Supplemental Table 1

References for Studies and Data Sets Included in Meta-analysis Models

	Source ID	Sample ID
Atzaba-Poria, N., & Pike, A. (2008). Correlates of parental differential treatment: Parental and contextual factors during middle childhood. *Child Development, 79*(1), 217-232. doi:10.1111/j.1467-8624.2007.01121.x	1	1
Browne, D. T., Wade, M., Plamondon, A., Leckie, G., Perlman, M., Madigan, S., & Jenkins, J. M. (2018). Child and Contextual Effects in the Emergence of Differential Maternal Sensitivity Across Siblings. *Developmental Psychology 54(*7), 1265–1276. https://doi.org/10.1037/dev0000506	2	2
Buist, K. L. (2015). *Kindergarten children sample 1: Mixed-sex sibling pairs* [Unpublished raw data].	3	3
Buist, K. L. (2015*). Kindergarten children sample 1: Same-sex sibling pairs* [Unpublished raw data].	4	3
Buist, K. L. (2015). *Kindergarten children sample 2: Mixed-sex sibling pairs* [Unpublished raw data].	5	3
Buist, K. L. (2015). *Kindergarten children sample 2: Same-sex sibling pairs* [Unpublished raw data].	6	3
Campione-Barr, N. (2007–2008). *Parent and siblings relationship study: Same-sex pairs* [Unpublished raw data].	7	4
Campione-Barr, N. (2007–2008). *Parent and siblings relationship study: Mixed-sex pairs* [Unpublished raw data].	8	4
Day, R. D., Bean, R. A., Coyne, S. M., Dyer, W. M., Harper, J. M., Padilla-Walker, L. M. (2013). *Flourishing families: Mixed-sex pairs—Time 6* [Unpublished raw data].	9	5
Day, R. D., Bean, R. A., Coyne, S. M., Dyer, W. M., Harper, J. M., Padilla-Walker, L. M. (2013). *Flourishing families: Same-sex pairs—Time 6* [Unpublished raw data].	10	5
Day, R. D., Bean, R. A., Coyne, S. M., Dyer, W. M., Harper, J. M., Padilla-Walker, L. M. (2014). *Flourishing families: Mixed-sex pairs—Time 7* [Unpublished raw data].	11	5
Day, R. D., Bean, R. A., Coyne, S. M., Dyer, W. M., Harper, J. M., Padilla-Walker, L. M. (2014). *Flourishing families: Same-sex pairs—Time 7* [Unpublished raw data].	12	5
Day, R. D., Bean, R. A., Coyne, S. M., Dyer, W. M., Harper, J. M., Padilla-Walker, L. M. (2015). *Flourishing families: Mixed-sex pairs—Time 8* [Unpublished raw data].	13	5

Day, R. D., Bean, R. A., Coyne, S. M., Dyer, W. M., Harper, J. M., Padilla-Walker, L. M. (2015). *Flourishing families: Same-sex pairs—Time 8* [Unpublished raw data].	14	5
Jenkins, J. M., Rabash, J., & O'Connor, T.G. (2003). The role of the shared family context in differential parenting. *Developmental Psychology, 39*(1), 99–113. https://doi.org/10.1037/0012-1649.39.1.99	15	6
Jensen, A. C. (2016). *Young adults' mental health study: Mixed-sex sibling pairs* [Unpublished raw data].	16	7
Jensen, A. C. (2016). *Young adults' mental health study: Mixed-sex sibling pairs* [Unpublished raw data].	17	7
Jensen, A. C. (2017). *The sibling influence on becoming adults study: Mixed-sex sibling pairs* [Unpublished raw data].	18	8
Jensen, A. C. (2017). *The sibling influence on becoming adults study: Same-sex sibling pairs* [Unpublished raw data].	19	8
Jensen, A. C. (2016). *Baby boomer's sibling relationship: Mixed-sex sibling pairs* [Unpublished raw data].	20	9
Jensen, A. C. (2016). *Baby boomer's sibling relationships: Same-sex sibling pairs* [Unpublished raw data].	21	9
Jensen, A. C., Whiteman, S. D., Rand, J. S., & Fingerman, K. L. (2017). You're just like your dad: Intergenerational patterns of differential treatment of siblings. *The Journals of Gerontology: Series B: Psychological Sciences and Social Sciences, 72*(6), 1073–1083. https://doi.org/10.1093/geronb/gbw033	22	10
McHale, S. M., & Crouter, A. C. (1998). *Adolescent project: Mixed-sex pairs—Time 1* [Unpublished raw data].	23	11
McHale, S. M., & Crouter, A. C. (1998). *Adolescent project: Same-sex pairs—Time 1* [Unpublished raw data].	24	11
McHale, S. M., & Crouter, A. C. (1999). *Adolescent project: Mixed-sex pairs—Time 2* [Unpublished raw data].	25	11
McHale, S. M., & Crouter, A. C. (1999). *Adolescent project: Same-sex pairs—Time 2* [Unpublished raw data].	26	11
McHale, S. M., & Crouter, A. C. (2000). *Adolescent project: mixed-sex pairs—Time 3* [Unpublished raw data].	27	11
McHale, S. M., & Crouter, A. C. (2000). *Adolescent project: Same-sex pairs—Time* 3 [Unpublished raw data].	28	11
McHale, S. M., & Crouter, A. C. (2004). *African American study: Mixed-sex pairs—Time 1* [Unpublished raw data].	29	12
McHale, S. M., & Crouter, A. C. (2004). *African American study: Same-sex pairs—Time 1* [Unpublished raw data].	30	12

McHale, S. M., & Crouter, A. C. (2005). *African American study: Mixed-sex pairs—Time 2* [Unpublished raw data].	31	12
McHale, S. M., & Crouter, A. C. (2005). *African American study: Same-sex pairs—Time 2* [Unpublished raw data].	32	12
McHale, S. M., & Crouter, A. C. (2006). *African American study: Mixed-sex pairs—Time 3* [Unpublished raw data].	33	12
McHale, S. M., & Crouter, A. C. (2006). *African American study: Same-sex pairs—Time 3* [Unpublished raw data].	34	12
McHale, S. M., & Crouter, A. C. (1996). *Middle childhood project: Mixed-sex pairs—Time 1* [Unpublished raw data].	35	13
McHale, S. M., & Crouter, A. C. (1996). *Middle childhood project: same-sex pairs—Time 1* [Unpublished raw data].	36	13
McHale, S. M., & Crouter, A. C. (1997). *Middle childhood project: mixed-sex pairs—Time 2* [Unpublished raw data].	37	13
McHale, S. M., & Crouter, A. C. (1997). *Middle childhood project: Same-sex pairs—Time 2* [Unpublished raw data].	38	13
McHale, S. M., & Crouter, A. C. (1998). *Middle childhood project: Mixed-sex pairs—Time 3* [Unpublished raw data].	39	13
McHale, S. M., & Crouter, A. C. (1998). *Middle childhood project: Same-sex pairs—Time 3* [Unpublished raw data].	40	13
McHale, S. M., & Crouter, A. C. (2001). *Middle childhood project: Mixed-sex pairs—Time 4* [Unpublished raw data].	41	13
McHale, S. M., & Crouter, A. C. (2001). *Middle childhood project: Same-sex pairs—Time 4* [Unpublished raw data].	42	13
McHale, S. M., & Crouter, A. C. (2002). *Middle childhood project: Mixed-sex pairs—Time 5* [Unpublished raw data].	43	13
McHale, S. M., & Crouter, A. C. (2002). *Middle childhood project: Same-sex pairs—Time 5* [Unpublished raw data].	44	13
McHale, S. M., & Crouter, A. C. (2001). *Middle childhood project: Mixed-sex pairs—Time 6* [Unpublished raw data].	45	13
McHale, S. M., & Crouter, A. C. (2001). *Middle childhood project: Same-sex pairs—Time 6* [Unpublished raw data].	46	13
McHale, S. M., & Crouter, A. C. (2002). *Middle childhood project: Mixed-sex pairs—Time 7* [Unpublished raw data].	47	13
McHale, S. M., & Crouter, A. C. (2002). *Middle childhood project: Same-sex pairs—Time 7* [Unpublished raw data].	48	13
McHale, S. M., & Crouter, A. C. (2003). *Middle childhood project: Mixed-sex pairs—Time 8* [Unpublished raw data].	49	13

McHale, S. M., & Crouter, A. C. (2003). *Middle childhood project: Same-sex pairs—Time 8* [Unpublished raw data].	50	13
McHale, S. M., & Crouter, A. C. (2004). *Middle childhood project: Same-sex pairs—Time 9* [Unpublished raw data].	51	13
McHale, S. M., & Crouter, A. C. (2006). *Middle childhood project: Same-sex pairs—Time 11* [Unpublished raw data].	52	13
McHale, S. M., & Crouter, A. C. (2006). *Middle childhood project: Mixed-sex pairs—Time 11* [Unpublished raw data].	53	13
Meunier, J. C., Boyle, M., O'Connor, T. G., & Jenkins, J. M. (2013). Multilevel mediation: Cumulative contextual risk, maternal differential treatment, and children's behavior within families. *Child Development, 84*(5), 1594–1615. https://doi.org/10.1111/cdev.12066	54	2
Padilla, J., McHale, S. M., De Jesús, S. A. R., Updegraff, K. A., & Umaña, T. A. J. (2018). Longitudinal course and correlates of parents' differential treatment of siblings in Mexican-origin families. *Family Process, 57*(4), 979–995. https://doi.org/10.1111/famp.12328	55	14
Solmeyer, A. R., Killoren, S. E., McHale, S. M., & Updegraff, K. A. (2011). Coparenting around siblings' differential treatment in Mexican-origin families. *Journal of Family Psychology, 25*(2), 251. https://doi.org/10.1037/a0023201	56	14
Suitor, J. J., Sechrist, J., & Pillemer, K. (2007). Within-family differences in mothers' support to adult children in Black and White families. *Research on Aging*, *29*(5), 410–435. https://doi.org/10.1177/0164027507303636	57	15
Whiteman, S. D. (2009-2010). *The Purdue, parent, and adolescent sibling study: Mixed-sex sibling pairs* [Unpublished raw data].	58	16
Whiteman, S. D. (2009-2010). *The Purdue, parent, and adolescent sibling study: Same-sex sibling pairs* [Unpublished raw data.]	59	16

Note. These sources have linked parents' stress, parents' physical health, parents' mental health, parents' SES, parents' employment status, and parents' marital status with parental differential treatment.

Supplemental Table 2

Source and Effect-Size Characteristics for Models Examining Correlations Between Parents' Stress and PDT

		Sample/Source characteristics														Effect-size characteristics			
Source ID	Sample ID	Peer-reviewed	Sample size	# of effect sizes	U.S.	Avg. fam. size	Avg. age difference	Used beta	Avg. age of child	White (%)	Latinx (%)	Black (%)	Same-sex sibling pairs (%)	Male (%)	Younger target siblings (%)	Variation or perception	Recollected or concurrent	Types of PDT	Child or parent report	PDT from mom, dad, or both
1	1	Y	172	16	N	2.43	2.2	N	6.30	90.2	-	-	46.51	49.71	50	V	C	P, N	C, P	M, D
9	5	N	164–170	2	Y	2.65	3.08	N	16.32	64	0	12.8	100	48.00	48	P	C	O	C	M, D
10	5	N	196–201	2	Y	2.60	3.16	N	16.25	70.7	0.5	12.5	0	48.29	46.34	P	C	O	C	M, D
11	5	N	163–170	2	Y	2.74	3.03	N	17.35	64.8	0	12.8	100	48.30	48.86	P	C	O	C	M, D
12	5	N	191–202	2	Y	2.66	3.18	N	17.21	70.9	0.5	12.0	0	48.54	46.12	P	C	O	C	M, D
13	5	N	142–162	2	Y	2.62	2.98	N	18.42	64.7	0	12.5	100	49.71	48.55	P	C	O	C	M, D
14	5	N	178–196	2	Y	2.6	3.13	N	18.3	71.9	0	10.6	0	48.28	45.81	P	C	O	C	M, D
23	11	N	192–198	108	Y	2.60	2.41	N	13.77	100	0	0	100	54.55	50	V, P	C	P, N, R	C	M, D, B
24	11	N	192–196	108	Y	2.56	2.53	N	13.67	100	0	0	0	50.00	50	V, P	C	P, N, R	C	M, D, B
25	11	N	186–196	108	Y	2.60	2.41	N	14.79	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
26	11	N	186–192	108	Y	2.57	2.54	N	14.67	100	0	0	0	50.00	50	V, P	C	O, P, N, R	C	M, D, B
27	11	N	188–196	108	Y	2.61	2.41	N	15.8	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
28	11	N	184–190	108	Y	2.57	2.54	N	15.67	100	0	0	0	50.00	50	V, P	C	O, P, N, R	C	M, D, B
29	12	N	162–198	84	Y	2.74	3.80	N	12.24	0	0	100	100	50.00	50	V, P	C	P, N, R	C	M, D, B
30	12	N	158–206	84	Y	2.86	3.62	N	12.22	0	0	100	0	48.48	50	V, P	C	P, N, R	C	M, D, B
31	12	N	80–112	36	Y	2.74	3.59	N	13.96	0	0	100	100	50.00	50	V	C	P, N	C	M, D
32	12	N	190–278	36	Y	2.76	3.77	N	13.02	0	0	100	0	50.00	50	V	C	P, N	C	M, D
33	12	N	132–184	24	Y	2.62	3.68	N	14.37	0	0	100	100	51.05	50	V	C	P	C	M, D
34	12	N	132–196	24	Y	2.60	3.82	N	14.49	0	0	100	0	48.91	50	V	C	P	C	M, D
35	13	N	166–196	108	Y	2.59	2.54	N	9.61	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
36	13	N	178-210	108	Y	2.49	2.68	N	9.52	100	0	0	0	50.00	50	V, P	C	P, N, R, C	C, P	M, D, B
37	13	N	169–196	144	Y	2.58	2.54	N	10.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B

38	13	N	193–206	144	Y	2.51	2.67	N	10.48	100	0	0	0	50.00	50	V, P	C	P, N, R, C	C, P	M, D, B
39	13	N	181–196	144	Y	2.6	2.54	N	11.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
40	13	N	192–206	144	Y	2.52	2.67	N	11.48	100	0	0	0	50.00	50	V, P	C	P, N, R, C	C, P	M, D, B
41	13	N	156–160	16	Y	2.6	2.53	N	13.09	100	0	0	100	48.98	50	V	C	P, N	C, P	M, D
42	13	N	172–176	16	Y	2.52	2.61	N	13.03	100	0	0	0	50.00	50	V	C	P, N	C, P	M, D
43	13	N	150–156	16	Y	2.61	2.55	N	14.09	100	0	0	100	48.98	50	V	C	P, N	C, P	M, D
44	13	N	160–166	16	Y	2.53	2.66	N	13.98	100	0	0	0	50.00	50	V	C	P, N	C, P	M, D
45	13	N	164–190	132	Y	2.57	2.51	N	15.22	100	0	0	100	48.96	50	V, P	C	P, N, R, C	C, P	M, D, B
46	13	N	168–196	132	Y	2.52	2.66	N	15.13	100	0	0	0	50.00	50	V, P	C	P, N, R, C	C, P	M, D, B
47	13	N	158–186	108	Y	2.56	2.50	N	16.12	100	0	0	100	48.94	50	V, P	C	P, N, R, C	C, P	M, D, B
48	13	N	168–192	108	Y	2.56	2.65	N	15.99	100	0	0	0	50.00	50	V, P	C	P, N, R, C	C, P	M, D, B
49	13	N	158–182	108	Y	2.34	2.53	N	17.11	100	0	0	100	47.83	50	V, P	C	P, N, R, C	C	M, D, B
50	13	N	90–196	216	Y	2.48	2.67	N	17.04	100	0	0	0	50.00	50	V, P	C	P, N, R, C	C, P	M, D, B
51	13	N	88–112	108	Y	2.34	2.49	N	17.66	100	0	0	100	42.37	50	V, P	C	P, N, R, C	C, P	M, D, B
52	13	N	106–112	16	Y	1.97	2.50	N	25.12	100	0	0	100	43.06	50	V, P	C	P, R	C	M, D
53	13	N	114–130	16	Y	2	2.65	N	24.85	100	0	0	0	50.99	50	V, P	C	P, R	C	M, D
54	2	Y	920	4	N	2.52	2.27	N	3.51	58.5	-	6.0	42.32	51.63	-	V	C	P, N	P	M
55	14	Y	246	12	Y	3.39	2.96	N	18.1	0	100	0	54.47	49.59	50	V	C	P, N	C	M, D

Note. Y = yes. N = no. P = positivity. N = negativity. R = resource. C = control. O = overall. M = mother. D = father. B = mother and father. PDT = parental differential treatment.

