Online Supplemental Material

Table A1

Means of the Fit Indices when Fit with the 2PL in the Dichotomous Conditions

			Fitted Model										
			2PL										
I	N	Data Gen. Model	Singles	Doubles	Triples	Singles*	Doubles*	Triples*	Yen's Q1	PPP	SRMSR	AIC	BIC
10	250	2PL	0.04	0.15	0.27	0.00	0.04	0.07	8.15	0.55	0.07	1,484	1,574
		3PL	0.03	0.15	0.35	0.00	0.12	0.28	8.11	0.68	0.07	1,948	2,039
		GGUM	0.16	0.98	1.48	0.16	5.27	8.39	11.27	0.64	0.10	2,756	2,846
		M2PL	0.10	0.40	0.55	0.00	0.43	0.34	8.39	0.91	0.05	1,552	1,642
	500	2PL	0.05	0.15	0.27	0.00	0.02	0.05	11.11	0.43	0.06	2,943	3,048
		3PL	0.03	0.21	0.47	0.00	0.14	0.41	13.46	0.61	0.06	3,915	4,020
		GGUM	0.21	1.56	2.45	0.29	5.86	9.99	24.14	0.56	0.08	5,550	5,655
		M2PL	0.10	0.40	0.59	0.00	0.34	0.37	11.30	0.88	0.03	3,074	3,179
	1000	2PL	0.05	0.16	0.31	0.00	0.04	0.08	16.98	0.30	0.05	5,897	6,015
		3PL	0.06	0.27	0.70	0.00	0.16	0.71	19.96	0.51	0.05	7,834	7,952
		GGUM	0.32	2.73	4.23	0.42	6.54	10.71	41.08	0.48	0.08	11,128	11,246
		M2PL	0.08	0.38	0.64	0.00	0.30	0.47	18.79	0.84	0.03	6,086	6,204
	2000	2PL	0.05	0.15	0.32	0.00	0.05	0.14	28.66	0.18	0.04	11,781	11,913
		3PL	0.12	0.42	1.15	0.05	0.32	1.25	27.37	0.42	0.04	15,662	15,794
		GGUM	0.81	5.07	7.64	0.99	7.14	10.96	98.19	0.36	0.07	22,214	22,346
		M2PL	0.07	0.43	0.79	0.00	0.36	0.73	27.12	0.78	0.02	12,185	12,317
20	250	2PL	0.06	0.15	0.26	0.00	0.05	0.08	8.44	0.75	0.06	3,076	3,257
		3PL	0.06	0.21	0.37	0.00	0.18	0.30	8.11	0.77	0.06	3,905	4,086
		GGUM	0.51	1.32	1.91	1.32	8.41	13.14	11.09	0.62	0.10	5,733	5,914
		M2PL	0.22	0.56	0.72	0.03	1.03	0.99	8.80	0.94	0.05	3,366	3,546
	500	2PL	0.02	0.27	0.49	0.00	0.18	0.25	11.89	0.71	0.05	8,854	9,062
		3PL	0.02	0.29	0.53	0.00	0.25	0.37	13.55	0.76	0.05	9,816	10,025
		GGUM	0.74	2.74	3.38	2.07	12.40	15.48	27.54	0.54	0.09	11,456	11,664
		MGRM	0.40	0.72	0.80	0.35	1.11	0.98	11.67	0.62	0.05	5,438	5,647

	1000	2PL	0.00	0.21	0.27	0.03	0.28	0.50	16.15	0.62	0.03	19,407	19,824
		3PL	0.00	0.25	0.36	0.06	0.31	0.54	19.40	0.68	0.04	20,837	21,254
		GGUM	5.67	15.02	17.11	1.46	3.22	3.65	54.09	0.39	0.08	23,746	24,163
		M2PL	1.28	2.18	1.82	0.67	1.01	1.04	16.38	0.48	0.04	12,924	13,341
	2000	2PL	0.03	0.25	0.47	0.00	0.14	0.29	24.30	0.51	0.03	35,333	35,597
		3PL	0.02	0.31	0.62	0.00	0.21	0.50	31.53	0.60	0.03	39,234	39,498
		GGUM	5.11	11.37	12.67	7.30	16.58	18.51	106.44	0.32	0.08	45,502	45,766
		M2PL	0.42	0.67	0.78	0.27	0.57	0.68	24.96	0.33	0.03	21,598	21,862
40	250	2PL	0.09	0.18	0.32	0.01	0.07	0.11	8.63	0.91	0.05	7,134	7,495
		3PL	0.13	0.28	0.45	0.05	0.27	0.42	9.74	0.87	0.06	8,558	8,920
		GGUM	0.94	1.67	2.22	5.69	11.78	16.63	9.50	0.58	0.10	11,991	12,353
		M2PL	0.31	0.70	0.88	0.19	1.76	1.82	7.91	0.97	0.05	7,270	7,631
	500	2PL	0.03	0.28	0.50	0.00	0.21	0.27	8.92	0.88	0.04	19,407	19,824
		3PL	0.06	0.31	0.54	0.00	0.25	0.36	9.81	0.86	0.04	20,837	21,254
		GGUM	1.46	3.22	3.65	5.67	15.02	17.11	34.67	0.51	0.09	23,746	24,163
		M2PL	0.67	1.01	1.04	1.28	2.18	1.82	8.40	0.87	0.04	12,924	13,341
	1000	2PL	0.03	0.27	0.49	0.00	0.14	0.22	9.91	0.84	0.03	38,761	39,234
		3PL	0.03	0.30	0.56	0.00	0.19	0.34	11.89	0.82	0.03	41,637	42,110
		GGUM	2.37	5.51	6.20	5.72	14.68	16.62	61.29	0.45	0.08	47,259	47,731
		M2PL	0.72	1.02	1.04	0.87	1.35	1.24	9.49	0.81	0.03	25,819	26,292
	2000	2PL	0.03	0.27	0.48	0.00	0.15	0.31	12.12	0.79	0.02	77,611	78,139
		3PL	0.03	0.32	0.61	0.00	0.21	0.47	16.11	0.77	0.02	83,394	83,922
		GGUM	3.67	9.70	11.02	5.12	14.07	16.03	114.27	0.38	0.08	94,347	94,875
		M2PL	0.65	0.92	0.97	0.57	0.90	0.96	11.19	0.73	0.02	51,567	52,095

Note. I = number of items, N = sample size, Data Gen. Models = data generation models. Shaded cells indicate correctly specified models. Singles, doubles and triples are Drasgow et al.'s (1995) chi-square model-data fit statistics.

