2
0.1
0.0



Language Levels, Quiz Levels


Company Scores
Language Levels
Overall Levels
value
0.20
0.15
0.10
0.05
0.00
5
4
3
2
1
1
2
3
4
5
0
0.02
0.01
0.03
0.02
0.02
0.03
0.03
0.06
0.02
0.03
0.02
0.07
0.04
0.02
0.03
0.06
0.07
0.04
0
0.22
0.09
0.04
0.02
0



Percentile Method