Supplemental Table 3

Source and Effect-Size Characteristics for Models Examining Correlations Between Parents' Physical Health and PDT

		Sample/Source characteristics													Effect-size characteristics				
Source ID	Sample ID	Peer-reviewed	Sample size	# of effect sizes	U.S.	Avg. fam. size	Avg. age difference	Used beta	Avg. age of child	White (%)	Latinx (%)	Black (%)	Same-sex sibling pairs (%)	Male (%)	Younger target siblings (%)	Variation or perception	Recollected or concurrent	Types of PDT	Child or parent report	PDT from mom, dad, or both
23	11	N	198	36	Y	2.60	2.41	N	13.77	100	0	0	100	54.55	50	V, P	C	P, N, R	C	M, D, B
24	11	N	192–196	36	Y	2.56	2.53	N	13.67	100	0	0	0	50	50	V, P	C	P, N, R	C	M, D, B
25	11	N	192–196	36	Y	2.60	2.41	N	14.79	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
26	11	N	192	36	Y	2.57	2.54	N	14.67	100	0	0	0	50	50	V, P	C	O, P, N, R	C	M, D, B
27	11	N	194–196	36	Y	2.61	2.41	N	15.80	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
28	11	N	188–190	36	Y	2.57	2.54	N	15.67	100	0	0	0	50	50	V, P	C	P, N, R	C	M, D, B
37	13	N	195–196	48	Y	2.58	2.54	N	10.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
38	13	N	205–206	48	Y	2.51	2.67	N	10.48	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
39	13	N	195–196	48	Y	2.60	2.54	N	11.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
40	13	N	202–206	48	Y	2.52	2.67	N	11.48	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
45	13	N	168–188	44	Y	2.57	2.51	N	15.22	100	0	0	100	48.96	50	V, P	C	P, N, R, C	C, P	M, D, B
46	13	N	182–194	44	Y	2.52	2.66	N	15.13	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
57	15	Y	542	3	Y	4.40	-	Y	42.80	69.8	0	30.2	-	50.7	-	P	C	P, R	P	M
58	16	N	300–334	18	Y	2.82	2.70	N	15.77	74.2	3.1	20.8	100	42.86	50	V, P	C	O, P, N	C	M, D
59	16	N	279–316	18	Y	2.78	2.59	N	15.92	67.3	4.8	25.6	0	50	50	V, P	C	O, P, N	C	M, D

Note. Y = yes. N = no. P = positivity. N = negativity. R = resource. C = control. O = overall. M = mother. D = father. B = mother and father. PDT = parental differential treatment.

Supplemental Table 4

Source and Effect-Size Characteristics for Models Examining Correlations Between Parents' Mental Health and PDT

		Sample/Source characteristics													Effect-size characteristics				
Source ID	Sample ID	Peer-reviewed	Sample size	# of effect sizes	U.S.	Avg. fam. size	Avg. age difference	Used beta	Avg. age of child	White (%)	Latinx (%)	Black (%)	Same-sex sibling pairs (%)	Male (%)	Younger target siblings (%)	Variation or perception	Recollected or concurrent	Types of PDT	Child or parent report	PDT from mom, dad, or both
9	5	N	164–170	2	Y	2.65	3.08	N	16.32	64	0	12.8	100	48	48	P	C	O	C	M, D
10	5	N	196–201	2	Y	2.60	3.16	N	16.25	70.70	0.5	12.5	0	48.29	46.34	P	C	O	C	M, D
11	5	N	163–170	2	Y	2.74	3.03	N	17.35	64.80	0	12.8	100	48.30	48.86	P	C	O	C	M, D
12	5	N	191–202	2	Y	2.66	3.18	N	17.21	70.90	0.5	12.0	0	48.54	46.12	P	C	O	C	M, D
23	11	N	198	36	Y	2.60	2.41	N	13.77	100	0	0	100	54.55	50	V, P	C	P, N, R	C	M, D, B
24	11	N	194–196	36	Y	2.56	2.53	N	13.67	100	0	0	0	50	50	V, P	C	P, N, R	C	M, D, B
25	11	N	194–196	36	Y	2.60	2.41	N	14.79	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
26	11	N	192	36	Y	2.57	2.54	N	14.67	100	0	0	0	50	50	V, P	C	O, P, N, R	C	M, D, B
27	11	N	194–196	36	Y	2.61	2.41	N	15.8	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
28	11	N	190	36	Y	2.57	2.54	N	15.67	100	0	0	0	50	50	V, P	C	O, P, N, R	C	M, D, B
29	12	N	183–198	28	Y	2.74	3.80	N	12.24	0	0	100	100	50	50	V, P	C	P, N, R	C	M, D, B
30	12	N	189–206	28	Y	2.86	3.62	N	12.22	0	0	100	0	48.48	50	V, P	C	P, N, R	C	M, D, B
31	12	N	92–112	12	Y	2.74	3.59	N	13.96	0	0	100	100	50	50	V	C	P, N	C	M, D
32	12	N	228–278	12	Y	2.76	3.77	N	13.02	0	0	100	0	50	50	V	C	P, N	C	M, D
33	12	N	152–184	8	Y	2.62	3.68	N	14.37	0	0	100	100	51.05	50	V	C	P	C	M, D
34	12	N	168–196	8	Y	2.60	3.82	N	14.49	0	0	100	0	48.91	50	V	C	P	C	M, D
35	13	N	190–196	36	Y	2.59	2.54	N	9.61	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
36	13	N	190–210	36	Y	2.49	2.68	N	9.52	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
37	13	N	195–196	48	Y	2.58	2.54	N	10.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
38	13	N	205–206	48	Y	2.51	2.67	N	10.48	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
39	13	N	195–196	48	Y	2.60	2.54	N	11.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B

40	13	N	204–206	48	Y	2.52	2.67	N	11.48	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
45	13	N	170–190	44	Y	2.57	2.51	N	15.22	100	0	0	100	48.96	50	V, P	C	P, N, R, C	C, P	M, D, B
46	13	N	184–196	44	Y	2.52	2.66	N	15.13	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
47	13	N	170–184	36	Y	2.56	2.50	N	16.12	100	0	0	100	48.94	50	V, P	C	P, N, R, C	C, P	M, D, B
48	13	N	178–192	36	Y	2.56	2.65	N	15.99	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
49	13	N	164–182	36	Y	2.34	2.53	N	17.11	100	0	0	100	47.83	50	V, P	C	P, N, R, C	C	M, D, B
50	13	N	98–196	72	Y	2.48	2.67	N	17.04	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
51	13	N	96–110	36	Y	2.34	2.49	N	17.66	100	0	0	100	42.37	50	V, P	C	P, N, R, C	C, P	M, D, B
52	13	N	119–138	16	Y	1.97	2.50	N	25.12	100	0	0	100	43.06	50	V, P	C	P, R	C	M, D
53	13	N	122–143	16	Y	2.00	2.65	N	24.85	100	0	0	0	50.99	50	V, P	C	P, R	C	M, D
54	2	Y	920	4	N	2.52	2.27	N	3.51	58.5	-	6.0	42.32	51.63	-	V	C	P, N	P	M
55	14	Y	246	12	Y	3.39	2.96	Y	18.10	0	100	0	54.47	49.59	50	V	C	P, N	C	M, D

Note. Y = yes. N = no. P = positivity. N = negativity. R = resource. C = control. O = overall. M = mother. D = father. B = mother and father. PDT = parental differential treatment.

Supplemental Table 5

Source and Effect-Size Characteristics for Models Examining Correlations Between Parents' SES and PDT.

		Sample/Source characteristics													Effect-size characteristics				
Source ID	Sample ID	Peer-reviewed	Sample size	# of effect sizes	U.S.	Avg. fam. size	Avg. age difference	Used beta	Avg. age of child	White (%)	Latinx (%)	Black (%)	Same-sex sibling pairs (%)	Male (%)	Younger target siblings (%)	Variation or perception	Recollected or concurrent	Types of PDT	Child or parent report	PDT from mom, dad, or both
1	1	Y	172	8	N	2.43	2.20	N	6.30	90.2	-	-	46.51	49.71	50	V	C	P, N	C, P	M, D
2	2	Y	396	1	N	2.33	2.41	Y	1.37	56.5	-	12	-	51	50	V	C	P	P	M
3	3	N	303–308	6	N	2.59	2.71	N	5.28	90.1	-	-	100	50.65	53.25	P	C	P, C	P	M
4	3	N	322–327	6	N	2.60	2.70	N	5.19	90.1	-	-	0	48.32	49.22	P	C	P, C	P	M
5	3	N	113–115	6	N	2.64	3.81	N	5.4	90.1	-	-	100	45.22	50.43	P	C	P, C	P	D
6	3	N	107–108	6	N	2.61	2.75	N	5.3	90.1	-	-	0	52.78	50	P	C	P, C	P	D
7	4	N	48–58	24	Y	2.62	2.61	N	13.44	88.6	1.3	3.8	100	45.57	50	V	C	P, N, C	C	M, D
8	4	N	62–68	24	Y	2.53	2.97	N	13.76	83.3	1.5	9.1	0	50	50	V	C	P, N, C	C	M, D
9	5	N	169–175	2	Y	2.65	3.08	N	16.32	64	0	12.8	100	48	48	P	C	O	C	M, D
10	5	N	200–205	2	Y	2.60	3.16	N	16.25	70.7	0.5	12.5	0	48.29	46.34	P	C	O	C	M, D
11	5	N	169–176	2	Y	2.74	3.03	N	17.35	64.8	0	12.8	100	48.3	48.86	P	C	O	C	M, D
12	5	N	195–206	2	Y	2.66	3.18	N	17.21	70.9	0.5	12.0	0	48.54	46.12	P	C	O	C	M, D
13	5	N	152–73	2	Y	2.62	2.98	N	18.42	64.7	0	12.5	100	49.71	48.55	P	C	O	C	M, D
14	5	N	185–203	2	Y	2.60	3.13	N	18.30	71.9	0	10.6	0	48.28	45.81	P	C	O	C	M, D
15	6	Y	3762	1	N	2.20	-	N	7.46	-	-	-	-	-	-	P	C	N	P	-
16	7	N	138–154	16	Y	3.15	3.79	N	25.31	70.1	10.5	6.5	100	44.2	49.4	P	C	O, P, N, R	C	M, D
17	7	N	128–140	16`	Y	3.14	4.00	N	25.45	72.9	8.6	7.9	0	36.4	50.7	P	C	O, P, N, R	C	M, D
18	8	N	808–895	40	Y	3.21	4.11	N	25.36	74.6	5.7	8.8	100	52.07	44.8	P	C	O, P, N, R	C	M, D
19	8	N	781–855	40	Y	3.17	4.00	N	25.52	74.7	5.5	7.6	0	47.82	46.78	P	C	O, P, N R	C	M, D
22	10	Y	157	2	Y	3.34	-	N	23.42	66.0	-	-	33	53	-	V	C	P, R	P	B
23	11	N	192–198	54	Y	2.60	2.41	N	13.77	100	0	0	100	54.55	50	V, P	C	P, N, R	C	M, D, B
24	11	N	190–196	54	Y	2.56	2.53	N	13.67	100	0	0	0	50	50	V, P	C	P, N, R	C	M, D, B

25	11	N	192–196	54	Y	2.60	2.41	N	14.79	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
26	11	N	188–192	54	Y	2.57	2.54	N	14.67	100	0	0	0	50	50	V, P	C	O, P, N, R	C	M, D, B
27	11	N	190–194	54	Y	2.61	2.41	N	15.8	100	0	0	100	54.08	50	V, P	C	O, P, N, R	C	M, D, B
28	11	N	186–188	54	Y	2.57	2.54	N	15.67	100	0	0	0	50	50	V, P	C	O, P, N, R	C	M, D, B
29	12	N	180–198	42	Y	2.74	3.80	N	12.24	0	0	100	100	50	50	V, P	C	P, N, R	C	M, D, B
30	12	N	187–206	42	Y	2.86	3.62	N	12.22	0	0	100	0	48.48	50	V, P	C	P, N, R	C	M, D, B
31	12	N	92–112	18	Y	2.74	3.59	N	13.96	0	0	100	100	50	50	V	C	P, N	C	M, D
32	12	N	236–278	18	Y	2.76	3.77	N	13.02	0	0	100	0	50	50	V	C	P, N	C	M, D
33	12	N	150–184	12	Y	2.62	3.68	N	14.37	0	0	100	100	51.05	50	V	C	P	C	M, D
34	12	N	164–196	12	Y	2.60	3.82	N	14.49	0	0	100	0	48.91	50	V	C	P	C	M, D
35	13	N	188–196	54	Y	2.59	2.54	N	9.61	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
36	13	N	178–210	54	Y	2.49	2.68	N	9.52	100	0	0	0	50	50	V, P	C	P, N, R, C	C	M, D, B
37	13	N	190–196	72	Y	2.58	2.54	N	10.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
38	13	N	197–206	72	Y	2.51	2.67	N	10.48	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
39	13	N	183–196	72	Y	2.60	2.54	N	11.57	100	0	0	100	48.98	50	V, P	C	P, N, R, C	C, P	M, D, B
40	13	N	194–206	72	Y	2.52	2.67	N	11.48	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
45	13	N	174–190	66	Y	2.57	2.51	N	15.22	100	0	0	100	48.96	50	V, P	C	P, N, R, C	C, P	M, D, B
46	13	N	178–196	66	Y	2.52	2.66	N	15.13	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
47	13	N	170–186	54	Y	2.56	2.50	N	16.12	100	0	0	100	48.94	50	V, P	C	P, N, R, C	C, P	M, D, B
48	13	N	176–190	54	Y	2.56	2.65	N	15.99	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
49	13	N	162–180	54	Y	2.34	2.53	N	17.11	100	0	0	100	47.83	50	V, P	C	P, N, R, C	C	M, D, B
50	13	N	100–190	108	Y	2.48	2.67	N	17.04	100	0	0	0	50	50	V, P	C	P, N, R, C	C, P	M, D, B
51	13	N	96–113	54	Y	2.34	2.49	N	17.66	100	0	0	100	42.37	50	V, P	C	P, N, R, C	C, P	M, D, B
52	13	N	120–126	8	Y	1.97	2.50	N	25.12	100	0	0	100	43.06	50	V, P	C	P, R	C	M, D
53	13	N	122–133	8	Y	2.00	2.65	N	24.85	100	0	0	0	50.99	50	V, P	C	P, R	C	M, D
54	2	Y	920	4	N	2.52	2.27	N	3.51	58.5	-	6.0	42.32	51.63	-	V	C	P, N	P	M

55	14	Y	246	4	Y	3.39	2.96	Y	-	0	100	0	54.47	49.59	50	V	C	P, N	C	M, D
56	14	Y	243	8	Y	3.72	2.93	N	14.24	0	100	0	-	50	50	P	C	P, N	P	M, D
57	15	Y	542	3	Y	4.40	-	Y	42.80	69.8	0	30.2	-	50.70	-	P	C	P, R	P	M
58	16	N	285–332	12	Y	2.82	2.70	N	15.77	74.2	3.1	20.8	100	42.86	50	V, P	C	O, P, N	C	M, D
59	16	N	273–316	12	Y	2.78	2.59	N	15.92	67.3	4.8	25.6	0	50	50	V, P	C	O, P, N	C	M, D

Note. Y = yes. N = no. P = positivity. N = negativity. R = resource. C = control. O = overall. M = mother. D = father. B = mother and father. PDT = parental differential treatment.

Supplemental Table 6

Source and Effect-Size Characteristics for Models Examining Correlations Between Parents' Employment Status and PDT

		Sample/Source characteristics													Effect-size characteristics				
Source ID	Sample ID	Peer-reviewed	Sample size	# of effect sizes	U.S.	Avg. fam. size	Avg. age difference	Used beta	Avg. age of child	White (%)	Latinx (%)	Black (%)	Same-sex sibling pairs (%)	Male (%)	Younger target siblings (%)	Variation or perception	Recollected or concurrent	Types of PDT	Child or parent report	PDT from mom, dad, or both
3	3	N	303–308	4	N	2.59	2.71	N	5.28	90.1	-	-	100	50.65	53.25	P	C	P, C	P	M
4	3	N	322–327	4	N	2.60	2.70	N	5.19	90.1	-	-	0	48.32	49.22	P	C	P, C	P	M
5	3	N	113–115	4	N	2.64	3.81	N	5.40	90.1	-	-	100	45.22	50.43	P	C	P, C	P	D
6	3	N	107–108	4	N	2.61	2.75	N	5.30	90.1	-	-	0	52.78	50	P	C	P, C	P	D
57	15	Y	542	3	Y	4.40	-	Y	42.80	69.8	0	30.2	-	50.70	-	P	C	P, R	P	M
58	16	N	300–336	6	Y	2.82	2.70	N	15.77	74.2	3.1	20.8	100	42.86	50	V, P	C	O, P, N	C	M, D
59	16	N	316	3	Y	2.78	2.59	N	15.92	67.3	4.8	25.6	0	50	50	V, P	C	O, P, N	C	M, D

Note. Y = yes. N = no. P = positivity. N = negativity. R = resource. C = control. O = overall. M = mother. D = father. B = mother and father. PDT = parental differential treatment.

Supplemental Table 7

Source and Effect-Size Characteristics for Models Examining Correlations Between Parents' Marital Status and PDT

		Sample/Source characteristics													Effect-size characteristics				
Source ID	Sample ID	Peer-reviewed	Sample size	# of effect sizes	U.S.	Avg. fam. size	Avg. age difference	Used beta	Avg. age of child	White (%)	Latinx (%)	Black (%)	Same-sex sibling pairs (%)	Male (%)	Younger target siblings (%)	Variation or perception	Recollected or concurrent	Types of PDT	Child or parent report	PDT from mom, dad, or both
3	3	N	307–308	2	N	2.59	2.70	N	5.28	90.1	-	-	100	50.65	53.25	P	C	P, C	P	M
4	3	N	326–327	2	N	2.60	2.70	N	5.19	90.1	-	-	0	48.32	49.22	P	C	P, C	P	M
5	3	N	115	2	N	2.64	3.81	N	5.40	90.1	-	-	100	45.22	50.43	P	C	P, C	P	D
6	3	N	108	2	N	2.61	2.75	N	5.30	90.1	-	-	0	52.78	50	P	C	P, C	P	D
16	7	N	138–154	16	Y	3.15	3.79	N	25.31	70.1	10.5	6.5	100	44.20	49.40	P	C	O, P, N, R	C	M, D
17	7	N	128–140	16`	Y	3.14	4.00	N	25.45	72.9	8.6	7.9	0	36.40	50.70	P	C	O, P, N, R	C	M, D
18	8	N	808–895	20	Y	3.21	4.11	N	25.36	74.6	5.7	8.8	100	52.07	44.80	P	C	O, P, N, R	C	M, D
19	8	N	781–855	20	Y	3.17	4.00	N	25.52	74.7	5.5	7.6	0	47.82	46.78	P	C	O, P, N R	C	M, D
20	9	N	47–49	20	Y	3.77	4.71	N	59.32	86.0	2.3	7.3	100	33.90	47.90	P	R, C	O, N	C	M, D
21	9	N	54–55	20	Y	3.71	4.33	N	58.56	88.5	3.0	4.7	0	29.90	47.90	P	R, C	O, N	C	M, D
22	10	Y	157	2	Y	3.34	-	N	23.42	66.0	-	-	33	53	-	V	C	P, R	P	B
57	15	Y	542	3	Y	4.4	-	Y	42.8	69.8	0	30.2	-	50.7	-	P	C	P, R	P	M
58	16	N	300–336	6	Y	2.82	2.70	N	15.77	74.2	3.1	20.8	100	42.86	50	V, P	C	O, P, N	C	M, D
59	16	N	279–316	6	Y	2.78	2.59	N	15.92	67.3	4.8	25.6	0	50	50	V, P	C	O, P, N	C	M, D

Note. Y = yes. N = no. P = positivity. N = negativity. R = resource. C = control. O = overall. M = mother. D = father. B = mother and father. PDT = parental differential treatment.