*Indicates the sample size adjusted model fit index (Chernyshenko, Stark, Drasgow, & Roberts, 2007).

Table A2

Means of the Fit Indices when Fit with the GGUM in the Dichotomous Conditions

			Fitted Model										
			GGUM										
I	N	Data Gen. Model	Singles	Doubles	Triples	Singles*	Doubles*	Triples*	Yen's Q1	PPP	SRSMR	AIC	BIC
10	250	GGUM	0.14	0.33	0.52	0.01	0.36	0.49	8.47	0.40	0.07	2,456	2,591
		2PL	10.39	6.60	4.25	117.90	71.64	42.87	40.59	0.22	0.13	2,735	2,871
		3PL	5.41	3.66	2.65	56.92	35.12	22.26	74.90	0.06	0.14	3,060	3,196
		M2PL	2.62	4.71	5.02	27.27	50.78	53.06	50.54	0.32	0.18	1,547	1,683
	500	GGUM	0.12	0.30	0.48	0.00	0.16	0.24	11.61	0.34	0.05	4,888	5,044
		2PL	12.90	8.53	5.68	73.09	46.67	29.43	76.90	0.04	0.14	5,668	5,825
		3PL	9.56	6.30	4.32	53.53	33.61	21.35	67.99	0.10	0.12	5,745	5,902
		M2PL	2.18	7.58	8.39	10.32	42.03	46.36	71.25	0.12	0.15	3,043	3,200
	1000	GGUM	0.09	0.26	0.45	0.00	0.11	0.17	16.76	0.28	0.04	9,805	9,982
		2PL	13.06	9.96	7.50	37.22	27.91	20.51	189.20	0.00	0.17	12,883	13,060
		3PL	12.04	7.90	5.50	34.33	21.79	14.53	157.60	0.05	0.11	11,593	11,770
		M2PL	3.64	12.82	13.67	9.51	36.81	39.18	98.81	0.03	0.15	6,214	6,391
	2000	GGUM	0.06	0.24	0.43	0.00	0.11	0.24	25.12	0.25	0.04	19,635	19,833
		2PL	11.28	12.80	12.06	16.43	18.69	17.59	352.40	0.00	0.17	25,991	26,189
		3PL	20.60	13.96	9.83	30.41	20.44	14.24	367.50	0.01	0.13	24,319	24,517
		M2PL	3.48	19.15	20.79	4.79	28.23	30.69	120.00	0.00	0.15	11,886	12,084
20	250	GGUM	0.04	0.26	0.48	0.00	0.28	0.35	8.84	0.62	0.06	5,568	5,839
		2PL	6.31	4.04	2.73	65.63	38.23	22.38	39.89	0.02	0.14	6,282	6,553
		3PL	1.64	1.30	1.22	11.48	6.90	5.24	43.81	0.01	0.12	6,159	6,430
		M2PL	2.64	2.52	2.21	23.73	21.01	17.38	44.32	0.14	0.12	3,465	3,737
	500	GGUM	0.06	0.29	0.50	0.00	0.21	0.28	11.94	0.67	0.04	10,100	10,413
		2PL	9.62	6.17	4.08	52.87	32.12	19.54	223.56	0.00	0.14	12,860	13,173
		3PL	1.99	1.54	1.43	7.73	4.75	3.92	125.96	0.00	0.10	12,091	12,404
		M2PL	4.46	7.45	7.62	22.70	39.91	40.82	134.43	0.02	0.15	7,640	7,953
	1000	GGUM	0.04	0.27	0.47	0.00	0.13	0.20	16.82	0.59	0.03	20,202	20,556
		2PL	10.08	6.78	4.71	28.26	18.34	12.14	396.91	0.00	0.13	25,303	25,657

		3PL	2.86	2.12	1.94	6.68	4.41	3.83	245.42	0.00	0.10	24,315	24,670
		M2PL	12.34	20.42	20.60	35.13	59.26	59.79	442.08	0.00	0.17	18,170	18,525
	2000	GGUM	0.03	0.25	0.46	0.00	0.14	0.28	26.67	0.49	0.02	40,455	40,851
		2PL	7.96	6.37	5.20	11.45	9.06	7.30	766.93	0.00	0.13	50,729	51,125
		3PL	3.60	2.71	2.61	4.91	3.56	3.41	521.05	0.00	0.11	49,614	50,010
		M2PL	37.01	55.91	56.97	55.01	83.37	84.95	1,376.28	0.00	0.21	43,787	44,184
40	250	GGUM	0.05	0.30	0.52	0.00	0.41	0.49	8.21	0.81	0.06	10,747	11,289
		2PL	7.34	4.68	3.15	77.35	45.25	26.84	36.80	0.00	0.11	13,732	14,275
		3PL	2.15	1.66	1.46	16.45	10.00	7.28	31.02	0.01	0.09	12,912	13,455
		M2PL	9.07	6.46	4.75	98.49	66.70	46.10	127.30	0.02	0.14	11,078	11,620
	500	GGUM	0.04	0.28	0.50	0.00	0.22	0.28	8.42	0.79	0.04	21,451	22,076
		2PL	6.63	4.35	3.05	34.84	21.09	13.31	200.90	0.00	0.10	26,944	27,569
		3PL	2.97	2.23	1.89	13.19	8.55	6.45	135.37	0.00	0.09	25,896	26,522
		M2PL	11.05	10.39	9.20	61.47	57.32	50.19	340.83	0.01	0.14	23,395	24,021
	1000	GGUM	0.03	0.27	0.49	0.00	0.14	0.22	9.57	0.75	0.03	42,929	43,638
		2PL	7.29	4.82	3.46	19.88	12.47	8.39	382.48	0.00	0.10	54,094	54,803
		3PL	4.37	3.33	2.79	11.18	8.01	6.38	271.68	0.00	0.08	52,388	53,097
		M2PL	17.86	19.12	17.89	51.61	55.36	51.67	752.52	0.01	0.15	48,974	49,683
	2000	GGUM	0.03	0.27	0.48	0.00	0.16	0.31	12.25	0.67	0.02	85,898	86,690
		2PL	12.16	9.12	6.94	17.74	13.18	9.91	848.79	0.00	0.10	110,861	111,654
		3PL	7.82	5.92	4.82	11.24	8.38	6.74	560.50	0.00	0.08	105,777	106,569
		M2PL	33.66	39.90	39.76	49.99	59.35	59.14	1,464.59	0.02	0.15	97,794	98,586

Note. *I* = number of items, *N* = sample size, Data Gen. Models = data generation models. Shaded cells indicate correctly specified models. Singles, doubles and triples are Drasgow et al.'s (1995) chi-square model-data fit statistics.
*Indicates the sample size adjusted model fit index (Chernyshenko, Stark, Drasgow, & Roberts, 2007).

Table A3

Means of the Fit Indices when Fit with the GRM in the Polytomous Conditions

			Fitted Model										
			Graded Response Model (GRM)										
I	N	Data Gen Model	Singles	Doubles	Triples	Singles*	Doubles*	Triples*	Yen's Q1	PPP	SRSMR	AIC	BIC
10	250	GRM	0.25	0.57	0.73	0.17	0.28	0.54	37.06	0.77	0.07	5176	5357
		GGUM	1.22	1.64	1.67	9.73	9.90	9.52	57.80	0.56	0.16	5151	5331
		MGRM	0.32	0.48	0.59	0.00	0.15	0.24	33.27	0.87	0.06	3739	3920
	500	GRM	0.14	0.56	0.75	0.00	0.17	0.32	36.00	0.75	0.05	10345	10554
		GGUM	2.08	2.59	2.64	10.21	10.77	10.84	68.46	0.53	0.15	10222	10431
		MGRM	0.21	0.50	0.65	0.00	0.15	0.22	36.31	0.83	0.05	7,457	7,666
	1000	GRM	0.09	0.56	0.77	0.00	0.18	0.41	40.65	0.64	0.04	20,663	20,899
		GGUM	2.23	4.58	4.53	5.64	11.76	11.60	87.09	0.46	0.15	20,453	20,689
		MGRM	0.16	0.51	0.69	0.00	0.15	0.29	40.43	0.71	0.04	14,941	15,177
	2000	GRM	0.07	0.56	0.81	0.00	0.36	0.71	43.43	0.49	0.04	41,298	41,563
		GGUM	3.91	8.24	7.52	5.49	11.86	10.78	124.80	0.39	0.15	40,755	41,019
		MGRM	0.15	0.53	0.73	0.01	0.33	0.60	50.75	0.52	0.03	29,852	30,116
20	250	GRM	0.34	0.64	0.78	0.19	0.42	0.71	35.14	0.89	0.06	10,711	11,073
		GGUM	1.21	1.81	2.08	10.46	11.99	14.43	50.55	0.52	0.20	12,298	12,660
		MGRM	0.65	0.68	0.74	0.61	0.51	0.62	33.76	0.96	0.05	8,209	8,571
	500	GRM	0.20	0.60	0.78	0.00	0.21	0.37	36.67	0.91	0.04	21,386	21,803
		GGUM	1.41	3.16	3.29	6.29	14.16	14.72	74.86	0.56	0.19	23,001	23,418
		MGRM	0.41	0.61	0.73	0.04	0.24	0.32	32.78	0.95	0.04	16,425	16,842
	1000	GRM	0.12	0.58	0.80	0.00	0.20	0.47	41.14	0.89	0.03	42,782	43,255
		GGUM	1.08	5.01	4.90	2.02	13.02	12.71	135.61	0.50	0.18	46,378	46,851
		MGRM	0.29	0.59	0.76	0.02	0.23	0.40	35.81	0.93	0.03	32,868	33,341
	2000	GRM	0.09	0.57	0.81	0.00	0.37	0.72	48.69	0.82	0.03	85,445	85,973
		GGUM	2.58	9.55	9.10	3.48	13.83	13.65	269.88	0.40	0.18	89,881	90,409
		MGRM	0.22	0.59	0.78	0.03	0.40	0.67	40.82	0.87	0.02	65,681	66,209
40	250	GRM	0.54	0.77	0.86	0.36	0.77	1.12	36.05	0.93	0.06	22,337	23,060
		GGUM	0.97	2.04	2.42	4.17	14.21	18.41	54.26	0.58	0.20	25,976	26,699
		MGRM	1.38	1.04	0.89	6.61	2.69	1.52	32.83	0.98	0.05	17,270	17,993

500	GRM	0.35	0.69	0.84	0.02	0.34	0.51	34.68	0.96	0.04	44,684	45,518
	GGUM	1.79	3.44	3.55	6.77	15.72	16.32	82.36	0.54	0.19	54,326	55,160
	MGRM	0.94	0.90	0.89	1.52	0.92	0.77	32.36	0.98	0.04	34,580	35,414
1000	GRM	0.22	0.64	0.83	0.00	0.27	0.54	35.73	0.96	0.03	89,344	90,289
	GGUM	3.56	6.27	5.44	8.93	16.83	14.32	170.52	0.62	0.18	91,281	92,226
	MGRM	0.65	0.78	0.86	0.42	0.52	0.63	33.51	0.98	0.03	69,116	70,061
2000	GRM	0.16	0.61	0.84	0.01	0.43	0.76	38.34	0.94	0.02	178,554	179,610
	GGUM	7.13	11.91	9.02	10.22	17.36	13.04	306.67	0.53	0.18	182,351	183,407
	MGRM	0.44	0.70	0.84	0.24	0.55	0.76	34.75	0.96	0.02	138,227	139,283

Note. I = number of items, N = sample size, Data Gen. Models = data generation models. Shaded cells indicate correctly specified models. Singles, doubles and triples are Drasgow et al.'s (1995) chi-square model-data fit statistics.
*Indicates the sample size adjusted model fit index (Chernyshenko, Stark, Drasgow, & Roberts, 2007).