Supplemental Table 8

Summary of Descriptive Statistics for Moderators for Models Examining Parents' Stress, Parents' Physical Health, and Parents' Mental Health as Correlated With PDT

	Stress (N = 2,918 participants)	Physical health (N = 1,990 participants)	Mental health (N = 2,746 participants)
	Proportion or *M* (*SD*)		
Level 4 (sample)	N = 7	N = 4	N = 6
United States	.71	1.00	.83
Level 3 (source)	N = 40	N = 15	N = 33
Family size	2.57 (.21)	2.72 (.47)	2.58 (.23)
Caucasian (%)	76.44 (37.84)	94.09 (12.31)	73.60 (40.68)
Male (%)	49.51 (2.19)	50.21 (2.78)	49.53 (2.40)
Younger than sibling (%)	49.15 (2.93)	50.00 (.00)	49.67 (.97)
Siblings same sex (%)	51.08 (48.71)	50.00 (51.89)	51.42 (49.22)
Age spacing	2.81 (.45)	2.54 (.10)	2.84 (.47)
Child age	14.45 (3.95)	15.82 (7.71)	14.57 (3.98)
Peer-reviewed article	.01	.01	.02
Level 2 (effect size)	N = 2,780	N = 535	N = 936
Parent reported on			
Mom PDT	.45	.46	.46
Dad PDT	.45	.45	.45
Both parents PDT	.09	.09	.09
PDT reporter			
Child(ren)	.81	.81	.82
Parent(s)	.19	.19	.18
Measurement type			
Difference score-based measure	.58	.57	.57
Perception-based measure	.42	.43	.43
Domain of PDT			
Overall PDT	.02	.05	.03
Positive interactions PDT	.45	.43	.45
Negative interactions PDT	.27	.29	.26
Resource PDT	.18	.15	.18
Control PDT	.09	.07	.09
Effect-size type			
Bivariate correlation	1.00	.99	1.00
Standardized semipartial correlation	.00	.01	.00

Sample size	201.78 (120.69)	236.00 (96.19)	210.58 (131.16)

Note. PDT = parental differential treatment.

Supplemental Table 9

Summary of Descriptive Statistics for Moderators for Models Examining Parents' SES, Parents' Employment Status, and Parents' Marital Status as Correlated With PDT

	SES (N = 11,166 participants)	Employment status (N = 2,015 participants)	Marital status (N = 4,706 participants)
	Proportion or *M* (*SD*)		
Level 4 (sample)	N = 16	N = 4	N = 8
United States	.69	.50	.75
Level 3 (source)	N = 53	N = 7	N = 14
Family size	2.69 (.39)	2.92 (.66)	3.14 (.53)
Caucasian (%)	73.37 (34.47)	81.67 (10.71)	78.89 (9.61)
Male (%)	48.65 (4.46)	48.65 (3.48)	45.56 (7.38)
Younger than sibling (%)	49.54 (1.88)	50.00 (1.82)	49.21 (2.44)
Siblings same sex (%)	50.54 (48.49)	50.00 (54.77)	48.77 (48.25)
Age spacing	2.91 (.53)	2.81 (.45)	3.59 (.74)
Child age	14.85 (7.05)	13.67 (13.78)	24.19 (18.26)
Peer-reviewed article	.02	.11	.04
Level 2 (effect size)	N = 1,587	N = 28	N = 137
Parent reported on			
Mom PDT	.46	.61	.50
Dad PDT	.45	.39	.48
Both parents PDT	.08	.00	.01
PDT reporter			
Child(ren)	.82	.32	.91
Parent(s)	.18	.68	.09
Measurement type			
Difference score-based measure	.53	.21	.07
Perception-based measure	.47	.79	.93
Domain of PDT			
Overall PDT	.04	.11	.26
Positive interactions PDT	.43	.43	.22
Negative interactions PDT	.26	.11	.32
Resource PDT	.18	.07	.17
Control PDT	.09	.29	.03
Effect-size type			
Bivariate correlation	.99	.89	.98

Standardized semipartial correlation	.01	.11	.02
Sample size	306.02 (515.89)	292.71 (148.75)	311.21 (274.94)

Note. PDT = parental differential treatment. SES = socioeconomic status.

Measurent Types Examples

The first type of approach is labeled perception-based measures (PBM). In PBM, researchers ask the individual participant to make a comparison of the PDT (e.g., Kowal & Kramer, 1997; Suitor et al., 2016). For example, the researcher may ask a child how much time their mother spends with them as compared to a sibling. The second type of approach is termed difference score-based methods (DSBM). In DSBM, researchers assess PDT by collecting separate reports of parenting toward each child and then calculate difference scores or a difference from the family-wide average of parenting (e.g., Meunier et al., 2013; Shanahan et al., 2008). For example, a father may report on how affectionate he is toward his firstborn, and then later in the study, he may report on how affectionate he is toward his second-born child. Researchers would then create a difference score to assess the level of PDT.

Analysis Code

NOTE: DataID is the variable for each unique sample ID. ArticleID is the variable for each unique source ID. ESNum is the variable for each unique effect size ID.