Table A4

Means of the Fit Indices when Fit with the GGUM in the Polytomous Conditions

			Fitted Model										
			GGUM										
I	N	Data Gen. Model	Singles	Doubles	Triples	Singles*	Doubles*	Triples*	Yen's Q1	PPP	SRMSR	AIC	BIC
10	250	GGUM	0.09	0.44	0.59	0.00	0.19	0.29	35.41	0.90	0.06	4,326	4,552
		GRM	6.80	2.77	1.73	71.67	22.79	10.12	160.80	0.01	0.18	7,401	7,627
		MGRM	5.65	2.39	1.72	59.24	19.59	10.95	198.50	0.23	0.12	5,226	5,452
	500	GGUM	0.05	0.47	0.65	0.00	0.12	0.20	34.86	0.88	0.04	8,608	8,869
		GRM	11.48	4.52	2.48	64.10	22.22	9.89	245.60	0.00	0.17	14,570	14,830
		MGRM	7.60	3.22	2.15	41.15	14.65	8.10	277.30	0.15	0.09	9,800	10,060
	1000	GGUM	0.03	0.50	0.71	0.00	0.14	0.32	41.08	0.79	0.03	17,299	17,594
		GRM	15.28	5.88	3.10	43.89	15.66	7.31	447.00	0.00	0.15	28,257	28,553
		MGRM	11.15	4.65	2.91	31.47	11.96	6.71	354.50	0.04	0.08	19,530	19,825
	2000	GGUM	0.03	0.51	0.74	0.00	0.30	0.61	46.76	0.68	0.02	34,491	34,821
		GRM	24.63	8.96	4.27	36.44	12.94	5.90	864.90	0.00	0.15	56,116	56,446
		MGRM	24.69	9.27	4.97	36.53	13.40	6.95	835.00	0.01	0.07	39,610	39,940
20	250	GGUM	0.09	0.48	0.63	0.00	0.18	0.32	35.18	0.99	0.05	9,366	9,818
		GRM	6.92	3.00	1.96	72.77	25.33	12.77	111.30	0.03	0.15	14,569	15,021
		MGRM	19.51	7.31	4.42	223.40	76.97	42.20	585.90	0.05	0.19	15,235	15,687
	500	GGUM	0.08	0.51	0.70	0.00	0.14	0.25	36.18	0.99	0.04	18,678	19,200
		GRM	13.73	5.41	3.08	77.48	27.46	13.46	1,177.00	0.00	0.15	29,792	30,314
		MGRM	31.44	11.32	6.33	183.70	62.91	32.99	9,714.00	0.01	0.17	30,433	30,954
	1000	GGUM	0.06	0.54	0.75	0.00	0.17	0.38	39.08	0.98	0.03	37,339	37,930
		GRM	24.48	8.84	4.46	71.45	24.51	11.37	2,122.00	0.00	0.15	59,913	60,503
		MGRM	62.95	21.43	10.72	186.90	62.30	30.17	14,452.00	0.00	0.18	62,984	63,575
	2000	GGUM	0.06	0.54	0.78	0.00	0.34	0.67	45.31	0.95	0.02	74,699	75,360
		GRM	23.23	8.63	4.25	34.34	12.44	5.87	2,534.00	0.00	0.11	112,000	112,000
		MGRM	76.97	25.27	11.95	114.90	37.41	17.43	17,921.00	0.00	0.14	110,000	111,000
40	250	GGUM	0.17	0.54	0.67	0.01	0.24	0.41	34.38	0.99	0.05	19,879	20,783
		GRM	11.42	4.37	2.43	127.00	41.86	18.44	91.08	0.10	0.12	29,151	30,055
		MGRM	20.43	7.23	4.08	234.30	75.90	38.16	508.90	0.16	0.13	27,397	28,301

500	GGUM	0.15	0.57	0.74	0.00	0.18	0.30	32.75	1.00	0.04	39,743	40,786
	GRM	22.44	8.32	3.96	129.80	44.98	18.76	1,284.00	0.03	0.10	57,860	58,903
	MGRM	34.72	12.28	6.56	203.30	68.67	34.38	6,488.00	0.04	0.11	54,773	55,816
1000	GGUM	0.07	0.55	0.78	0.00	0.30	0.60	41.13	0.96	0.02	76,047	76,853
	GRM	33.21	12.07	5.48	97.67	34.22	14.43	1,568.00	0.02	0.08	110,000	111,000
	MGRM	54.26	18.75	9.35	160.80	54.23	26.05	11,992.00	0.00	0.10	107,000	108,000
2000	GGUM	0.10	0.57	0.81	0.01	0.37	0.71	34.76	0.99	0.02	159,000	160,000
	GRM	28.21	10.84	4.92	41.82	15.76	6.87	1423.00	0.01	0.05	202,000	204,000
	MGRM	88.79	28.91	13.38	132.70	42.87	19.57	13,287.00	0.00	0.09	207,000	208,000

Note. I = number of items, N = sample size, Data Gen. Models = data generation models. Shaded cells indicate correctly specified models. Singles, doubles and triples are Drasgow et al.'s (1995) chi-square model-data fit statistics.
*Indicates the sample size adjusted model fit index (Chernyshenko, Stark, Drasgow, & Roberts, 2007).