Functions used in the code

```
withold <- function (data, expr) {
  call <- match.call()
  analyses <- as.list(seq_len(data$m))
  for (i in seq_along(analyses)) {
    data.i <- complete(data, i)
    analyses[[i]] <- eval(expr = substitute(expr), envir = data.i, enclos = parent.frame())
    if (is.expression(analyses[[i]]))
      analyses[[i]] <- eval(expr = analyses[[i]], envir = data.i, enclos = parent.frame())
  }
  object <- list(call = call, call1 = data$call, nmis = data$nmis, analyses = analyses)
  oldClass(object) <- c("mira", "matrix")
  object
}

MetaModelsDescriptiveData <- function(x) {
  xData<-x %>% distinct(DataID, .keep_all = T) #keeping only one line per data set ID
  myvarsData <- c("ArticleID", "DataID", "AgeDif", "USA", "SampleSize") #creating vector with needed
variables
  xData <- xData[myvarsData] #keeping only needed variables
  xData$USA[xData$USA<0] <- NA
  xData$SampleSize[xData$DataID == 82] <- 761
  xData$SampleSize[xData$DataID == 83] <- 363
  xData$SampleSize[xData$DataID == 84] <- 335
  xData$SampleSize[xData$DataID == 85] <- 600
  xData$SampleSize[xData$DataID == 86] <- 223
  xData$SampleSize[xData$DataID == 87] <- 650
  xData$SampleSize[xData$DataID == 88] <- 491
  xData$SampleSize[xData$DataID == 89] <- 293
  xData$SampleSize[xData$DataID == 90] <- 1750
  xData$SampleSize[xData$DataID == 49] <- 392
  xData$SampleSize[xData$DataID == 65] <- 402
  xData$SampleSize[xData$DataID == 33] <- 380
  xData$SampleSize[xData$DataID == 48] <- 406
  xData$SampleSize[xData$DataID == 93] <- 274
  #retrieve stats for data set level variables
  TotalParticipants<-sum(xData$SampleSize)
  USAPec <- round(sum(xData$USA)/length(xData$USA),2)
  USATotal<- length(xData$USA[xData$USA==1])
  EuropeTotal<- length(xData$USA[xData$USA==0])
```

```
 #removing duplicates for Article level variables
 xArticle<-x %>% distinct(ArticleID, .keep_all = T) #keeping only one line per data set ID
 myvarsArticle <- c("ArticleID", "DataID", "SibAge",  "AgeDif", "FamSize", "WhitePerc", "GCSamePerc",
"MalePerc", "RelBirthYoungPerc", "SampleSize") #creating vector with needed variables
 xArticle <- xArticle[myvarsArticle] #keeping only needed variables
 xArticle$SibAge[xArticle$SibAge<0] <- NA
 xArticle$AgeDif[xArticle$AgeDif<0] <- NA
 xArticle$FamSize[xArticle$FamSize<0] <- NA
 xArticle$WhitePerc[xArticle$WhitePerc<0] <- NA
 xArticle$GCSamePerc[xArticle$GCSamePerc<0] <- NA
 xArticle$MalePerc[xArticle$MalePerc<0] <- NA
 xArticle$RelBirthYoungPerc[xArticle$RelBirthYoungPerc<0] <- NA
 xArticle$WhitePerc[xArticle$WhitePerc>10] <- NA
 xArticle$GCSamePerc[xArticle$GCSamePerc>10] <- NA
 xArticle$MalePerc[xArticle$MalePerc>10] <- NA
 xArticle$RelBirthYoungPerc[xArticle$RelBirthYoungPerc>10] <- NA
 xArticle$WhitePerc <- xArticle$WhitePerc*10
 xArticle$GCSamePerc <- xArticle$GCSamePerc*10
 xArticle$MalePerc <- xArticle$MalePerc*10
 xArticle$RelBirthYoungPerc <- xArticle$RelBirthYoungPerc*10
 #retrieve stats for article level variables
 ChildAgeMean<-round(mean(xArticle$SibAge, na.rm = T),2)
 ChildAgeSD<-round(sd(xArticle$SibAge, na.rm = T),2)
 AgeDifMean<-round(mean(xArticle$AgeDif, na.rm = T),2)
 AgeDifSD<-round(sd(xArticle$AgeDif, na.rm = T),2)
 FamSizeMean<-round(mean(xArticle$FamSize, na.rm = T),2)
 FamSizeSD<-round(sd(xArticle$FamSize, na.rm = T),2)
 WhitePercMean<-round(mean(xArticle$WhitePerc, na.rm = T),2)
 WhitePercSD<-round(sd(xArticle$WhitePerc, na.rm = T),2)
 GCSamePercMean<-round(mean(xArticle$GCSamePerc, na.rm = T),2)
 GCSamePercSD<-round(sd(xArticle$GCSamePerc, na.rm = T),2)
 MalePercMean<-round(mean(xArticle$MalePerc, na.rm = T),2)
 MalePercSD<-round(sd(xArticle$MalePerc, na.rm = T),2)
 RelBirthYoungPercMean<-round(mean(xArticle$RelBirthYoungPerc, na.rm = T),2)
 RelBirthYoungPercSD<-round(sd(xArticle$RelBirthYoungPerc, na.rm = T),2)
 SampleSizeMean<-round(mean(xArticle$SampleSize, na.rm = T),2)
 SampleSizeSD<-round(sd(xArticle$SampleSize, na.rm = T),2)
 #tidy up data for effect size level variables
 myvarsALL <- c("ESNum", "PeerRev", "UsedBeta", "OverallPDT", "PosPDT", "NegPDT", "ResourcePDT",
"ControlPDT", "ChildPDTReport", "MomPDT", "DadPDT", "BothParsPDT", "Variation", "Recollected")
#creating vector with needed variables
 xALL <- x[myvarsALL] #keeping only needed variables
 xALL$PeerRev[xALL$PeerRev<0] <- NA
 xALL$UsedBeta[xALL$UsedBeta<0] <- NA
 xALL$OverallPDT[xALL$OverallPDT<0] <- NA
 xALL$PosPDT[xALL$PosPDT<0] <- NA
 xALL$NegPDT[xALL$NegPDT<0] <- NA
 xALL$ResourcePDT[xALL$ResourcePDT<0] <- NA
```

```
xALL$ControlPDT[xALL$ControlPDT<0] <- NA
xALL$ChildPDTReport[xALL$ChildPDTReport<0] <- NA
xALL$MomPDT[xALL$MomPDT<0] <- NA
xALL$DadPDT[xALL$DadPDT<0] <- NA
xALL$BothParsPDT[xALL$BothParsPDT<0] <- NA
xALL$Variation[xALL$Variation<0] <- NA
xALL$Recollected[xALL$Recollected<0] <- NA
xALL$PeerRev[xALL$PeerRev>1] <- NA
xALL$UsedBeta[xALL$UsedBeta>1] <- NA
xALL$OverallPDT[xALL$OverallPDT>1] <- NA
xALL$PosPDT[xALL$PosPDT>1] <- NA
xALL$NegPDT[xALL$NegPDT>1] <- NA
xALL$ResourcePDT[xALL$ResourcePDT>1] <- NA
xALL$ControlPDT[xALL$ControlPDT>1] <- NA
xALL$ChildPDTReport[xALL$ChildPDTReport>1] <- NA
xALL$MomPDT[xALL$MomPDT>1] <- NA
xALL$DadPDT[xALL$DadPDT>1] <- NA
xALL$BothParsPDT[xALL$BothParsPDT>1] <- NA
xALL$Variation[xALL$Variation>1] <- NA
xALL$Recollected[xALL$Recollected>1] <- NA
#retrieve stats for effect size level variables
PeerReviewedPec <- round(sum(xALL$PeerRev)/length(xALL$PeerRev),2)
UsedBetaPec <- round(sum(xALL$UsedBeta)/length(xALL$UsedBeta),2)
OverallPDTPec <- round(sum(xALL$OverallPDT)/length(xALL$OverallPDT),2)
PosPDTPec <- round(sum(xALL$PosPDT)/length(xALL$PosPDT),2)
NegPDTPec <- round(sum(xALL$NegPDT)/length(xALL$NegPDT),2)
ResourcePDTPec <- round(sum(xALL$ResourcePDT)/length(xALL$ResourcePDT),2)
ControlPDTPec <- round(sum(xALL$ControlPDT)/length(xALL$ControlPDT),2)
ChildPDTReportPec <- round(sum(xALL$ChildPDTReport)/length(xALL$ChildPDTReport),2)
MomPDTPec <- round(sum(xALL$MomPDT)/length(xALL$MomPDT),2)
DadPDTPec <- round(sum(xALL$DadPDT)/length(xALL$DadPDT),2)
BothParsPDTPec <- round(sum(xALL$BothParsPDT)/length(xALL$BothParsPDT),2)
VariationPec <- round(sum(xALL$Variation)/length(xALL$Variation),2)
RecollectedPec <- round(sum(xALL$Recollected)/length(xALL$Recollected),2)
PeerReviewedTotal<- length(xALL$PeerRev[xALL$PeerRev==1])
NotPeerReviewedTotal<- length(xALL$PeerRev[xALL$PeerRev==0])
VariationTotal<- length(xALL$Variation[xALL$Variation==1])
PerceptionTotal<- length(xALL$Variation[xALL$Variation==0])
DadPDTTotal<- length(xALL$DadPDT[xALL$DadPDT==1])
MomPDTTotal<- length(xALL$MomPDT[xALL$MomPDT==1])
BothParsPDTTotal<- length(xALL$BothParsPDT[xALL$BothParsPDT==1])
RecollectedTotal<- length(xALL$Recollected[xALL$Recollected==1])
ConcurrentTotal<- length(xALL$Recollected[xALL$Recollected==0])
ChildPDTReportTotal<- length(xALL$ChildPDTReport[xALL$ChildPDTReport==1])
ParentPDTReportTotal<- length(xALL$ChildPDTReport[xALL$ChildPDTReport==0])
UsedBetaTotal<- length(xALL$UsedBeta[xALL$UsedBeta==1])
UsedCorrTotal<- length(xALL$UsedBeta[xALL$UsedBeta==0])
OverallPDTTotal<- length(xALL$OverallPDT[xALL$OverallPDT==1])
```

```
PosPDTTotal<- length(xALL$PosPDT[xALL$PosPDT==1])
NegPDTTotal<- length(xALL$NegPDT[xALL$NegPDT==1])
ResourcePDTTotal<- length(xALL$ResourcePDT[xALL$ResourcePDT==1])
ControlPDTTotal<- length(xALL$ControlPDT[xALL$ControlPDT==1])
DO.NOT.INCLUDE.THE.REST.IN.THE.TABLE <- " "
LEVEL.FOUR.VARIABLES <- " "
LEVEL.THREE.VARIABLES <- " "
LEVEL.TWO.VARIABLES <- " "
#create factors for statistics and names and bind together
Numbers<- c(LEVEL.FOUR.VARIABLES,
        USAPec,
        LEVEL.THREE.VARIABLES,
        AgeDifMean, AgeDifSD,
        FamSizeMean, FamSizeSD,
        WhitePercMean, WhitePercSD,
        GCSamePercMean, GCSamePercSD,
        MalePercMean, MalePercSD,
        RelBirthYoungPercMean, RelBirthYoungPercSD,
        SampleSizeMean, SampleSizeSD,
        ChildAgeMean, ChildAgeSD,
        PeerReviewedPec,
        LEVEL.TWO.VARIABLES,
        UsedBetaPec,
        OverallPDTPec, PosPDTPec, NegPDTPec, ResourcePDTPec, ControlPDTPec,
        ChildPDTReportPec,
        MomPDTPec, DadPDTPec, BothParsPDTPec,
        VariationPec,
        RecollectedPec,
        TotalParticipants,
        DO.NOT.INCLUDE.THE.REST.IN.THE.TABLE,
        USATotal, EuropeTotal,
        PeerReviewedTotal, NotPeerReviewedTotal,
        VariationTotal, PerceptionTotal,
        DadPDTTotal, MomPDTTotal, BothParsPDTTotal,
        RecollectedTotal, ConcurrentTotal,
        ChildPDTReportTotal, ParentPDTReportTotal,
        UsedBetaTotal, UsedCorrTotal,
        OverallPDTTotal, PosPDTTotal, NegPDTTotal, ResourcePDTTotal, ControlPDTTotal)
Variables <- c("Level Four Variables",
          "Proportion USA",
          "Level Three Variables",
          "Age Spacing Mean", "Age Spacing SD",
          "Family Size Mean", "Family Size SD",
          "% White Mean", "% White SD",
          "% Same Sex Mean", "% Same Sex SD",
          "% Male Mean", "% Male SD",
          "% Younger Mean", "% Younger SD",
          "Sample Size Mean", "Sample Size SD",
```

```
            "Child Age Mean", "Child Age SD",
            "Proportion Peer Reviewed",
            "Level Two Variables",
            "Proportion Used Beta",
            "Proportion Overall PDT", "Proportion Positive PDT", "Proportion Negative PDT", "Proportion
Resource PDT", "Proportion Control PDT",
            "Proportion Child PDT Report",
            "Proportion Mom PDT", "Proportion Dad PDT", "Proportion Both Parents PDT",
            "Proportion Variation",
            "Proportion Recollected",
            "Total # of Participants",
            "DO NOT INCLUDE THE REST IN THE TABLE",
            "Number USA", "Number Europe",
            "Number Peer Reviewed", "Number Not Peer Reviewed",
            "Number Variation", "Number Perception",
            "Nuumber Dad PDT", "Number Mom PDT", "Number Both Pars PDT",
            "RecollectedTotal", "ConcurrentTotal",
            "Number Child Report", "Number Parent Report",
            "Number Used Beta", "Number Used Correlation",
            "Number Overall PDT", "Number Positive PDT", "Number Negative PDT", "Number Resource
PDT", "Number Control PDT")
  ToDisplay <- cbind (Variables, Numbers)
  #export to a csv file with the name of the original data frame in the file name
  fname <- sprintf("Findings/Descriptive Stats/%s Descriptive Stats.csv",deparse(substitute(x)))
  write.csv(ToDisplay, file = fname)
  #return the values and end the function
  return(ToDisplay)}

MetaVarCompsModel1 <- function(objectM1){

  #prepare the object for extraction
  objectM1 <- getfit(objectM1)

  #Retrieve the Level2 Sigma2
  M1L2Sigma2 <- objectM1[["sigma2"]][1]

  #Retrieve the Level3 Sigma2
  M1L3Sigma2 <- objectM1[["sigma2"]][2]

  #Retrieve the Level3 Sigma2
  M1L4Sigma2 <- objectM1[["sigma2"]][3]

  #calculate total variance
  M1TotsVar <- sum(M1L2Sigma2,M1L3Sigma2,M1L4Sigma2)

  #ICC for level 2
  M1L2ICC <- M1L2Sigma2/M1TotsVar
```

```
  #ICC for level 3
  M1L3ICC <- M1L3Sigma2/M1TotsVar

  #ICC for level 3
  M1L4ICC <- M1L4Sigma2/M1TotsVar

  #setting which values to report
  ValuesM1 <- c(M1L2ICC,
                M1L3ICC,
                M1L4ICC
  )
  NamesM1 <- c("Level 2 ICC - Model 1",
               "Level 3 ICC - Model 1",
               "Level 4 ICC - Model 1"
  )
  ToDisplayM1 <- rbind(NamesM1, ValuesM1)

  #return the values and end the function
  return(ToDisplayM1)
}

MetaVarCompsModel2 <- function(object){

  #prepare the object for extraction
  object <- getfit(object)

  #Retrieve the Q
  QList <- rep(NA,50) #an empty list to put the info in
  for (i in 1:50){QList[i] <-object[[i]]["QE"]} # a loop to retrieve the info from each model
  QListCreated <- print(unlist(QList)) #prints the info for each model as a vector instead of a list
  Q <- round(mean(QListCreated),5) #creates the mean for the info

  #Retrieve the Q's p-value
  QPList <- rep(NA,50) #an empty list to put the info in
  for (i in 1:50){QPList[i] <-object[[i]]["QEp"]} # a loop to retrieve the info from each model
  QpListCreated <- print(unlist(QPList)) #prints the info for each model as a vector instead of a list
  Qp <- round(mean(QpListCreated),4) #creates the mean for the info

  #Retrieve the F for omnibus test of moderators
  FList <- rep(NA,50) #an empty list to put the info in
  for (i in 1:50){FList[i] <-object[[i]]["QM"]} # a loop to retrieve the info from each model
  FListCreated <- print(unlist(FList)) #prints the info for each model as a vector instead of a list
  FforMods <- round(mean(FListCreated),5) #creates the mean for the info

  #Retrieve the F's p-value
  FPList <- rep(NA,50) #an empty list to put the info in
  for (i in 1:50){FPList[i] <-object[[i]]["QMp"]} # a loop to retrieve the info from each model
  FpListCreated <- print(unlist(FPList)) #prints the info for each model as a vector instead of a list
```

```
FforModsp <- round(mean(FpListCreated),4) #creates the mean for the info

#Retrieve the Level2 Sigma2
L2Sig2List <- rep(NA,50) #an empty list to put the info in
for (i in 1:50){L2Sig2List[i] <-object[[i]][["sigma2"]][1]} # a loop to retrieve the info from each model
L2Sig2ListCreated <- print(unlist(L2Sig2List)) #prints the info for each model as a vector instead of a list
L2Sigma2 <- mean(L2Sig2ListCreated) #creates the mean for the info

#Retrieve the Level3 Sigma2
L3Sig2List <- rep(NA,50) #an empty list to put the info in
for (i in 1:50){L3Sig2List[i] <-object[[i]][["sigma2"]][2]} # a loop to retrieve the info from each model
L3Sig2ListCreated <- print(unlist(L3Sig2List)) #prints the info for each model as a vector instead of a list
L3Sigma2 <- mean(L3Sig2ListCreated) #creates the mean for the info

#Retrieve the Level3 Sigma2
L4Sig2List <- rep(NA,50) #an empty list to put the info in
for (i in 1:50){L4Sig2List[i] <-object[[i]][["sigma2"]][3]} # a loop to retrieve the info from each model
L4Sig2ListCreated <- print(unlist(L4Sig2List)) #prints the info for each model as a vector instead of a list
L4Sigma2 <- mean(L4Sig2ListCreated) #creates the mean for the info

#calculate total variance
TotalVar <- sum(L2Sigma2, L3Sigma2, L4Sigma2)

#ICC for level 2
L2ICC <- L2Sigma2/TotalVar

#ICC for level 3
L3ICC <- L3Sigma2/TotalVar

#ICC for level 3
L4ICC <- L4Sigma2/TotalVar

#setting which values to report
ValuesM2 <- c(Q,
        Qp,
        FforMods,
        FforModsp,
        L2Sigma2,
        L2ICC,
        L3Sigma2,
        L3ICC,
        L4Sigma2,
        L4ICC
)
NamesM2 <- c("Q - Model 2",
        "p-value for Q - Model 2",
        "F value for omnibus test of moderators - Model 2",
        "p-value for F value - Model 2",
```

```
            "Level 2 Sigma2 - Model 2",
            "Level 2 ICC - Model 2",
            "Level 3 Sigma2 - Model 2",
            "Level 3 ICC - Model 2",
            "Level 4 Sigma2 - Model 2",
            "Level 4 ICC - Model 2"
  )
  ToDisplayM2 <- rbind(NamesM2, ValuesM2)

  #return the values and end the function
  return(ToDisplayM2)
}

MetaVarCompsModel2LevelVarianceTest <- function(FullModel, RestrictedModelLevel2,
RestrictedModelLevel3, RestrictedModelLevel4){

  #create an empty list for each level
  LRTListL2 <- as.list(rep(NA,50)) #an empty list to put the info in
  LRTPValueListL2 <- as.list(rep(NA,50)) #an empty list to put the info in

  LRTListL3 <- as.list(rep(NA,50)) #an empty list to put the info in
  LRTPValueListL3 <- as.list(rep(NA,50)) #an empty list to put the info in

  LRTListL4 <- as.list(rep(NA,50)) #an empty list to put the info in
  LRTPValueListL4 <- as.list(rep(NA,50)) #an empty list to put the info in

  #run the loop for each level
  for (i in 1:50){
    LRTListL2[[i]] <- anova(FullModel[["analyses"]][[i]], RestrictedModelLevel2[["analyses"]][[i]])[["LRT"]]
#this runs each test and saves the LRT value in a list
    LRTPValueListL2[[i]] <- anova(FullModel[["analyses"]][[i]],
RestrictedModelLevel2[["analyses"]][[i]])[["pval"]] #this runs each test and saves the pvalue in a list

    LRTListL3[[i]] <- anova(FullModel[["analyses"]][[i]], RestrictedModelLevel3[["analyses"]][[i]])[["LRT"]]
#this runs each test and saves the LRT value in a list
    LRTPValueListL3[[i]] <- anova(FullModel[["analyses"]][[i]],
RestrictedModelLevel3[["analyses"]][[i]])[["pval"]] #this runs each test and saves the pvalue in a list

    LRTListL4[[i]] <- anova(FullModel[["analyses"]][[i]], RestrictedModelLevel4[["analyses"]][[i]])[["LRT"]]
#this runs each test and saves the LRT value in a list
    LRTPValueListL4[[i]] <- anova(FullModel[["analyses"]][[i]],
RestrictedModelLevel4[["analyses"]][[i]])[["pval"]] #this runs each test and saves the pvalue in a list
  }

  #take each list and turn it into a vector so we can pool the LRT and p-value info
  LRTListCreatedL2 <- print(unlist(LRTListL2)) #prints the info for each model as a vector instead of a list
  LRTModel2L2 <- round(mean(LRTListCreatedL2),5) #creates the mean for the info
```

```
  LRTPValueListCreatedL2 <- print(unlist(LRTPValueListL2)) #prints the info for each model as a vector
instead of a list
  LRTPValueModel2L2 <- round(mean(LRTPValueListCreatedL2),5) #creates the mean for the info

  LRTListCreatedL3 <- print(unlist(LRTListL3)) #prints the info for each model as a vector instead of a list
  LRTModel2L3 <- round(mean(LRTListCreatedL3),5) #creates the mean for the info
  LRTPValueListCreatedL3 <- print(unlist(LRTPValueListL3)) #prints the info for each model as a vector
instead of a list
  LRTPValueModel2L3 <- round(mean(LRTPValueListCreatedL3),5) #creates the mean for the info

  LRTListCreatedL4 <- print(unlist(LRTListL4)) #prints the info for each model as a vector instead of a list
  LRTModel2L4 <- round(mean(LRTListCreatedL4),5) #creates the mean for the info
  LRTPValueListCreatedL4 <- print(unlist(LRTPValueListL4)) #prints the info for each model as a vector
instead of a list
  LRTPValueModel2L4 <- round(mean(LRTPValueListCreatedL4),5) #creates the mean for the info

  #setting which values to report
  ValuesM2VarComps <- c(
    LRTModel2L2,
    LRTPValueModel2L2,
    LRTModel2L3,
    LRTPValueModel2L3,
    LRTModel2L4,
    LRTPValueModel2L4
  )
  NamesM2VarComps <- c(
    "LRT for Model 2 - Level 2",
    "LRT P Value for Model 2 - Level 2",
    "LRT for Model 2 - Level 3",
    "LRT P Value for Model 2 - Level 3",
    "LRT for Model 2 - Level 4",
    "LRT P Value for Model 2 - Level 4"
  )
  ToDisplayM2VarComps <- rbind(NamesM2VarComps, ValuesM2VarComps)

  #return the values and end the function
  return(ToDisplayM2VarComps)
}
```

```
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()       Four-level multilevel meta models          () () ()
# () () ()       testing links between Stress               () () ()
# () () ()       and absolute PDT                        () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()


#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
setwd("")

#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#for info on withold(), see https://www.metafor-
project.org/doku.php/tips:multiple_imputation_with_mice_and_metafor for more details
#I created the MetaModelsDescriptiveData(), MetaVarCompsModel1(), MetaVarCompsModel2(), and
MetaVarCompsModel2LevelVarianceTest()
#functions - more notes are available in the original file "Meta Analysis Functions.R"
library(metafor)
library(PcAux)
library(mice)
library(psych)
library(tidyverse)
library(car)


#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
WholeMeta <- read.csv(file = "Programs and Data/FinalCleanPDTMeta.csv", header = T)
```

```
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Stress
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Stress
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Stress
Meta1 <- subset(x = WholeMeta, subset = MagnitudeDirectional == 0 & Outcome == 63)



#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
Meta2 <- Meta1[,

c("ESNum","SampleSize","Correlation","ArticleID","PeerRev","USA","DataID","FamSize","AgeDif",

"UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","ControlPDT",

"ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","Perception",
        "Recollected","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")]

MetaModelsDescriptiveData(Meta2) # this saves the descriptive stats for the non-imputed model
#we take descriptive stats from this data frame so that it includes missing data and the non-centered
versions of continous variables



#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#we will drop the non-centered versions of continuous variables, because we want the imputation to be
based on the centered versions
Meta3 <- Meta2
attach(Meta3)
Meta3$FamSizeC <- FamSize-mean(FamSize, na.rm = T)
Meta3$AgeDifC <- AgeDif-mean(AgeDif, na.rm = T)
Meta3$AgeC <- Age-mean(Age, na.rm = T)
Meta3$WhitePercC <- WhitePerc-mean(WhitePerc, na.rm = T)
Meta3$GCSamePercC <- GCSamePerc-mean(GCSamePerc, na.rm = T)
Meta3$MalePercC <- MalePerc-mean(MalePerc, na.rm = T)
Meta3$RelBirthYoungPercC <- RelBirthYoungPerc-mean(RelBirthYoungPerc, na.rm = T)
detach(Meta3)
DropVars <- names(Meta3) %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
Meta3 <- Meta3[!DropVars]
DropVars <- Meta3 %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
```

```
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#Do this before the PCAs, it doesn't like doing it on an imputed data frame, so do it first
Meta4 <- escalc(measure = "ZCOR", ri =Correlation,
            ni = SampleSize, data = Meta3, vtype = "LS", append = TRUE)




#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#re-do this step every time

MetaNOMISS <- Meta4[, which(colMeans(is.na(Meta4)) ==.00)]#creating data frame with only variables
with no missing values

names(MetaNOMISS)#I run this to get the names and then sort those into the type of variable below

## Prepare the data by first sorting out what types of variables I have - continuous variables are left off
the code but listed below
myNoms   <- c("PeerRev",       "USA","UsedBeta",      "OverallPDT",    "PosPDT",        "NegPDT",
"EvalPDT",       "ResourcePDT",    "FairPDT", "ControlPDT",     "ChildPDTReport",
"ParentPDTReport","MomPDT",        "DadPDT",        "BothParsPDT",    "Variation", "Perception",
"Recollected")
myIds    <- c("ESNum", "ArticleID", "DataID")
#myOrds   <- c() #none this time
#Continuous - "SampleSize",     "Correlation","FamSizeC",      "AgeDifC",       "AgeC",          "WhitePercC",
"GCSamePercC", "MalePercC",      "yi",            "vi"




#Create data frame for creating PCAs - prep data step
ReadyForPCA <- prepData(rawData = MetaNOMISS,
                    nomVars = myNoms,
                    ordVars = NULL, # make sure to update this if there are ordinal variables
                    idVars = myIds,
                    moderators = NULL,
                    verbose = 2L,
                    nProcess = 4,
                    simMode = F)
y #the prepData code required a yes or no to continue, this tells it to continue

## Create principal component auxiliary variables:
```

```
PCAOut <- createPcAux(pcAuxData = ReadyForPCA,
                      nComps = c(0.5,0.5),
                      maxPolyPow = 3L,
                      doImputation = FALSE,
                      interactType = 1)
y #the createPcAux code required a yes or no to continue, this tells it to continue


## Merge the PC auxiliaries with the original data:
Meta5 <- mergePcAux(pcAuxData = PCAOut, rawData = Meta4) #use this data for imputation


#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES
AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES
AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES
AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
Meta5$FamSizeH <- Meta5$FamSizeC - sd(Meta5$FamSizeC, na.rm = T)
Meta5$FamSizeL <- Meta5$FamSizeC + sd(Meta5$FamSizeC, na.rm = T)
Meta5$AgeDifH <- Meta5$AgeDifC - sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeDifL <- Meta5$AgeDifC + sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeH <- Meta5$AgeC - sd(Meta5$AgeC, na.rm = T)
Meta5$AgeL <- Meta5$AgeC + sd(Meta5$AgeC, na.rm = T)
Meta5$WhitePercH <- Meta5$WhitePercC - sd(Meta5$WhitePercC, na.rm = T)
Meta5$WhitePercL <- Meta5$WhitePercC + sd(Meta5$WhitePercC, na.rm = T)
Meta5$GCSamePercH <- Meta5$GCSamePercC - sd(Meta5$GCSamePercC, na.rm = T)
Meta5$GCSamePercL <- Meta5$GCSamePercC + sd(Meta5$GCSamePercC, na.rm = T)
Meta5$MalePercH <- Meta5$MalePercC - sd(Meta5$MalePercC, na.rm = T)
Meta5$MalePercL <- Meta5$MalePercC + sd(Meta5$MalePercC, na.rm = T)
Meta5$RelBirthYoungPercH <- Meta5$RelBirthYoungPercC - sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$RelBirthYoungPercL <- Meta5$RelBirthYoungPercC + sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$NonPeerRev <- 1 #creating a variable where non-peer reviewed (mostly raw data) is coded as 1,
and peer reviewed as zero
Meta5$NonPeerRev[Meta5$PeerRev == 1] <- 0
Meta5$Concurrent <- 1 # creating a variable where concurrent PDT is 1, recollected is zero
Meta5$Concurrent[Meta5$Recollected == 1] <- 0
#measurement type X reporter interactions
Meta5$DSBMXCR <- Meta5$Variation*Meta5$ChildPDTReport
Meta5$DSBMXPR <- Meta5$Variation*Meta5$ParentPDTReport
Meta5$PBMXCR <- Meta5$Perception*Meta5$ChildPDTReport
Meta5$PBMXPR <- Meta5$Perception*Meta5$ParentPDTReport



#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
```

```
#Only run this the first time, and then comment it out after the test run
#Meta5 <- Meta5[sample(nrow(Meta5), 20),]


#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
predMatrix <- make.predictorMatrix(Meta5)
#changing the matrix so that only the PCAs are used as auxiliary variables for imputing
######re-do this every time to make sure column numbers match
predMatrix[,c(1:32, 44:63)] <- 0  #A value of 1 in this matrix indicates that the corresponding column variable is used to predict the corresponding row variable
#only the PCAs should have 1s, and they should go down all the rows
predMatrix #verify - always verify this, it will mess everything up if it is wrong - only PCAs should have 1s
#checking the imputation method for each variable - it only really matters for the variables with missing data
#if a nominal variable is set to pmm, change it from numeric to factor so that it will use logistic reg
impMethod <- make.method(Meta5)
impMethod


#IMPUTE THE DATA
#IMPUTE THE DATA
#IMPUTE THE DATA
imp <- mice(Meta5, print=FALSE, m=50, predictorMatrix=predMatrix, method=impMethod, seed=1234)



#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#this runs the analysis
Model1 <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE, data=Meta5)
#this summarizes the analysis
summary(Model1, digits=3, transf=transf.ztor)
#this saves the analysis
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Model 1.txt")
RE_model <- Model1
RE_model
sink(NULL)

#this calculates the ICCs
M1ICCs<- MetaVarCompsModel1(Model1)
#this saves the ICCs
```

```
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Model 1 ICCs.txt")
RE_model <- M1ICCs
RE_model
sink(NULL)


#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#no need to use the imputed data set here because there are no imputed values on the effect size
FailSafe <- fsn(Meta5$yi, Meta5$vi,type="Rosenthal", alpha=.05, digits = 4)
FailSafe
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Fail Safe Number.txt")
RE_model <- FailSafe
RE_model
sink(NULL)

#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#Build a model without within-source variance
NoLevel2Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                sigma2=c(0,NA, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the within-source variance
WithinSource <- anova(Model1,NoLevel2Var)
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Within Source Variance.txt")
RE_model <- WithinSource
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#Build a model without between-source variance
NoLevel3Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                sigma2=c(NA,0, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-source variance
BetweenSource<-anova(Model1,NoLevel3Var)
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Between Source Variance.txt")
RE_model <- BetweenSource
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
```

```
#Build a model without between-sample variance
NoLevel4Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,NA, 0), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-sample variance
BetweenSample<-anova(Model1,NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Between Sample Variance.txt")
RE_model <- BetweenSample
RE_model
sink(NULL)


#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#check frequencies of all nominal variables
sapply(Meta5[,c(5,6,8:23)], table)
table(Meta5$Variation, Meta5$ChildPDTReport)

#No UsedBeta, 0 and 2780
#No Recollected, 0 and 2780

#this runs the analysis by imputation (50 of them)
Model2 <- withold(imp, rma.mv(yi, vi,
mods = ~ USA+ #Level 4
FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC + AgeC + PeerRev
+ #Level 3
DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT + ResourcePDT + ControlPDT,
#Level 2
random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
Model2Pooled <- summary(pool(Model2))
Model2Pooled[-1] <- round(Model2Pooled[-1], digits=4)
Model2Pooled$LowerBound <- Model2Pooled$estimate - (Model2Pooled$std.error*2)
Model2Pooled$UpperBound <- Model2Pooled$estimate + (Model2Pooled$std.error*2)
Model2Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Model 2.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)


#retrieve the variance components for Model 2
DidWork <- MetaVarCompsModel2(Model2)
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Model 2 - variance components.txt")
RE_model <- DidWork
```

```
RE_model
sink(NULL)

#test the significance of the variance components in Model 2 - this is done using the function below. The function takes the model for each imputation, and constrains the variance at that level to be zero and then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50 imputations
#for the MetaVarCompsModel2LevelVarianceTest function, the needed arguments are (unrestricted model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model2NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ USA+ #Level 4
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA),
tdist=TRUE))
Model2NoLevel3Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ USA+ #Level 4
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA),
tdist=TRUE))
Model2NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ USA+ #Level 4
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0),
tdist=TRUE))
M2VarTest <- MetaVarCompsModel2LevelVarianceTest(Model2, Model2NoLevel2Var,
Model2NoLevel3Var, Model2NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Model 2 - variance components significance
tests.txt")
RE_model <- M2VarTest
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3

#No Model3, there were no parent reported PBM
```

```
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE

#HIGH PERCENT SIBY
#HIGH PERCENT SIBY
#HIGH PERCENT SIBY

HIGHSIBY <- withold(imp, rma.mv(yi, vi,
                    mods = ~ USA+ #Level 4
                      FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercH + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                      DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
HIGHSIBY <- summary(pool(HIGHSIBY))
HIGHSIBY[-1] <- round(HIGHSIBY[-1], digits=4)
HIGHSIBY

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - High SibY.txt")
RE_model <- HIGHSIBY
RE_model
sink(NULL)



#LOW PERCENT SIBY
#LOW PERCENT SIBY
#LOW PERCENT SIBY

LOWSIBY <- withold(imp, rma.mv(yi, vi,
                   mods = ~ USA+ #Level 4
                     FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercL + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                     DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                   random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
LOWSIBY <- summary(pool(LOWSIBY))
LOWSIBY[-1] <- round(LOWSIBY[-1], digits=4)
```

```
LOWSIBY

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - Low SibY.txt")
RE_model <- LOWSIBY
RE_model
sink(NULL)


#HIGH PERCENT SAME SEX
#HIGH PERCENT SAME SEX
#HIGH PERCENT SAME SEX

HIGHSAMESEX <- withold(imp, rma.mv(yi, vi,
                        mods = ~ USA+ #Level 4
                         FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercH +
AgeDifC + AgeC + PeerRev + #Level 3
                         DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                        random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
HIGHSAMESEX <- summary(pool(HIGHSAMESEX))
HIGHSAMESEX[-1] <- round(HIGHSAMESEX[-1], digits=4)
HIGHSAMESEX

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - High Percent Same Sex")
RE_model <- HIGHSAMESEX
RE_model
sink(NULL)


#LOW PERCENT SAME SEX
#LOW PERCENT SAME SEX
#LOW PERCENT SAME SEX

LOWSAMESEX <- withold(imp, rma.mv(yi, vi,
                       mods = ~ USA+ #Level 4
                         FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercL +
AgeDifC + AgeC + PeerRev + #Level 3
                         DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                       random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
LOWSAMESEX <- summary(pool(LOWSAMESEX))
LOWSAMESEX[-1] <- round(LOWSAMESEX[-1], digits=4)
```

```
LOWSAMESEX

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - Low Percent Same Sex")
RE_model <- LOWSAMESEX
RE_model
sink(NULL)


#MOM PDT
#MOM PDT
#MOM PDT

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - Mom PDT.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)


#Dad PDT
#Dad PDT
#Dad PDT

DADPDT <- withold(imp, rma.mv(yi, vi,
                      mods = ~ USA+ #Level 4
                        FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                        MomPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                      random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
DADPDT <- summary(pool(DADPDT))
DADPDT[-1] <- round(DADPDT[-1], digits=4)
DADPDT

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - Dad PDT.txt")
RE_model <- DADPDT
RE_model
sink(NULL)


#PARENT REPORT
#PARENT REPORT
#PARENT REPORT
```

```
#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - Parent Reported PDT.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)


#CHILD REPORT
#CHILD REPORT
#CHILD REPORT

CHILDREPORT <- withold(imp, rma.mv(yi, vi,
                        mods = ~ USA+ #Level 4
                          FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                          DadPDT + BothParsPDT + ParentPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                        random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
CHILDREPORT <- summary(pool(CHILDREPORT))
CHILDREPORT[-1] <- round(CHILDREPORT[-1], digits=4)
CHILDREPORT

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - Child Reported PDT.txt")
RE_model <- CHILDREPORT
RE_model
sink(NULL)




#PBM
#PBM
#PBM

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - PBM.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)




#DSBM
#DSBM
```

```
#DSBM

DSBM <- withold(imp, rma.mv(yi, vi,
                    mods = ~ USA+ #Level 4
                      FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC +
AgeC + PeerRev + #Level 3
                      DadPDT + BothParsPDT + ChildPDTReport + Perception + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
DSBM <- summary(pool(DSBM))
DSBM[-1] <- round(DSBM[-1], digits=4)
DSBM

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Stress/Stress - Simple Slopes - DSBM.txt")
RE_model <- DSBM
RE_model
sink(NULL)
```

```
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () ()       Four-level multilevel meta models          () () ()
# () () ()       testing links between Physical Health      () () ()
# () () ()       and absolute PDT                           () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()


#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
setwd("")

#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#for info on withold(), see https://www.metafor-
project.org/doku.php/tips:multiple_imputation_with_mice_and_metafor for more details
#I created the MetaModelsDescriptiveData(), MetaVarCompsModel1(), MetaVarCompsModel2(), and
MetaVarCompsModel2LevelVarianceTest()
#functions - more notes are available in the original file "Meta Analysis Functions.R"
library(metafor)
library(PcAux)
library(mice)
library(psych)
library(tidyverse)
library(car)

#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
WholeMeta <- read.csv(file = "Programs and Data/FinalCleanPDTMeta.csv", header = T)
```

```
#SUBSET A NEW DATA FRAME THAT HAS ONLY CAPhysical Health WITH ABSOLUTE PDT AND Physical Health
#SUBSET A NEW DATA FRAME THAT HAS ONLY CAPhysical Health WITH ABSOLUTE PDT AND Physical Health
#SUBSET A NEW DATA FRAME THAT HAS ONLY CAPhysical Health WITH ABSOLUTE PDT AND Physical Health
Meta1 <- subset(x = WholeMeta, subset = MagnitudeDirectional == 0 & Outcome == 66)



#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
Meta2 <- Meta1[,

c("ESNum","SampleSize","Correlation","ArticleID","PeerRev","USA","DataID","FamSize","AgeDif",

"UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","ControlPDT",

"ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","Perception",
        "Recollected","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")]

MetaModelsDescriptiveData(Meta2) # this saves the descriptive stats for the non-imputed model
#we take descriptive stats from this data frame so that it includes missing data and the non-centered versions of continous variables



#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#we will drop the non-centered versions of continuous variables, because we want the imputation to be based on the centered versions
Meta3 <- Meta2
attach(Meta3)
Meta3$FamSizeC <- FamSize-mean(FamSize, na.rm = T)
Meta3$AgeDifC <- AgeDif-mean(AgeDif, na.rm = T)
Meta3$AgeC <- Age-mean(Age, na.rm = T)
Meta3$WhitePercC <- WhitePerc-mean(WhitePerc, na.rm = T)
Meta3$GCSamePercC <- GCSamePerc-mean(GCSamePerc, na.rm = T)
Meta3$MalePercC <- MalePerc-mean(MalePerc, na.rm = T)
Meta3$RelBirthYoungPercC <- RelBirthYoungPerc-mean(RelBirthYoungPerc, na.rm = T)
detach(Meta3)
DropVars <- names(Meta3) %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
Meta3 <- Meta3[!DropVars]
DropVars <- Meta3 %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
```

```
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#Do this before the PCAs, it doesn't like doing it on an imputed data frame, so do it first
Meta4 <- escalc(measure = "ZCOR", ri =Correlation,
          ni = SampleSize, data = Meta3, vtype = "LS", append = TRUE)



#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#re-do this step every time

MetaNOMISS <- Meta4[, which(colMeans(is.na(Meta4)) ==.00)]#creating data frame with only variables
with no missing values

names(MetaNOMISS)#I run this to get the names and then sort those into the type of variable below

## Prepare the data by first sorting out what types of variables I have - continuous variables are left off
the code but listed below
myNoms   <-
c("PeerRev","USA","UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","
ControlPDT","ChildPDTReport","ParentPDTReport", "MomPDT","DadPDT","BothParsPDT","Variation",
"Perception", "Recollected")
myIds    <- c("ESNum", "ArticleID", "DataID")
#myOrds   <- c() #none this time
#Continuous - "SampleSize", "Correlation","AgeC","WhitePercC","MalePercC","yi","vi","FamSizeC"



#Create data frame for creating PCAs - prep data step
ReadyForPCA <- prepData(rawData = MetaNOMISS,
                  nomVars = myNoms,
                  ordVars = NULL, # make sure to update this if there are ordinal variables
                  idVars = myIds,
                  moderators = NULL,
                  verbose = 2L,
                  nProcess = 4,
                  simMode = F)
y #the prepData code required a yes or no to continue, this tells it to continue
```

```
## Create principal component auxiliary variables:
PCAOut <- createPcAux(pcAuxData = ReadyForPCA,
                      nComps = c(0.5,0.5),
                      maxPolyPow = 3L,
                      doImputation = FALSE,
                      interactType = 1)
y #the createPcAux code required a yes or no to continue, this tells it to continue

## Merge the PC auxiliaries with the original data:
Meta5 <- mergePcAux(pcAuxData = PCAOut, rawData = Meta4) #use this data for imputation


#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES
AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES
AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES
AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
Meta5$FamSizeH <- Meta5$FamSizeC - sd(Meta5$FamSizeC, na.rm = T)
Meta5$FamSizeL <- Meta5$FamSizeC + sd(Meta5$FamSizeC, na.rm = T)
Meta5$AgeDifH <- Meta5$AgeDifC - sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeDifL <- Meta5$AgeDifC + sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeH <- Meta5$AgeC - sd(Meta5$AgeC, na.rm = T)
Meta5$AgeL <- Meta5$AgeC + sd(Meta5$AgeC, na.rm = T)
Meta5$WhitePercH <- Meta5$WhitePercC - sd(Meta5$WhitePercC, na.rm = T)
Meta5$WhitePercL <- Meta5$WhitePercC + sd(Meta5$WhitePercC, na.rm = T)
Meta5$GCSamePercH <- Meta5$GCSamePercC - sd(Meta5$GCSamePercC, na.rm = T)
Meta5$GCSamePercL <- Meta5$GCSamePercC + sd(Meta5$GCSamePercC, na.rm = T)
Meta5$MalePercH <- Meta5$MalePercC - sd(Meta5$MalePercC, na.rm = T)
Meta5$MalePercL <- Meta5$MalePercC + sd(Meta5$MalePercC, na.rm = T)
Meta5$RelBirthYoungPercH <- Meta5$RelBirthYoungPercC - sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$RelBirthYoungPercL <- Meta5$RelBirthYoungPercC + sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$NonPeerRev <- 1 #creating a variable where non-peer reviewed (mostly raw data) is coded as 1,
and peer reviewed as zero
Meta5$NonPeerRev[Meta5$PeerRev == 1] <- 0
Meta5$Concurrent <- 1 # creating a variable where concurrent PDT is 1, recollected is zero
Meta5$Concurrent[Meta5$Recollected == 1] <- 0
#measurement type X reporter interactions
Meta5$DSBMXCR <- Meta5$Variation*Meta5$ChildPDTReport