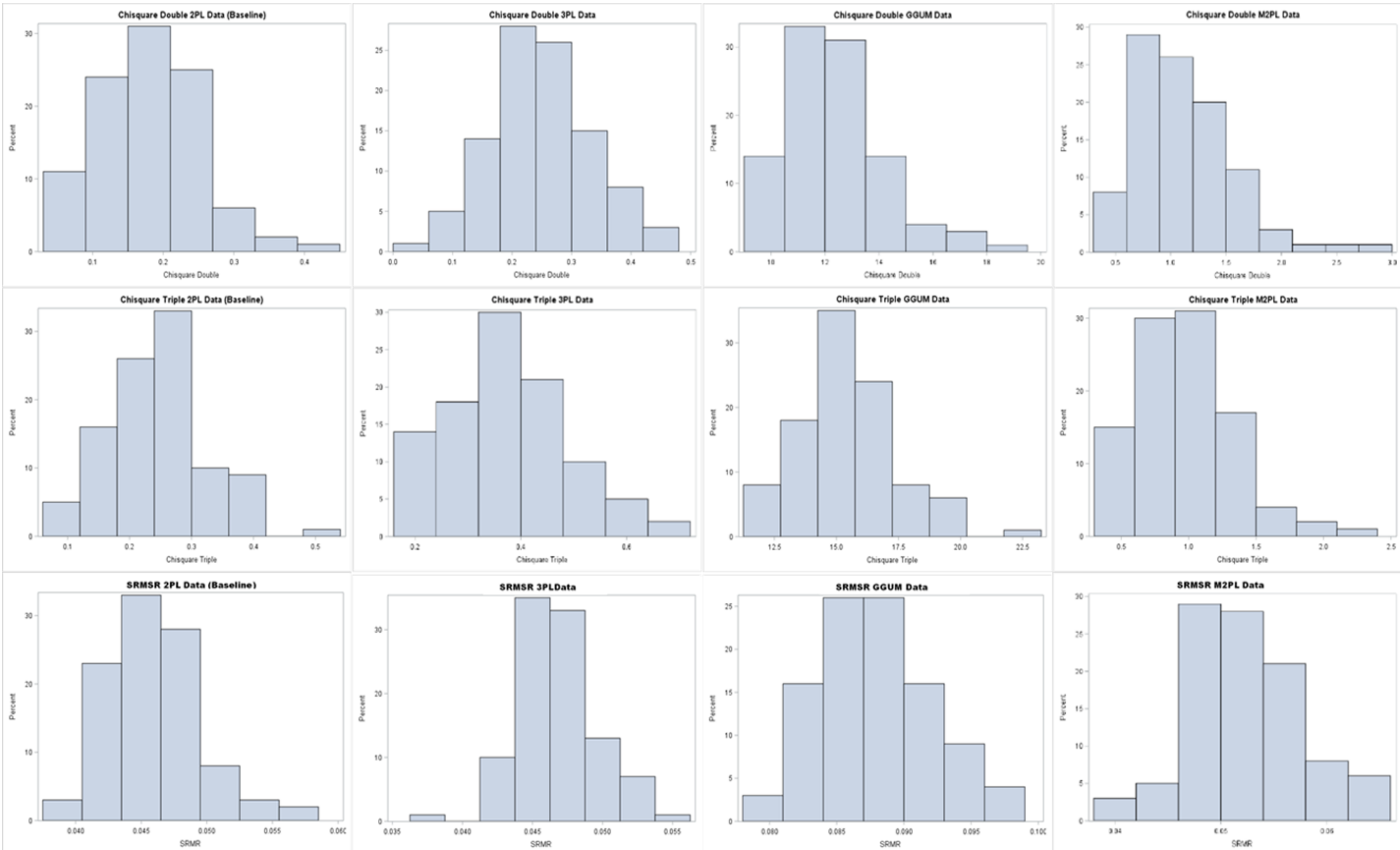

Figure A1. Distributions of model fit indices (from top row: Chi-square Doubles, Triples, and SRMSR) when fitting the 2PL model to data (from left column: generated from 2PL (Baseline), 3PL, GGUM, and M2PL) from conditions in which N=500, *i* = 20, and the data are dichotomous.

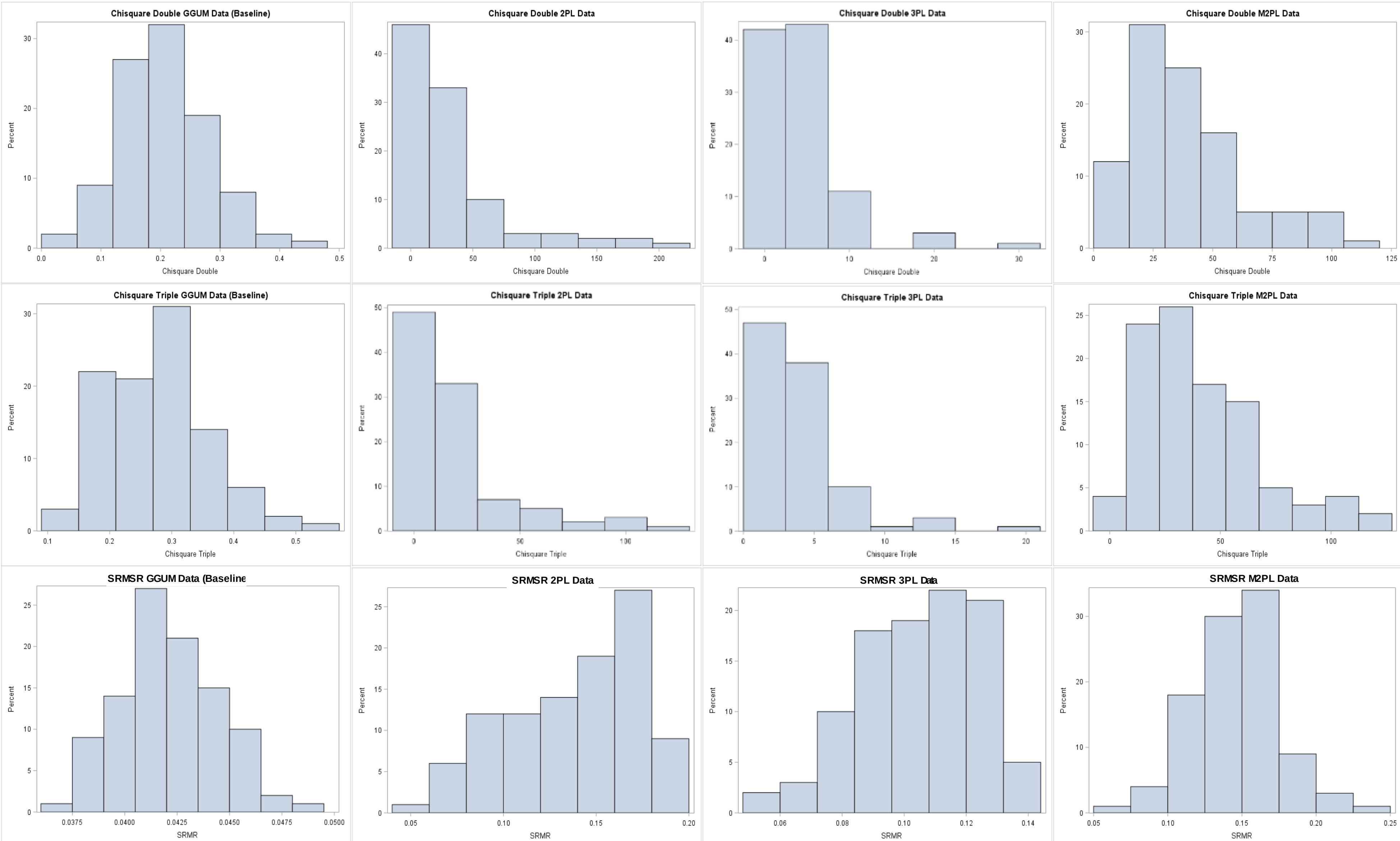

Figure A2. Distributions of model fit indices (from top row: Chi-square Doubles, Triples, and SRMSR) when fitting the GGUM to data (from left column: generated from GGUM (Baseline), 2PL, 3PL, and M2PL) from conditions in which N=500, *i* = 20, and the data are dichotomous.