Meta5$DSBMXPR <- Meta5$Variation*Meta5$ParentPDTReport
Meta5$PBMXCR <- Meta5$Perception*Meta5$ChildPDTReport
Meta5$PBMXPR <- Meta5$Perception*Meta5$ParentPDTReport



#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
```

```
#Only run this the first time, and then comment it out after the test run
#Meta5 <- Meta5[sample(nrow(Meta5), 100),]


#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
predMatrix <- make.predictorMatrix(Meta5)
#changing the matrix so that only the PCAs are used as auxiliary variables for imputing
######re-do this every time to make sure column numbers match
predMatrix[,c(1:32, 39:58)] <- 0  #A value of 1 in this matrix indicates that the corresponding column variable is used to predict the corresponding row variable
#only the PCAs should have 1s, and they should go down all the rows
predMatrix #verify - always verify this, it will mess everything up if it is wrong - only PCAs should have 1s
#checking the imputation method for each variable - it only really matters for the variables with missing data
#if a nominal variable is set to pmm, change it from numeric to factor so that it will use logistic reg
impMethod <- make.method(Meta5)
impMethod


#IMPUTE THE DATA
#IMPUTE THE DATA
#IMPUTE THE DATA
imp <- mice(Meta5, print=FALSE, m=50, predictorMatrix=predMatrix, method=impMethod, seed=1234)



#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#this runs the analysis
Model1 <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE, data=Meta5)
#this summarizes the analysis
summary(Model1, digits=3, transf=transf.ztor)
#this saves the analysis
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Model 1.txt")
RE_model <- Model1
RE_model
sink(NULL)

#this calculates the ICCs
M1ICCs<- MetaVarCompsModel1(Model1)
#this saves the ICCs
```

```
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Model 1 ICCs.txt")
RE_model <- M1ICCs
RE_model
sink(NULL)



#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#no need to use the imputed data set here because there are no imputed values on the effect size
FailSafe <- fsn(Meta5$yi, Meta5$vi,type="Rosenthal", alpha=.05, digits = 4)
FailSafe
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Fail Safe Number.txt")
RE_model <- FailSafe
RE_model
sink(NULL)



#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#Build a model without within-source variance
NoLevel2Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(0,NA, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the within-source variance
WithinSource <- anova(Model1,NoLevel2Var)
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Within Source
Variance.txt")
RE_model <- WithinSource
RE_model
sink(NULL)



#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#Build a model without between-source variance
NoLevel3Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,0, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-source variance
BetweenSource<-anova(Model1,NoLevel3Var)
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Between Source
Variance.txt")
```

```
RE_model <- BetweenSource
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#Build a model without between-sample variance
NoLevel4Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,NA, 0), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-sample variance
BetweenSample<-anova(Model1,NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Between Sample
Variance.txt")
RE_model <- BetweenSample
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#check frequencies of all nominal variables
sapply(Meta5[,c(5,6,8:23)], table)
table(Meta5$Variation, Meta5$ChildPDTReport)

#There is not variation at the sample level, so no USA
#No PeerRev 532 and 3
#No UsedBeta 532 and 3
#No OverallPDT 507   and 28
#PosPDT is the reference group for domain
#No Recollected 0 and 535

#this runs the analysis by imputation (50 of them)
Model2 <- withold(imp, rma.mv(yi, vi,
                              mods = ~
                                FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC
+ AgeC +  #Level 3
                                DadPDT + BothParsPDT + ChildPDTReport + Variation + NegPDT + ResourcePDT +
ControlPDT, #Level 2
                              random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
Model2Pooled <- summary(pool(Model2))
Model2Pooled[-1] <- round(Model2Pooled[-1], digits=4)
Model2Pooled$LowerBound <- Model2Pooled$estimate - (Model2Pooled$std.error*2)
```

```
Model2Pooled$UpperBound <- Model2Pooled$estimate + (Model2Pooled$std.error*2)
Model2Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Model 2.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)



#retrieve the variance components for Model 2
DidWork <- MetaVarCompsModel2(Model2)
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Model 2 - variance
components.txt")
RE_model <- DidWork
RE_model
sink(NULL)

#test the significance of the variance components in Model 2 - this is done using the function below. The
function takes the model for each imputation, and constrains the variance at that level to be zero and
then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50
imputations
#for the MetaVarCompsModel2LevelVarianceTest function, the needed arguments are (unrestricted
model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model2NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                             mods = ~
                               FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC +  #Level 3
                               DadPDT + BothParsPDT + ChildPDTReport + Variation + NegPDT + ResourcePDT +
ControlPDT, #Level 2
                             random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA),
tdist=TRUE))
Model2NoLevel3Var <- withold(imp, rma.mv(yi, vi,
                             mods = ~
                               FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC +  #Level 3
                               DadPDT + BothParsPDT + ChildPDTReport + Variation + NegPDT + ResourcePDT +
ControlPDT, #Level 2
                             random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA),
tdist=TRUE))
Model2NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                             mods = ~
                               FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC +  #Level 3
                               DadPDT + BothParsPDT + ChildPDTReport + Variation + NegPDT + ResourcePDT +
ControlPDT, #Level 2
```

```
                              random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0),
tdist=TRUE))
M2VarTest <- MetaVarCompsModel2LevelVarianceTest(Model2, Model2NoLevel2Var,
Model2NoLevel3Var, Model2NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Model 2 - variance
components significance tests.txt")
RE_model <- M2VarTest
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3

#There were not enough in each cell for the interaction, Parent PBM had 3 cases



#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE

#POS PDT
#POS PDT
#POS PDT
#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Simple Slopes - Pos
PDT.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)



#CONTROL PDT
#CONTROL PDT
#CONTROL PDT

#this runs the analysis by imputation (50 of them)
CONTROLPDT <- withold(imp, rma.mv(yi, vi,
                          mods = ~
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC +  #Level 3
```

```
                    DadPDT + BothParsPDT + ChildPDTReport + Variation + NegPDT + ResourcePDT +
PosPDT, #Level 2
                  random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
CONTROLPDT <- summary(pool(CONTROLPDT))
CONTROLPDT[-1] <- round(CONTROLPDT[-1], digits=4)
CONTROLPDT

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Physical Health/Physical Health - Simple Slopes - Control
PDT.txt")
RE_model <- CONTROLPDT
RE_model
sink(NULL)
```

```
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()       Four-level multilevel meta models          () () ()
# () () ()       testing links between Mental Health        () () ()
# () () ()       and absolute PDT                    () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()


#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
setwd("")

#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#for info on withold(), see https://www.metafor-
project.org/doku.php/tips:multiple_imputation_with_mice_and_metafor for more details
#I created the MetaModelsDescriptiveData(), MetaVarCompsModel1(), MetaVarCompsModel2(), and
MetaVarCompsModel2LevelVarianceTest()
#functions - more notes are available in the original file "Meta Analysis Functions.R"
library(metafor)
library(PcAux)
library(mice)
library(psych)
library(tidyverse)
library(car)

#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
WholeMeta <- read.csv(file = "Programs and Data/FinalCleanPDTMeta.csv", header = T)
```

```
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Mental Health
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Mental Health
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Mental Health
Meta1 <- subset(x = WholeMeta, subset = MagnitudeDirectional == 0 & Outcome == 65)



#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
Meta2 <- Meta1[,

c("ESNum","SampleSize","Correlation","ArticleID","PeerRev","USA","DataID","FamSize","AgeDif",

"UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","ControlPDT",

"ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","Perception",
        "Recollected","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")]

MetaModelsDescriptiveData(Meta2) # this saves the descriptive stats for the non-imputed model
#we take descriptive stats from this data frame so that it includes missing data and the non-centered
versions of continous variables



#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#we will drop the non-centered versions of continuous variables, because we want the imputation to be
based on the centered versions
Meta3 <- Meta2
attach(Meta3)
Meta3$FamSizeC <- FamSize-mean(FamSize, na.rm = T)
Meta3$AgeDifC <- AgeDif-mean(AgeDif, na.rm = T)
Meta3$AgeC <- Age-mean(Age, na.rm = T)
Meta3$WhitePercC <- WhitePerc-mean(WhitePerc, na.rm = T)
Meta3$GCSamePercC <- GCSamePerc-mean(GCSamePerc, na.rm = T)
Meta3$MalePercC <- MalePerc-mean(MalePerc, na.rm = T)
Meta3$RelBirthYoungPercC <- RelBirthYoungPerc-mean(RelBirthYoungPerc, na.rm = T)
detach(Meta3)
DropVars <- names(Meta3) %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
Meta3 <- Meta3[!DropVars]
DropVars <- Meta3 %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
```

```
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#Do this before the PCAs, it doesn't like doing it on an imputed data frame, so do it first
Meta4 <- escalc(measure = "ZCOR", ri =Correlation,
            ni = SampleSize, data = Meta3, vtype = "LS", append = TRUE)




#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#re-do this step every time

MetaNOMISS <- Meta4[, which(colMeans(is.na(Meta4)) ==.00)]#creating data frame with only variables
with no missing values

names(MetaNOMISS)#I run this to get the names and then sort those into the type of variable below

## Prepare the data by first sorting out what types of variables I have - continuous variables are left off
the code but listed below
myNoms   <-
c("PeerRev","USA","UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","
ControlPDT","ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","P
erception","Recollected")
myIds    <- c("ESNum", "ArticleID", "DataID")
#myOrds   <- c() #none this time
#Continuous -
"SampleSize","Correlation","FamSizeC","AgeDifC","AgeC","WhitePercC","GCSamePercC","MalePercC","y
i","vi"




#Create data frame for creating PCAs - prep data step
ReadyForPCA <- prepData(rawData = MetaNOMISS,
                    nomVars = myNoms,
                    ordVars = NULL, # make sure to update this if there are ordinal variables
                    idVars = myIds,
                    moderators = NULL,
                    verbose = 2L,
                    nProcess = 4,
                    simMode = F)
y #the prepData function often needs a yes or no to continue, this tells it to keep going

## Create principal component auxiliary variables:
```

```
PCAOut <- createPcAux(pcAuxData = ReadyForPCA,
                      nComps = c(0.5,0.5),
                      maxPolyPow = 3L,
                      doImputation = FALSE,
                      interactType = 1)
y #the createPcAux function often needs a yes or no to continue, this tells it to keep going


## Merge the PC auxiliaries with the original data:
Meta5 <- mergePcAux(pcAuxData = PCAOut, rawData = Meta4) #use this data for imputation


#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
Meta5$FamSizeH <- Meta5$FamSizeC - sd(Meta5$FamSizeC, na.rm = T)
Meta5$FamSizeL <- Meta5$FamSizeC + sd(Meta5$FamSizeC, na.rm = T)
Meta5$AgeDifH <- Meta5$AgeDifC - sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeDifL <- Meta5$AgeDifC + sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeH <- Meta5$AgeC - sd(Meta5$AgeC, na.rm = T)
Meta5$AgeL <- Meta5$AgeC + sd(Meta5$AgeC, na.rm = T)
Meta5$WhitePercH <- Meta5$WhitePercC - sd(Meta5$WhitePercC, na.rm = T)
Meta5$WhitePercL <- Meta5$WhitePercC + sd(Meta5$WhitePercC, na.rm = T)
Meta5$GCSamePercH <- Meta5$GCSamePercC - sd(Meta5$GCSamePercC, na.rm = T)
Meta5$GCSamePercL <- Meta5$GCSamePercC + sd(Meta5$GCSamePercC, na.rm = T)
Meta5$MalePercH <- Meta5$MalePercC - sd(Meta5$MalePercC, na.rm = T)
Meta5$MalePercL <- Meta5$MalePercC + sd(Meta5$MalePercC, na.rm = T)
Meta5$RelBirthYoungPercH <- Meta5$RelBirthYoungPercC - sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$RelBirthYoungPercL <- Meta5$RelBirthYoungPercC + sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$NonPeerRev <- 1 #creating a variable where non-peer reviewed (mostly raw data) is coded as 1, and peer reviewed as zero
Meta5$NonPeerRev[Meta5$PeerRev == 1] <- 0
Meta5$Concurrent <- 1 # creating a variable where concurrent PDT is 1, recollected is zero
Meta5$Concurrent[Meta5$Recollected == 1] <- 0
#measurement type X reporter interactions
Meta5$DSBMXCR <- Meta5$Variation*Meta5$ChildPDTReport
Meta5$DSBMXPR <- Meta5$Variation*Meta5$ParentPDTReport
Meta5$PBMXCR <- Meta5$Perception*Meta5$ChildPDTReport
Meta5$PBMXPR <- Meta5$Perception*Meta5$ParentPDTReport



#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
```

```
#Only run this the first time, and then comment it out after the test run
#Meta5 <- Meta5[sample(nrow(Meta5), 100),]


#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
predMatrix <- make.predictorMatrix(Meta5)
#changing the matrix so that only the PCAs are used as auxiliary variables for imputing
######re-do this every time to make sure column numbers match
predMatrix[,c(1:32, 43:62)] <- 0  #A value of 1 in this matrix indicates that the corresponding column variable is used to predict the corresponding row variable
#only the PCAs should have 1s, and they should go down all the rows
predMatrix #verify - always verify this, it will mess everything up if it is wrong - only PCAs should have 1s
#checking the imputation method for each variable - it only really matters for the variables with missing data
#if a nominal variable is set to pmm, change it from numeric to factor so that it will use logistic reg
impMethod <- make.method(Meta5)
impMethod


#IMPUTE THE DATA
#IMPUTE THE DATA
#IMPUTE THE DATA
imp <- mice(Meta5, print=FALSE, m=50, predictorMatrix=predMatrix, method=impMethod, seed=1234)



#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#this runs the analysis
Model1 <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE, data=Meta5)
#this summarizes the analysis
summary(Model1, digits=3, transf=transf.ztor)
#this saves the analysis
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 1.txt")
RE_model <- Model1
RE_model
sink(NULL)

#this calculates the ICCs
M1ICCs<- MetaVarCompsModel1(Model1)
#this saves the ICCs
```

```
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 1 ICCs.txt")
RE_model <- M1ICCs
RE_model
sink(NULL)


#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#no need to use the imputed data set here because there are no imputed values on the effect size
FailSafe <- fsn(Meta5$yi, Meta5$vi,type="Rosenthal", alpha=.05, digits = 4)
FailSafe
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Fail Safe Number.txt")
RE_model <- FailSafe
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#Build a model without within-source variance
NoLevel2Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                sigma2=c(0,NA, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the within-source variance
WithinSource <- anova(Model1,NoLevel2Var)
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Within Source Variance.txt")
RE_model <- WithinSource
RE_model
sink(NULL)



#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#Build a model without between-source variance
NoLevel3Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                sigma2=c(NA,0, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-source variance
BetweenSource<-anova(Model1,NoLevel3Var)
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Between Source
Variance.txt")
RE_model <- BetweenSource
RE_model
sink(NULL)
```

```
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#Build a model without between-sample variance
NoLevel4Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                sigma2=c(NA,NA, 0), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-sample variance
BetweenSample<-anova(Model1,NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Between Sample
Variance.txt")
RE_model <- BetweenSample
RE_model
sink(NULL)


#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#check frequencies of all nominal variables
sapply(Meta5[,c(5,6,8:23)], table)
table(Meta5$Variation, Meta5$ChildPDTReport)

#this runs the analysis by imputation (50 of them)
Model2 <- withold(imp, rma.mv(yi, vi,
                    mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                      GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                      BothParsPDT   + ChildPDTReport +
                      Variation     + NegPDT      + ResourcePDT + ControlPDT,
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
Model2Pooled <- summary(pool(Model2))
Model2Pooled[-1] <- round(Model2Pooled[-1], digits=4)
Model2Pooled$LowerBound <- Model2Pooled$estimate - (Model2Pooled$std.error*2)
Model2Pooled$UpperBound <- Model2Pooled$estimate + (Model2Pooled$std.error*2)
Model2Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 2.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)


#retrieve the variance components for Model 2
DidWork <- MetaVarCompsModel2(Model2)
```

```
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 2 - variance components.txt")
RE_model <- DidWork
RE_model
sink(NULL)

#test the significance of the variance components in Model 2 - this is done using the function below. The function takes the model for each imputation, and constrains the variance at that level to be zero and then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50 imputations
#for the MetaVarCompsModel2LevelVarianceTest function, the needed arguments are (unrestricted model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model2NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                         mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                          GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                          BothParsPDT   + ChildPDTReport +
                          Variation     + NegPDT      + ResourcePDT + ControlPDT,
                         random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA), tdist=TRUE))
Model2NoLevel3Var <- withold(imp, rma.mv(yi, vi,
                         mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                          GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                          BothParsPDT   + ChildPDTReport +
                          Variation     + NegPDT      + ResourcePDT + ControlPDT,
                         random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA), tdist=TRUE))
Model2NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                         mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                          GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                          BothParsPDT   + ChildPDTReport +
                          Variation     + NegPDT      + ResourcePDT + ControlPDT,
                         random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0), tdist=TRUE))
M2VarTest <- MetaVarCompsModel2LevelVarianceTest(Model2, Model2NoLevel2Var, Model2NoLevel3Var, Model2NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 2 - variance components significance tests.txt")
RE_model <- M2VarTest
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3
```

```
#this runs the analysis by imputation (50 of them)
Model3 <- withold(imp, rma.mv(yi, vi,
                  mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                   GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                   BothParsPDT   + ChildPDTReport +
                   Variation     + NegPDT      + ResourcePDT + ControlPDT + Variation:ChildPDTReport,
                  random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
Model3Pooled <- summary(pool(Model3))
Model3Pooled[-1] <- round(Model3Pooled[-1], digits=4)
Model3Pooled$LowerBound <- Model3Pooled$estimate - (Model3Pooled$std.error*2)
Model3Pooled$UpperBound <- Model3Pooled$estimate + (Model3Pooled$std.error*2)
Model3Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 3.txt")
RE_model <- Model3Pooled
RE_model
sink(NULL)


#test the significance of the variance components in Model 3 - this is done using the function below. The function takes the model for each imputation, and constrains the variance at that level to be zero and then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50 imputations
#for the MetaVarCompsModel3LevelVarianceTest function, the needed arguments are (unrestricted model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model3NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                           GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                           BothParsPDT   + ChildPDTReport +
                           Variation     + NegPDT      + ResourcePDT + ControlPDT +
Variation:ChildPDTReport,
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA),
tdist=TRUE))
Model3NoLevel3Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                           GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                           BothParsPDT   + ChildPDTReport +
                           Variation     + NegPDT      + ResourcePDT + ControlPDT +
Variation:ChildPDTReport,
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA),
tdist=TRUE))
Model3NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                           GCSamePercC    + AgeDifC             + AgeC + DadPDT +
```

```
                              BothParsPDT   + ChildPDTReport +
                              Variation     + NegPDT      + ResourcePDT + ControlPDT +
Variation:ChildPDTReport,
                            random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0),
tdist=TRUE))
M3VarTest <- MetaVarCompsModel2LevelVarianceTest(Model3, Model3NoLevel2Var,
Model3NoLevel3Var, Model3NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Model 3 - variance
components significance tests.txt")
RE_model <- M3VarTest
RE_model
sink(NULL)


#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE

#HIGH PERCENTAGE SAME SEX
#HIGH PERCENTAGE SAME SEX
#HIGH PERCENTAGE SAME SEX
GCHigh <- withold(imp, rma.mv(yi, vi,
                    mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                      GCSamePercH    + AgeDifC             + AgeC + DadPDT +
                      BothParsPDT   + ChildPDTReport +
                      Variation     + NegPDT      + ResourcePDT + ControlPDT,
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
GCHigh <- summary(pool(GCHigh))
GCHigh[-1] <- round(GCHigh[-1], digits=4)
GCHigh

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Same Sex
Percent High.txt")
RE_model <- GCHigh
RE_model
sink(NULL)


#LOW PERCENTAGE SAME SEX
#LOW PERCENTAGE SAME SEX
#LOW PERCENTAGE SAME SEX
GCLow <- withold(imp, rma.mv(yi, vi,
                    mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
```

```
                    GCSamePercL    + AgeDifC             + AgeC + DadPDT +
                    BothParsPDT   + ChildPDTReport +
                    Variation     + NegPDT      + ResourcePDT + ControlPDT,
                   random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
GCLow <- summary(pool(GCLow))
GCLow[-1] <- round(GCLow[-1], digits=4)
GCLow

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Same Sex
Percent Low.txt")
RE_model <- GCLow
RE_model
sink(NULL)



#HIGH AGE SPACING
#HIGH AGE SPACING
#HIGH AGE SPACING
AgeDifHigh <- withold(imp, rma.mv(yi, vi,
                        mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                          GCSamePercC    + AgeDifH             + AgeC + DadPDT +
                          BothParsPDT   + ChildPDTReport +
                          Variation     + NegPDT      + ResourcePDT + ControlPDT,
                        random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
AgeDifHigh <- summary(pool(AgeDifHigh))
AgeDifHigh[-1] <- round(AgeDifHigh[-1], digits=4)
AgeDifHigh

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Age Spacing
High.txt")
RE_model <- AgeDifHigh
RE_model
sink(NULL)


#LOW AGE SPACING
#LOW AGE SPACING
#LOW AGE SPACING
AgeDifLow <- withold(imp, rma.mv(yi, vi,
                       mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                         GCSamePercC    + AgeDifL             + AgeC + DadPDT +
                         BothParsPDT   + ChildPDTReport +
                         Variation     + NegPDT      + ResourcePDT + ControlPDT,
```

```
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
AgeDifLow <- summary(pool(AgeDifLow))
AgeDifLow[-1] <- round(AgeDifLow[-1], digits=4)
AgeDifLow

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Age Spacing
Low.txt")
RE_model <- AgeDifLow
RE_model
sink(NULL)



#PDT FROM MOM
#PDT FROM MOM
#PDT FROM MOM
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - PDT from
Mom.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)



#PDT FROM DAD
#PDT FROM DAD
#PDT FROM DAD
PDTDad <- withold(imp, rma.mv(yi, vi,
                      mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                        GCSamePercC    + AgeDifC             + AgeC + MomPDT +
                        BothParsPDT   + ChildPDTReport +
                        Variation     + NegPDT      + ResourcePDT + ControlPDT,
                      random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
PDTDad <- summary(pool(PDTDad))
PDTDad[-1] <- round(PDTDad[-1], digits=4)
PDTDad

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - PDT from
Dad.txt")
RE_model <- PDTDad
RE_model
sink(NULL)
```

```
#PDT FROM BOTH PARENTS
#PDT FROM BOTH PARENTS
#PDT FROM BOTH PARENTS
PDTBoth <- withold(imp, rma.mv(yi, vi,
                      mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                       GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                       MomPDT   + ChildPDTReport +
                       Variation     + NegPDT      + ResourcePDT + ControlPDT,
                      random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
PDTBoth <- summary(pool(PDTBoth))
PDTBoth[-1] <- round(PDTBoth[-1], digits=4)
PDTBoth

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - PDT from
Both Parents.txt")
RE_model <- PDTBoth
RE_model
sink(NULL)



#POS PDT
#POS PDT
#POS PDT
sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Pos PDT.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)


#NEG PDT
#NEG PDT
#NEG PDT
PDTNeg <- withold(imp, rma.mv(yi, vi,
                      mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                       GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                       BothParsPDT   + ChildPDTReport +
                       Variation     + PosPDT      + ResourcePDT + ControlPDT,
                      random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
PDTNeg <- summary(pool(PDTNeg))
PDTNeg[-1] <- round(PDTNeg[-1], digits=4)
PDTNeg

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Neg PDT.txt")
```

```
RE_model <- PDTNeg
RE_model
sink(NULL)


#RESOURCE PDT
#RESOURCE PDT
#RESOURCE PDT
PDTResource <- withold(imp, rma.mv(yi, vi,
                        mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                         GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                         BothParsPDT   + ChildPDTReport +
                         Variation     + NegPDT       + PosPDT + ControlPDT,
                        random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
PDTResource <- summary(pool(PDTResource))
PDTResource[-1] <- round(PDTResource[-1], digits=4)
PDTResource

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Resource
PDT.txt")
RE_model <- PDTResource
RE_model
sink(NULL)


#CONTROL PDT
#CONTROL PDT
#CONTROL PDT
PDTControl <- withold(imp, rma.mv(yi, vi,
                       mods = ~ FamSizeC      + WhitePercC + MalePercC    + RelBirthYoungPercC +
                        GCSamePercC    + AgeDifC             + AgeC + DadPDT +
                        BothParsPDT   + ChildPDTReport +
                        Variation     + NegPDT       + ResourcePDT + PosPDT,
                       random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
PDTControl <- summary(pool(PDTControl))
PDTControl[-1] <- round(PDTControl[-1], digits=4)
PDTControl

sink(file = "Findings/Imputed Data Findings/Mental Health/Mental Health - Simple Slopes - Control
PDT.txt")
RE_model <- PDTControl
RE_model
sink(NULL)
```

```
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()       Four-level multilevel meta models          () () ()
# () () ()       testing links between SES                  () () ()
# () () ()       and absolute PDT                        () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()


#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
setwd("")

#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#for info on withold(), see https://www.metafor-
project.org/doku.php/tips:multiple_imputation_with_mice_and_metafor for more details
#I created the MetaModelsDescriptiveData(), MetaVarCompsModel1(), MetaVarCompsModel2(), and
MetaVarCompsModel2LevelVarianceTest()
#functions - more notes are available in the original file "Meta Analysis Functions.R"
library(metafor)
library(PcAux)
library(mice)
library(psych)
library(tidyverse)
library(car)

#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
WholeMeta <- read.csv(file = "Programs and Data/FinalCleanPDTMeta.csv", header = T)
```

```
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND SES
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND SES
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND SES
Meta1 <- subset(x = WholeMeta, subset = MagnitudeDirectional == 0 & Outcome == 17)



#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
Meta2 <- Meta1[,

c("ESNum","SampleSize","Correlation","ArticleID","PeerRev","USA","DataID","FamSize","AgeDif",

"UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","ControlPDT",

"ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","Perception",
        "Recollected","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")]

MetaModelsDescriptiveData(Meta2) # this saves the descriptive stats for the non-imputed model
#we take descriptive stats from this data frame so that it includes missing data and the non-centered
versions of continous variables



#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#we will drop the non-centered versions of continuous variables, because we want the imputation to be
based on the centered versions
Meta3 <- Meta2
attach(Meta3)
Meta3$FamSizeC <- FamSize-mean(FamSize, na.rm = T)
Meta3$AgeDifC <- AgeDif-mean(AgeDif, na.rm = T)
Meta3$AgeC <- Age-mean(Age, na.rm = T)
Meta3$WhitePercC <- WhitePerc-mean(WhitePerc, na.rm = T)
Meta3$GCSamePercC <- GCSamePerc-mean(GCSamePerc, na.rm = T)
Meta3$MalePercC <- MalePerc-mean(MalePerc, na.rm = T)
Meta3$RelBirthYoungPercC <- RelBirthYoungPerc-mean(RelBirthYoungPerc, na.rm = T)
detach(Meta3)
DropVars <- names(Meta3) %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
Meta3 <- Meta3[!DropVars]
DropVars <- Meta3 %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
```

```
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#Do this before the PCAs, it doesn't like doing it on an imputed data frame, so do it first
Meta4 <- escalc(measure = "ZCOR", ri =Correlation,
            ni = SampleSize, data = Meta3, vtype = "LS", append = TRUE)




#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#re-do this step every time

MetaNOMISS <- Meta4[, which(colMeans(is.na(Meta4)) ==.00)]#creating data frame with only variables
with no missing values

names(MetaNOMISS)#I run this to get the names and then sort those into the type of variable below

## Prepare the data by first sorting out what types of variables I have - continuous variables are left off
the code but listed below
myNoms   <- c("PeerRev"  ,     "USA" ,  "UsedBeta"  ,     "OverallPDT"  ,   "PosPDT" ,       "NegPDT"   ,
"EvalPDT"   ,     "ResourcePDT"   ,  "FairPDT"    , "ControlPDT"   ,   "ChildPDTReport",  "ParentPDTReport",
"MomPDT"    ,    "DadPDT"    ,    "BothParsPDT",    "Variation"     ,"Perception"  ,   "Recollected")
myIds    <- c("ESNum", "ArticleID", "DataID")
#myOrds   <- c() #none this time
#Continuous -  "SampleSize"     "Correlation"    "FamSizeC"       "yi"            "vi"




#Create data frame for creating PCAs - prep data step
ReadyForPCA <- prepData(rawData = MetaNOMISS,
                  nomVars = myNoms,
                  ordVars = NULL, # make sure to update this if there are ordinal variables
                  idVars = myIds,
                  moderators = NULL,
                  verbose = 2L,
                  nProcess = 4,
                  simMode = F)
y #the prepData function often needs a yes or no to continue, this tells it to keep going

## Create principal component auxiliary variables:
PCAOut <- createPcAux(pcAuxData = ReadyForPCA,
                  nComps = c(0.5,0.5),
                  maxPolyPow = 3L,
```

```
            doImputation = FALSE,
            interactType = 1)
y #the createPcAux function often needs a yes or no to continue, this tells it to keep going


## Merge the PC auxiliaries with the original data:
Meta5 <- mergePcAux(pcAuxData = PCAOut, rawData = Meta4) #use this data for imputation


#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
Meta5$FamSizeH <- Meta5$FamSizeC - sd(Meta5$FamSizeC, na.rm = T)
Meta5$FamSizeL <- Meta5$FamSizeC + sd(Meta5$FamSizeC, na.rm = T)
Meta5$AgeDifH <- Meta5$AgeDifC - sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeDifL <- Meta5$AgeDifC + sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeH <- Meta5$AgeC - sd(Meta5$AgeC, na.rm = T)
Meta5$AgeL <- Meta5$AgeC + sd(Meta5$AgeC, na.rm = T)
Meta5$WhitePercH <- Meta5$WhitePercC - sd(Meta5$WhitePercC, na.rm = T)
Meta5$WhitePercL <- Meta5$WhitePercC + sd(Meta5$WhitePercC, na.rm = T)
Meta5$GCSamePercH <- Meta5$GCSamePercC - sd(Meta5$GCSamePercC, na.rm = T)
Meta5$GCSamePercL <- Meta5$GCSamePercC + sd(Meta5$GCSamePercC, na.rm = T)
Meta5$MalePercH <- Meta5$MalePercC - sd(Meta5$MalePercC, na.rm = T)
Meta5$MalePercL <- Meta5$MalePercC + sd(Meta5$MalePercC, na.rm = T)
Meta5$RelBirthYoungPercH <- Meta5$RelBirthYoungPercC - sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$RelBirthYoungPercL <- Meta5$RelBirthYoungPercC + sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$NonPeerRev <- 1 #creating a variable where non-peer reviewed (mostly raw data) is coded as 1, and peer reviewed as zero
Meta5$NonPeerRev[Meta5$PeerRev == 1] <- 0
Meta5$Concurrent <- 1 # creating a variable where concurrent PDT is 1, recollected is zero
Meta5$Concurrent[Meta5$Recollected == 1] <- 0
#measurement type X reporter interactions
Meta5$DSBMXCR <- Meta5$Variation*Meta5$ChildPDTReport
Meta5$DSBMXPR <- Meta5$Variation*Meta5$ParentPDTReport
Meta5$PBMXCR <- Meta5$Perception*Meta5$ChildPDTReport
Meta5$PBMXPR <- Meta5$Perception*Meta5$ParentPDTReport


#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#Only run this the first time, and then comment it out after the test run
#Meta5 <- Meta5[sample(nrow(Meta5), 100),]
```

```
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
predMatrix <- make.predictorMatrix(Meta5)
#changing the matrix so that only the PCAs are used as auxiliary variables for imputing
######re-do this every time to make sure column numbers match
predMatrix[,c(1:32, 42:61)] <- 0  #A value of 1 in this matrix indicates that the corresponding column variable is used to predict the corresponding row variable
#only the PCAs should have 1s, and they should go down all the rows
predMatrix #verify - always verify this, it will mess everything up if it is wrong - only PCAs should have 1s
#checking the imputation method for each variable - it only really matters for the variables with missing data
#if a nominal variable is set to pmm, change it from numeric to factor so that it will use logistic reg
impMethod <- make.method(Meta5)
impMethod


#IMPUTE THE DATA
#IMPUTE THE DATA
#IMPUTE THE DATA
imp <- mice(Meta5, print=FALSE, m=50, predictorMatrix=predMatrix, method=impMethod, seed=1234)



#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#this runs the analysis
Model1 <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE, data=Meta5)
#this summarizes the analysis
summary(Model1, digits=3, transf=transf.ztor)
#this saves the analysis
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 1.txt")
RE_model <- Model1
RE_model
sink(NULL)

#this calculates the ICCs
M1ICCs<- MetaVarCompsModel1(Model1)
#this saves the ICCs
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 1 ICCs.txt")
RE_model <- M1ICCs
RE_model
```

```
sink(NULL)



#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#no need to use the imputed data set here because there are no imputed values on the effect size
FailSafe <- fsn(Meta5$yi, Meta5$vi,type="Rosenthal", alpha=.05, digits = 4)
FailSafe
sink(file = "Findings/Imputed Data Findings/SES/SES - Fail Safe Number.txt")
RE_model <- FailSafe
RE_model
sink(NULL)

#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#Build a model without within-source variance
NoLevel2Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(0,NA, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the within-source variance
WithinSource <- anova(Model1,NoLevel2Var)
sink(file = "Findings/Imputed Data Findings/SES/SES - Within Source Variance.txt")
RE_model <- WithinSource
RE_model
sink(NULL)



#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#Build a model without between-source variance
NoLevel3Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,0, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-source variance
BetweenSource<-anova(Model1,NoLevel3Var)
sink(file = "Findings/Imputed Data Findings/SES/SES - Between Source Variance.txt")
RE_model <- BetweenSource
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#Build a model without between-sample variance
```

```
NoLevel4Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,NA, 0), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-sample variance
BetweenSample<-anova(Model1,NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/SES/SES - Between Sample Variance.txt")
RE_model <- BetweenSample
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#check frequencies of all nominal variables
sapply(Meta5[,c(5,6,8:23)], table)
table(Meta5$Variation, Meta5$ChildPDTReport)

#No Recollected
#No UsedBeta

#this runs the analysis by imputation (50 of them)
Model2 <- withold(imp, rma.mv(yi, vi,
                    mods = ~ USA+ #Level 4
                      FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                      DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
Model2Pooled <- summary(pool(Model2))
Model2Pooled[-1] <- round(Model2Pooled[-1], digits=4)
Model2Pooled$LowerBound <- Model2Pooled$estimate - (Model2Pooled$std.error*2)
Model2Pooled$UpperBound <- Model2Pooled$estimate + (Model2Pooled$std.error*2)
Model2Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 2.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)



#retrieve the variance components for Model 2
DidWork <- MetaVarCompsModel2(Model2)
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 2 - variance components.txt")
RE_model <- DidWork
```

```
RE_model
sink(NULL)

#test the significance of the variance components in Model 2 - this is done using the function below. The function takes the model for each imputation, and constrains the variance at that level to be zero and then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50 imputations
#for the MetaVarCompsModel2LevelVarianceTest function, the needed arguments are (unrestricted model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model2NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ USA+ #Level 4
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA),
tdist=TRUE))
Model2NoLevel3Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ USA+ #Level 4
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA),
tdist=TRUE))
Model2NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~ USA+ #Level 4
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0),
tdist=TRUE))
M2VarTest <- MetaVarCompsModel2LevelVarianceTest(Model2, Model2NoLevel2Var,
Model2NoLevel3Var, Model2NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 2 - variance components significance
tests.txt")
RE_model <- M2VarTest
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3
#RUNNING META REGRESSION - MODEL 3
```

```
#this runs the analysis by imputation (50 of them)