Ox Code for the Study Simulations

```
#include <oxstd.h>
#include <oxprob.h>

decl N=500, J=20;                 // Set the numbers of examinees and items
decl data="Data_2PL.dat";         // Read response data
decl pars="Par_2PL.par";          // Read 2PL item parameters
decl thetas="Thetas_2PL.txt";     // Read 2PL person parameters
decl grids="grids.txt";           // Read the grid point file
decl X;
decl theta0, alpha0, beta0;

/// 2PL model function///
TwoPL(const a,b,th){
        decl sN=rows(th);
        return((1)./(1+exp(-1.7*a'.*(th-ones(sN,1)*b'))));
}

like(const a,b,th){
        decl tmpPR=TwoPL(a,b,th);
        decl tmpLike=X.*log(tmpPR)+(1-X).*log(1-tmpPR);
        decl Like = isdotnan(tmpLike) .? 0.0 .: tmpLike; // replace NaN with 0.0
        return(Like);
}

//============== Begin Main Program ==============;
main()
{
ranseed(today());

//Read in Grid file
decl grid;
decl gridfile=fopen(grids);
fscan(gridfile,"%#M",61,1,&grid);
fclose(gridfile);

decl theta;
decl thetafile=fopen(thetas);
fscan(thetafile,"%#M",N,1,&theta);
fclose(thetafile);

decl genpar;
decl pfile=fopen(pars);
fscan(pfile,"%#M",J,2,&genpar);
fclose(pfile);
```

```
alpha0=genpar[][0];
beta0=genpar[][1];

//Read in actual data
decl file=fopen(data);
fscan(file,"%#M",N,J,&X);
fclose(file);

//Output first and last lines of data;
println("Responses for first line of data");
print(X[0][]);
println("Responses for last line of data");
print(X[N-1][]);

/////////////          Model Fit Statistics          ////////////

// Loglikelihood, Deviance, AIC, BIC
decl LL = sumr(sumc(like(alpha0,beta0,theta)));
decl npar = 2*J;
decl AIC = -2*LL+npar;
decl BIC = -2*LL+npar*log(N);

print("\n","-------       Model Fit Statistics        ------");
print("%r",{" log-Likelihood    "," Deviance    "," AIC    "," BIC    "},"%10.4f", LL|-2*LL|AIC|BIC);
print("\n");

////////////////////     Yen's Q1     //////////////////////

decl tempmatrix=X~theta;
decl ordermatrix=sortbyc(tempmatrix,J);
decl orderX=dropc(ordermatrix,J);
decl orderth=ordermatrix[][J];
decl orderP=TwoPL(alpha0,beta0,orderth);
decl NN=N/10;
decl YensQ=zeros(1,J);
for(decl i=0;i<10;i++){
        decl bigO=(1/NN)*sumc(orderX[(NN*i):(NN*i+(NN-1))][]);
        decl bigE=(1/NN)*sumc(orderP[(NN*i):(NN*i+(NN-1))][]);
        YensQ=YensQ+NN*(((bigO-bigE).^2)./((bigE).*(1-bigE)));
}

print("\n","-------  Yen's Q1 Statistics (Yen, 1981)  ------","\n");
print("-------  Critical Value of Q1: ",quanchi(0.95,10-2),"  ------","\n");
print("\n","-------  By Item  ------","\n");
print("%c", {"Item","  Q1"}, "%cf",{"%8.4g","%8.4g"}, range(1, J, 1)'~YensQ',"\n");
print("-------  Average  ------","\n");
```

```
print(sumr(YensQ)/J,"\n");

/////////////   Calculate  Drasgow's  chi-squares     /////////////

decl lgrid=rows(grid);
decl pgrid=zeros(lgrid,1);
for(decl  j=0; j<lgrid;  j++){
	pgrid[j]=densn(grid[j]);
}

// Singlet
decl p1out=TwoPL(alpha0,beta0,grid);
decl p0out=1-p1out;
decl pout=p0out~p1out;
decl pp0out=N*sumc(p0out.*(pgrid*ones(1,J)))/10;
decl pp1out=N*sumc(p1out.*(pgrid*ones(1,J)))/10;
decl exp0=pp0out';
decl exp1=pp1out';
decl obs0=sumc((1-X))';
decl obs1=sumc(X)';
decl single_chisq0=((obs0-exp0).^2)./exp0;
decl single_chisq1=((obs1-exp1).^2)./exp1;
decl single_chisq=single_chisq0+single_chisq1;

// Doublet
decl dJ=J*(J-1)/2;
decl vec_double=zeros(dJ,2);
decl counter=0;
for (decl j=0; j<J-1; j++){
	for (decl jj=j+1; jj<J; jj++){
		vec_double[counter][0]=j;
		vec_double[counter][1]=jj;
		counter=counter+1;
	}
}
decl freqcell=zeros(dJ,4);
for (decl i=0; i<N; i++){
	for (decl j=0; j<dJ; j++){
		counter=0;
		for(decl  k=0;k<2; k++){
			for(decl  kk=0;kk<2;kk++){
				if(X[i][vec_double[j][0]]==k  &&
X[i][vec_double[j][1]]==kk){freqcell[j][counter]=freqcell[j][counter]+1;}
				else  {freqcell[j][counter]=freqcell[j][counter];}
				counter=counter+1;
			}
```

```
                }
        }
}
//print(freqcell);
decl pout_double=zeros(dJ,4);
decl temp1out=zeros(lgrid,dJ);
counter=0;
for (decl k=0; k<2; k++){
        for (decl kk=0;kk<2;kk++){
                for(decl j=0;j<dJ;j++){
                        temp1out[][j]=pout[][k*J+vec_double[j][0]].*pout[][kk*J+vec_double[j][1]];
                        pout_double[][counter]=(N*sumc(temp1out.*(pgrid*ones(1,dJ)))/10)';
                }
        counter=counter+1;
        }
}
decl expcell=pout_double;

// Combining cells that do not have more than 5
decl index=zeros(dJ,4);
decl eval=zeros(dJ,4);
decl fval=zeros(dJ,4);
for(decl j=0;j<dJ;j++){
        counter=0;
        for(decl jj=0;jj<4;jj++){
                if(expcell[j][jj]<5){
                        index[j][jj]=1;
                        eval[j][0]=eval[j][0]+expcell[j][jj];
                        fval[j][0]=fval[j][0]+freqcell[j][jj];
                        counter=counter+1;
                }
                else{
                        eval[j][jj]=expcell[j][jj];
                        fval[j][jj]=freqcell[j][jj];
                }
        }
}
decl newdf=3-sumr(index);

// Calculate Chisquares
decl double_chisq=zeros(dJ,4);
for(decl j=0;j<dJ;j++){
        for(decl jj=0;jj<4;jj++){
                if(index[j][jj]==0){
                        double_chisq[j][jj]=((fval[j][jj]-eval[j][jj]).^2)./eval[j][jj];
                }
```

```
        }
}
double_chisq=sumr(double_chisq);

// Triplet
counter=0;
decl tJ=J*(J-1)*(J-2)/3/2;
decl vec_triple=zeros(tJ,3);
for (decl j=0; j<J-2; j++){
        for (decl jj=j+1; jj<J-1; jj++){
                for (decl jjj=jj+1; jjj<J; jjj++){
                        vec_triple[counter][0]=j;
                        vec_triple[counter][1]=jj;
                        vec_triple[counter][2]=jjj;
                        counter=counter+1;
                }
        }
}
decl obscell=zeros(tJ,8);
for (decl i=0; i<N; i++){
        for (decl j=0; j<tJ; j++){
                counter=0;
                for(decl k=0;k<2; k++){
                        for(decl kk=0;kk<2;kk++){
                                for(decl kkk=0;kkk<2;kkk++){
                                        if(X[i][vec_triple[j][0]]==k  && X[i][vec_triple[j][1]]==kk  &&
X[i][vec_triple[j][2]]==kkk){obscell[j][counter]=obscell[j][counter]+1;}
                                        else {obscell[j][counter]=obscell[j][counter];}
                                        counter=counter+1;
                                }
                        }
                }
        }
}
//print(freqcell);
decl pout_triple=zeros(tJ,8);
decl temp2out=zeros(lgrid,tJ);
counter=0;
for (decl k=0; k<2; k++){
        for (decl kk=0;kk<2;kk++){
                for (decl kkk=0;kkk<2;kkk++){
                        for(decl j=0;j<tJ;j++){

        temp2out[][j]=pout[][k*J+vec_triple[j][0]].*pout[][kk*J+vec_triple[j][1]].*pout[][kkk*J+vec_tri
ple[j][2]];
                                pout_triple[][counter]=(N*sumc(temp2out.*(pgrid*ones(1,tJ)))/10)';
```

```
                        }
                        counter=counter+1;
                }
        }
}
decl Ecell=pout_triple;

// Combining cells that do not have more than 5
decl index2=zeros(tJ,8);
decl eval2=zeros(tJ,8);
decl fval2=zeros(tJ,8);
for(decl j=0;j<tJ;j++){
        counter=0;
        for(decl jj=0;jj<8;jj++){
                if(Ecell[j][jj]<5){
                        index2[j][jj]=1;
                        eval2[j][0]=eval2[j][0]+Ecell[j][jj];
                        fval2[j][0]=fval2[j][0]+obscell[j][jj];
                        counter=counter+1;
                }
                else{
                        eval2[j][jj]=Ecell[j][jj];
                        fval2[j][jj]=obscell[j][jj];
                }
        }
}
decl newdf2=7-sumr(index2);

// Calculate Chisquares
decl triple_chisq=zeros(tJ,8);
for(decl j=0;j<tJ;j++){
        for(decl jj=0;jj<8;jj++){
                if(index2[j][jj]==0){
                        triple_chisq[j][jj]=((fval2[j][jj]-eval2[j][jj]).^2)./eval2[j][jj];
                }
        }
}
triple_chisq=sumr(triple_chisq);

// Calculate Adjusted Chisquares
decl single_adjchisq=(1+3000.*(single_chisq-1)./N);
for(decl i=0;i<J;i++){
        if (single_adjchisq[i]<0) {single_adjchisq[i]=0;}
}
decl double_adjchisq=(newdf+3000.*(double_chisq-newdf)./N);
for(decl i=0;i<dJ;i++){
```

```
        if (double_adjchisq[i]<0) {double_adjchisq[i]=0;}
}
decl triple_adjchisq=(newdf2+3000.*(triple_chisq-newdf2)./N);
for(decl i=0;i<tJ;i++){
        if (triple_adjchisq[i]<0) {triple_adjchisq[i]=0;}
}
decl singlet=(single_chisq)~(single_chisq/1)~(single_adjchisq)~(single_adjchisq/1);
decl mean_singlet=sumc(singlet)/J;
decl doublet=(double_chisq)~(double_chisq./newdf)~(double_adjchisq)~(double_adjchisq./newdf);
decl mean_doublet=sumc(doublet)/dJ;
decl triplet=(triple_chisq)~(triple_chisq./newdf2)~(triple_adjchisq)~(triple_adjchisq./newdf2);
decl mean_triplet=sumc(triplet)/tJ;


print("\n","------  Drasgow Chi-squares by Items  ------ ","\n");
print("--- Singlets --- ");
print( "%c", {"Singlets","DF", "Chi", "Chi/df", "AdjChi", "AdjChi/df"},
"%cf",{"%8.4g","%8.4g","%8.4g", "%8.4g", "%8.4g", "%10.4g"}, range(1, J, 1)'~ones(J,1)~singlet);
print("\n");
print("--- Doublets --- ");
print( "%c", {"Doublets","DF", "Chi", "Chi/df", "AdjChi", "AdjChi/df"},
"%cf",{"%8.4g","%8.4g","%8.4g", "%8.4g", "%8.4g", "%10.4g"}, range(1, dJ, 1)'~newdf~doublet);
print("\n");
print("--- Triplets --- ");
print( "%c", {"Triplets","DF", "Chi", "Chi/df", "AdjChi", "AdjChi/df"},
"%cf",{"%8.4g","%8.4g","%8.4g", "%8.4g", "%8.4g", "%10.4g"}, range(1, tJ, 1)'~newdf2~triplet);

print("\n","--- Drasgow Chi-squares Overall --- ");
print("\n","-------  Goodfit < ",3,"  ------","\n");
print("\n","--- Chi-Square --- ");
print( "%c", {"Singlet","Doublet", "Triplet"}, "%cf",{"%8.4g","%8.4g",
"%8.4g"},mean_singlet[0]~mean_doublet[0]~mean_triplet[0]);
print("\n","--- Adjusted Chi-Square --- ","\n");
print( "%c", {"Singlet","Doublet", "Triplet"}, "%cf",{"%8.4g","%8.4g",
"%8.4g"},mean_singlet[2]~mean_doublet[2]~mean_triplet[2]);
print("\n","--- Chi-Square / df Ratio --- ","\n");
print( "%c", {"Singlet","Doublet", "Triplet"}, "%cf",{"%8.4g","%8.4g",
"%8.4g"},mean_singlet[1]~mean_doublet[1]~mean_triplet[1]);
print("\n","--- Adjusted Chi-Square / df Ratio --- ","\n");
print( "%c", {"Singlet","Doublet", "Triplet"}, "%cf",{"%8.4g","%8.4g",
"%8.4g"},mean_singlet[3]~mean_doublet[3]~mean_triplet[3]);
print("\n");

//////////// Standardized Root Mean Squared Residual (SRMSR) //////////
decl tmpPRvals=TwoPL(alpha0, beta0, theta);
decl tmpRand=ranu(N,J);
```

```
decl tmpXX = zeros(N,J);
for (decl p=0;p<N;p++){
                for (decl j=0;j<J;j++){
                        if(tmpPRvals[p][j] > tmpRand[p][j]){ tmpXX[p][j]=1;        }
                        else {tmpXX[p][j]=0;}
                }
        }
decl covGen=variance(tmpXX);
decl covObs=variance(X);
decl diff=covObs-covGen;
decl ss=sqrt(diagonal(covObs));
for (decl i=0;i<J;i++){
        for (decl j=0;j<J;j++){
                diff[i][j]=diff[i][j]/ss[i]/ss[j];
        }
}
diff=upper(diff.^2);
decl srmr = sqrt(sumc(sumr(diff))/(J*(J-1)));

print("\n","------- Standardized Root Mean Squared Residual (SRMSR) ------","\n");
print(srmr,"\n");

}
```

Annotated R Code for Estimating IRT Models in the mirt Package

```
######################################################
######################################################
#
#   Example code
#
######################################################
######################################################
```

Step 1. Install "mirt" from Github

https://github.com/philchalmers/mirt. This link describes the necessary steps. Installing #"mirt" from Github allows you to have the latest version.
It should be noted that only mirt 1.28.9 and above allow the calculation of SRMR for GGUM. You #may also need to install the package "psych" if you do not have it yet because the data we used #for examples came from this package.

Step 2. Load packages.

```
library(mirt)
library(psych)
```

Step 3. Define the measurement model.

This step is similar to what you need to do in lavaan or mplus for a measurement model. We will use a unidimensional model as an example. More complex examples can be found in the help document.

```
Model.Dich<-"F1=1-8"
```
The above model means that we have a factor F1 measured by 8 items. It will stay the same whether you fit dominance models or ideal point models.
```
Model.Poly<-"F1=1-10"
```
The above model means that we have a factor F1 measured by 10 items. It will stay the same whether you fit dominance models or ideal point models.

Step 4. Prepare data

We will use the dichotomous SAT data contained in mirt and the polytomous positive affect data from psych as examples. mirt requires all items have their own unique names

```
SAT12[SAT12 == 8] <- NA
```

```
SAT<- key2binary(SAT12,
key = c(1,4,5,2,3,1,2,1,3,1,2,4,2,1,5,3,4,4,1,4,3,3,4,1,3,5,1,3,1,5,4,3))[,c(1,6,8,10,19,14,27,29)]
colnames(SAT)<-paste0("Item",1:8)
SAT<-subset(SAT,rowSums(SAT)>=0)

PA<-msq[,c(1,3,11,21,27,28,39,41,49,62)]
PA<-subset(PA,rowSums(PA)>=0)


# Step 5. Fit models.  We will show how to fit 1/2/3/-PL models, graded response model (GRM), and
GGUM in mirt.

#   We will use Expectation-Maximization algorithm to estimate the model. The only thing that needs to
be changed for different models is "itemtype".
#Please refer to the help document for more detailed information regarding how to specify other types of
IRT models.
#   Note that ideal point models usually take longer than dominance models.



One_pl<-mirt(data=SAT,model = Model.Dich,itemtype = "Rasch",method = "EM")  # 1PL model

Two_pl<-mirt(data=SAT,model = Model.Dich,itemtype = "2PL",method = "EM")    # 2PL model

Three_pl<-mirt(data=SAT,model = Model.Dich,itemtype = "3PL",method = "EM")  # 3PL model

Graded<-mirt(data=PA,model = Model.Poly,itemtype = "graded",method = "EM")    # Graded response
model

GGUM<-mirt(data=PA,model = Model.Poly,itemtype = "ggum",method = "EM") # GGUM for
polytomous data


# Step 6. Calculate model fit indices.

#   The function M2 will be used to calculate indices including RMSEA and SRMR

Fit.1pl<-M2(One_pl)
Fit.2pl<-M2(Two_pl)
Fit.3pl<-M2(Three_pl)
Fit.GRM<-M2(Graded)
Fit.GGUM<-M2(GGUM)

print(Fit.1pl)
print(Fit.2pl)
print(Fit.3pl)
```

```
print(Fit.GRM)
print(Fit.GGUM)

AIC.1pl<-extract.mirt(One_pl,"AIC")
AIC.2pl<-extract.mirt(Two_pl,"AIC")
AIC.3pl<-extract.mirt(Three_pl,"AIC")
AIC.GRM<-extract.mirt(Graded,"AIC")
AIC.GGUM<-extract.mirt(GGUM,"AIC")

BIC.1pl<-extract.mirt(One_pl,"BIC")
BIC.2pl<-extract.mirt(Two_pl,"BIC")
BIC.3pl<-extract.mirt(Three_pl,"BIC")
BIC.GRM<-extract.mirt(Graded,"BIC")
BIC.GGUM<-extract.mirt(GGUM,"BIC")


# Step 7. Obtain item parameters

coef(One_pl,simplify=T)
coef(Two_pl,simplify=T)
coef(Three_pl,simplify=T)
coef(Graded,simplify=T)
coef(GGUM,simplify=T)


# Step 8. Plot

# 8.1 trace lines

plot(One_pl,type="trace")
plot(Two_pl,type="trace")
plot(Three_pl,type="trace")
plot(Graded,type="trace")
plot(GGUM,type="trace")


# 8.2 item response curve

plot(One_pl,type="itemscore")
plot(Two_pl,type="itemscore")
plot(Three_pl,type="itemscore")
plot(Graded,type="itemscore")
plot(GGUM,type="itemscore")


# 8.3 test information
```

```
plot(One_pl,type="info")
plot(Two_pl,type="info")
plot(Three_pl,type="info")
plot(Graded,type="info")
plot(GGUM,type="info")


# 9. Calculate thetas and empirical reliability (model-based reliability)

Score.1pl<-fscores(One_pl,method = "EAP",full.scores.SE = T)
Score.2pl<-fscores(Two_pl,method = "EAP",full.scores.SE = T)
Score.3pl<-fscores(Three_pl,method = "EAP",full.scores.SE = T)
Score.GRM<-fscores(Graded,method = "EAP",full.scores.SE = T)
Score.GGUM<-fscores(GGUM,method = "EAP",full.scores.SE = T)

empirical_rxx(Score.1pl)
empirical_rxx(Score.2pl)
empirical_rxx(Score.3pl)
empirical_rxx(Score.GRM)
empirical_rxx(Score.GGUM)
```