Model3 <- withold(imp, rma.mv(yi, vi,
                  mods = ~ USA+ #Level 4
                   FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                   DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT +
                   Variation:ChildPDTReport, #Interactions
                  random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
Model3Pooled <- summary(pool(Model3))
Model3Pooled[-1] <- round(Model3Pooled[-1], digits=4)
Model3Pooled$LowerBound <- Model3Pooled$estimate - (Model3Pooled$std.error*2)
Model3Pooled$UpperBound <- Model3Pooled$estimate + (Model3Pooled$std.error*2)
Model3Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 3.txt")
RE_model <- Model3Pooled
RE_model
sink(NULL)

#retrieve the variance components for Model 3
DidWork3 <- MetaVarCompsModel2(Model3)
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 3 - variance components.txt")
RE_model <- DidWork3
RE_model
sink(NULL)


#test the significance of the variance components in Model 3 - this is done using the function below. The
function takes the model for each imputation, and constrains the variance at that level to be zero and
then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50
imputations
#for the MetaVarCompsModel3LevelVarianceTest function, the needed arguments are (unrestricted
model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model3NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                             mods = ~ USA+ #Level 4
                              FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                              DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                             Variation:ChildPDTReport, #Interactions
                             random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA),
tdist=TRUE))
Model3NoLevel3Var <- withold(imp, rma.mv(yi, vi,
```

```
                        mods = ~ USA+ #Level 4
                         FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                         DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                        Variation:ChildPDTReport, #Interactions
                        random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA),
tdist=TRUE))
Model3NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                        mods = ~ USA+ #Level 4
                         FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
                         DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                        Variation:ChildPDTReport, #Interactions
                        random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0),
tdist=TRUE))
M3VarTest <- MetaVarCompsModel2LevelVarianceTest(Model3, Model3NoLevel2Var,
Model3NoLevel3Var, Model3NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/SES/SES - Model 3 - variance components significance
tests.txt")
RE_model <- M3VarTest
RE_model
sink(NULL)


#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE



#LARGE FAMILIES
#LARGE FAMILIES
#LARGE FAMILIES

#this runs the analysis by imputation (50 of them)
LFAMS <- withold(imp, rma.mv(yi, vi,
                    mods = ~ USA + #Level 4
                     FamSizeH + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                     DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))
```

```
#this summarizes and pools the analysis
LFAMS <- summary(pool(LFAMS))
LFAMS[-1] <- round(LFAMS[-1], digits=4)
LFAMS

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Simple Slopes - Large Families.txt")
RE_model <- LFAMS
RE_model
sink(NULL)



#SMALL FAMILIES
#SMALL FAMILIES
#SMALL FAMILIES

#this runs the analysis by imputation (50 of them)
SFAMS <- withold(imp, rma.mv(yi, vi,
                  mods = ~ USA + #Level 4
                    FamSizeL + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC +
AgeC + PeerRev + #Level 3
                    DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                  random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
SFAMS <- summary(pool(SFAMS))
SFAMS[-1] <- round(SFAMS[-1], digits=4)
SFAMS

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Simple Slopes - Small Families.txt")
RE_model <- SFAMS
RE_model
sink(NULL)


#HIGH PERCENT SIBY
#HIGH PERCENT SIBY
#HIGH PERCENT SIBY

#this runs the analysis by imputation (50 of them)
SIBYHIGH <- withold(imp, rma.mv(yi, vi,
                    mods = ~ USA + #Level 4
                      FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercH + GCSamePercC +
AgeDifC + AgeC + PeerRev + #Level 3
```

```
                    DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                  random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
SIBYHIGH <- summary(pool(SIBYHIGH))
SIBYHIGH[-1] <- round(SIBYHIGH[-1], digits=4)
SIBYHIGH

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Simple Slopes - High SibY Percent.txt")
RE_model <- SIBYHIGH
RE_model
sink(NULL)


#LOW PERCENT SIBY
#LOW PERCENT SIBY
#LOW PERCENT SIBY

#this runs the analysis by imputation (50 of them)
SIBYLOW <- withold(imp, rma.mv(yi, vi,
                  mods = ~ USA + #Level 4
                    FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercL + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                    DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + NegPDT +
ResourcePDT + ControlPDT, #Level 2
                  random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
SIBYLOW <- summary(pool(SIBYLOW))
SIBYLOW[-1] <- round(SIBYLOW[-1], digits=4)
SIBYLOW

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Simple Slopes - Low SibY Percent.txt")
RE_model <- SIBYLOW
RE_model
sink(NULL)



#NEG PDT
#NEG PDT
#NEG PDT

#this runs the analysis by imputation (50 of them)
NEGPDT <- withold(imp, rma.mv(yi, vi,
```

```
                    mods = ~ USA + #Level 4
                     FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC
+ AgeC + PeerRev + #Level 3
                     DadPDT + BothParsPDT + ChildPDTReport + Variation + PosPDT + OverallPDT +
ResourcePDT + ControlPDT, #Level 2
                    random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
NEGPDT <- summary(pool(NEGPDT))
NEGPDT[-1] <- round(NEGPDT[-1], digits=4)
NEGPDT

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/SES/SES - Simple Slopes - Neg PDT.txt")
RE_model <- NEGPDT
RE_model
sink(NULL)
```

```
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () ()       Four-level multilevel meta models          () () ()
# () () ()       testing links between Employment Status     () () ()
# () () ()       and absolute PDT                    () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () ()                                   () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()


#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
setwd("")

#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#for info on withold(), see https://www.metafor-
project.org/doku.php/tips:multiple_imputation_with_mice_and_metafor for more details
#I created the MetaModelsDescriptiveData(), MetaVarCompsModel1(), MetaVarCompsModel2(), and
MetaVarCompsModel2LevelVarianceTest()
#functions - more notes are available in the original file "Meta Analysis Functions.R"
library(metafor)
library(PcAux)
library(mice)
library(psych)
library(tidyverse)
library(car)

#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
WholeMeta <- read.csv(file = "Programs and Data/FinalCleanPDTMeta.csv", header = T)
```

```
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Employment Status
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Employment Status
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Employment Status
Meta1 <- subset(x = WholeMeta, subset = MagnitudeDirectional == 0 & Outcome == 13)



#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
Meta2 <- Meta1[,

c("ESNum","SampleSize","Correlation","ArticleID","PeerRev","USA","DataID","FamSize","AgeDif",

"UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","ControlPDT",

"ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","Perception",
        "Recollected","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")]

MetaModelsDescriptiveData(Meta2) # this saves the descriptive stats for the non-imputed model
#we take descriptive stats from this data frame so that it includes missing data and the non-centered
versions of continous variables



#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#we will drop the non-centered versions of continuous variables, because we want the imputation to be
based on the centered versions
Meta3 <- Meta2
attach(Meta3)
Meta3$FamSizeC <- FamSize-mean(FamSize, na.rm = T)
Meta3$AgeDifC <- AgeDif-mean(AgeDif, na.rm = T)
Meta3$AgeC <- Age-mean(Age, na.rm = T)
Meta3$WhitePercC <- WhitePerc-mean(WhitePerc, na.rm = T)
Meta3$GCSamePercC <- GCSamePerc-mean(GCSamePerc, na.rm = T)
Meta3$MalePercC <- MalePerc-mean(MalePerc, na.rm = T)
Meta3$RelBirthYoungPercC <- RelBirthYoungPerc-mean(RelBirthYoungPerc, na.rm = T)
detach(Meta3)
DropVars <- names(Meta3) %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
Meta3 <- Meta3[!DropVars]
DropVars <- Meta3 %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
```

```
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#Do this before the PCAs, it doesn't like doing it on an imputed data frame, so do it first
Meta4 <- escalc(measure = "ZCOR", ri =Correlation,
            ni = SampleSize, data = Meta3, vtype = "LS", append = TRUE)




#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#re-do this step every time

MetaNOMISS <- Meta4[, which(colMeans(is.na(Meta4)) ==.00)]#creating data frame with only variables
with no missing values

names(MetaNOMISS)#I run this to get the names and then sort those into the type of variable below

## Prepare the data by first sorting out what types of variables I have - continuous variables are left off
the code but listed below
myNoms   <-
c("PeerRev","USA","UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","
ControlPDT","ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","P
erception","Recollected")
myIds    <- c("ESNum","ArticleID","DataID")
#myOrds   <- c() #none this time
#Continuous - "SampleSize"     "Correlation"  "FamSizeC"       "AgeC"          "WhitePercC"
"MalePercC"       "yi"  "vi"



#Create data frame for creating PCAs - prep data step
ReadyForPCA <- prepData(rawData = MetaNOMISS,
                  nomVars = myNoms,
                  ordVars = NULL, # make sure to update this if there are ordinal variables
                  idVars = myIds,
                  moderators = NULL,
                  verbose = 2L,
                  nProcess = 4,
                  simMode = F)
y #the prepData function often needs a yes or no to continue, this tells it to keep going

## Create principal component auxiliary variables:
PCAOut <- createPcAux(pcAuxData = ReadyForPCA,
                nComps = c(0.5,0.5),
```

```
            maxPolyPow = 3L,
            doImputation = FALSE,
            interactType = 1)
y #the createPcAux function often needs a yes or no to continue, this tells it to keep going


## Merge the PC auxiliaries with the original data:
Meta5 <- mergePcAux(pcAuxData = PCAOut, rawData = Meta4) #use this data for imputation


#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
Meta5$FamSizeH <- Meta5$FamSizeC - sd(Meta5$FamSizeC, na.rm = T)
Meta5$FamSizeL <- Meta5$FamSizeC + sd(Meta5$FamSizeC, na.rm = T)
Meta5$AgeDifH <- Meta5$AgeDifC - sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeDifL <- Meta5$AgeDifC + sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeH <- Meta5$AgeC - sd(Meta5$AgeC, na.rm = T)
Meta5$AgeL <- Meta5$AgeC + sd(Meta5$AgeC, na.rm = T)
Meta5$WhitePercH <- Meta5$WhitePercC - sd(Meta5$WhitePercC, na.rm = T)
Meta5$WhitePercL <- Meta5$WhitePercC + sd(Meta5$WhitePercC, na.rm = T)
Meta5$GCSamePercH <- Meta5$GCSamePercC - sd(Meta5$GCSamePercC, na.rm = T)
Meta5$GCSamePercL <- Meta5$GCSamePercC + sd(Meta5$GCSamePercC, na.rm = T)
Meta5$MalePercH <- Meta5$MalePercC - sd(Meta5$MalePercC, na.rm = T)
Meta5$MalePercL <- Meta5$MalePercC + sd(Meta5$MalePercC, na.rm = T)
Meta5$RelBirthYoungPercH <- Meta5$RelBirthYoungPercC - sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$RelBirthYoungPercL <- Meta5$RelBirthYoungPercC + sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$NonPeerRev <- 1 #creating a variable where non-peer reviewed (mostly raw data) is coded as 1, and peer reviewed as zero
Meta5$NonPeerRev[Meta5$PeerRev == 1] <- 0
Meta5$Concurrent <- 1 # creating a variable where concurrent PDT is 1, recollected is zero
Meta5$Concurrent[Meta5$Recollected == 1] <- 0
#measurement type X reporter interactions
Meta5$DSBMXCR <- Meta5$Variation*Meta5$ChildPDTReport
Meta5$DSBMXPR <- Meta5$Variation*Meta5$ParentPDTReport
Meta5$PBMXCR <- Meta5$Perception*Meta5$ChildPDTReport
Meta5$PBMXPR <- Meta5$Perception*Meta5$ParentPDTReport



#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#Only run this the first time, and then comment it out after the test run
#Meta5 <- Meta5[sample(nrow(Meta5), 100),]
```

```
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
predMatrix <- make.predictorMatrix(Meta5)
#changing the matrix so that only the PCAs are used as auxiliary variables for imputing
######re-do this every time to make sure column numbers match
predMatrix[,c(1:32, 36:55)] <- 0  #A value of 1 in this matrix indicates that the corresponding column variable is used to predict the corresponding row variable
#only the PCAs should have 1s, and they should go down all the rows
predMatrix #verify - always verify this, it will mess everything up if it is wrong - only PCAs should have 1s
#checking the imputation method for each variable - it only really matters for the variables with missing data
#if a nominal variable is set to pmm, change it from numeric to factor so that it will use logistic reg
impMethod <- make.method(Meta5)
impMethod


#IMPUTE THE DATA
#IMPUTE THE DATA
#IMPUTE THE DATA
imp <- mice(Meta5, print=FALSE, m=50, predictorMatrix=predMatrix, method=impMethod, seed=1234)



#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#this runs the analysis
Model1 <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE, data=Meta5)
#this summarizes the analysis
summary(Model1, digits=3, transf=transf.ztor)
#this saves the analysis
sink(file = "Findings/Imputed Data Findings/Employment Status/Employment Status - Model 1.txt")
RE_model <- Model1
RE_model
sink(NULL)

#this calculates the ICCs
M1ICCs<- MetaVarCompsModel1(Model1)
#this saves the ICCs
sink(file = "Findings/Imputed Data Findings/Employment Status/Employment Status - Model 1 ICCs.txt")
RE_model <- M1ICCs
```

```
RE_model
sink(NULL)


#send text message
Sys.setenv(TWILIO_SID = "AC7ef37e5096b51b23bb6e95c9d469b0e9")
Sys.setenv(TWILIO_TOKEN = "0b5fa8428303e87734b385f79c5d0a31")
my_message <- tw_send_message(
  to = +18148765027,
  from = +14322781092,
  body = paste("Employment Status Model 1 finished"))



#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#no need to use the imputed data set here because there are no imputed values on the effect size
FailSafe <- fsn(Meta5$yi, Meta5$vi,type="Rosenthal", alpha=.05, digits = 4)
FailSafe
sink(file = "Findings/Imputed Data Findings/Employment Status/Employment Status - Fail Safe
Number.txt")
RE_model <- FailSafe
RE_model
sink(NULL)

#send text message
Sys.setenv(TWILIO_SID = "AC7ef37e5096b51b23bb6e95c9d469b0e9")
Sys.setenv(TWILIO_TOKEN = "0b5fa8428303e87734b385f79c5d0a31")
my_message <- tw_send_message(
  to = +18148765027,
  from = +14322781092,
  body = paste("Employment Status Fail Safe Number finished"))

#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#Build a model without within-source variance
NoLevel2Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(0,NA, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the within-source variance
WithinSource <- anova(Model1,NoLevel2Var)
sink(file = "Findings/Imputed Data Findings/Employment Status/Employment Status - Within Source
Variance.txt")
RE_model <- WithinSource
RE_model
sink(NULL)
```

```
#send text message
Sys.setenv(TWILIO_SID = "AC7ef37e5096b51b23bb6e95c9d469b0e9")
Sys.setenv(TWILIO_TOKEN = "0b5fa8428303e87734b385f79c5d0a31")
my_message <- tw_send_message(
  to = +18148765027,
  from = +14322781092,
  body = paste("Employment Status Level 2 Variation finished"))



#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#Build a model without between-source variance
NoLevel3Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,0, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-source variance
BetweenSource<-anova(Model1,NoLevel3Var)
sink(file = "Findings/Imputed Data Findings/Employment Status/Employment Status - Between Source
Variance.txt")
RE_model <- BetweenSource
RE_model
sink(NULL)

#send text message
Sys.setenv(TWILIO_SID = "AC7ef37e5096b51b23bb6e95c9d469b0e9")
Sys.setenv(TWILIO_TOKEN = "0b5fa8428303e87734b385f79c5d0a31")
my_message <- tw_send_message(
  to = +18148765027,
  from = +14322781092,
  body = paste("Employment Status Level 3 Variation finished"))


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#Build a model without between-sample variance
NoLevel4Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,NA, 0), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-sample variance
BetweenSample<-anova(Model1,NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Employment Status/Employment Status - Between Sample
Variance.txt")
RE_model <- BetweenSample
RE_model
sink(NULL)
```

```
#send text message
Sys.setenv(TWILIO_SID = "AC7ef37e5096b51b23bb6e95c9d469b0e9")
Sys.setenv(TWILIO_TOKEN = "0b5fa8428303e87734b385f79c5d0a31")
my_message <- tw_send_message(
  to = +18148765027,
  from = +14322781092,
  body = paste("Employment Status Level 4 Variation finished"))



#NO OTHER MODELS - THERE WAS NOT SIGNIFICANT VARIATION AROUND THE INTERCEPT
#NO OTHER MODELS - THERE WAS NOT SIGNIFICANT VARIATION AROUND THE INTERCEPT
#NO OTHER MODELS - THERE WAS NOT SIGNIFICANT VARIATION AROUND THE INTERCEPT
```

```
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()       Four-level multilevel meta models          () () ()
# () () ()       testing links between Marital Status       () () ()
# () () ()       and absolute PDT                    () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () ()                                  () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()
# () () () () () () () () () () () () () () () () () () ()


#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
#SET THE WORKING DIRCTORY
setwd("C:/Users/18148/Box/alexjensen/Research/Working Papers/META - PDT as outcome")

#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#ACTIAVTE ALL NEEDED PACKAGES AND CREATE THE withold() FUNCTION AND THE
MetaModelsDescriptiveData() FUNCTION
#for info on withold(), see https://www.metafor-
project.org/doku.php/tips:multiple_imputation_with_mice_and_metafor for more details
#I created the MetaModelsDescriptiveData(), MetaVarCompsModel1(), MetaVarCompsModel2(), and
MetaVarCompsModel2LevelVarianceTest()
#functions - more notes are available in the original file "Meta Analysis Functions.R"
library(metafor)
library(PcAux)
library(mice)
library(psych)
library(tidyverse)
library(car)

#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
#IMPORT THE ENTIRE META DATA FRAME
WholeMeta <- read.csv(file = "Programs and Data/FinalCleanPDTMeta.csv", header = T)
```

```
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Marital Status
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Marital Status
#SUBSET A NEW DATA FRAME THAT HAS ONLY CASES WITH ABSOLUTE PDT AND Marital Status
Meta1 <- subset(x = WholeMeta, subset = MagnitudeDirectional == 0 & Outcome == 14)



#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
#CREATE A SUBSET WITH ONLY THE NEEDED VARIABLES AND EXPORT THE DESCRIPTIVE STATS
Meta2 <- Meta1[,
c("ESNum","SampleSize","Correlation","ArticleID","PeerRev","USA","DataID","FamSize","AgeDif",
"UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","ControlPDT",
"ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","Perception",
"Recollected","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")]

MetaModelsDescriptiveData(Meta2) # this saves the descriptive stats for the non-imputed model
#we take descriptive stats from this data frame so that it includes missing data and the non-centered
versions of continous variables



#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#CREATE A NEW DATA FRAME WITH CENTERED VERSIONS OF ALL CONTINOUS VARIABLES
#we will drop the non-centered versions of continuous variables, because we want the imputation to be
based on the centered versions
Meta3 <- Meta2
attach(Meta3)
Meta3$FamSizeC <- FamSize-mean(FamSize, na.rm = T)
Meta3$AgeDifC <- AgeDif-mean(AgeDif, na.rm = T)
Meta3$AgeC <- Age-mean(Age, na.rm = T)
Meta3$WhitePercC <- WhitePerc-mean(WhitePerc, na.rm = T)
Meta3$GCSamePercC <- GCSamePerc-mean(GCSamePerc, na.rm = T)
Meta3$MalePercC <- MalePerc-mean(MalePerc, na.rm = T)
Meta3$RelBirthYoungPercC <- RelBirthYoungPerc-mean(RelBirthYoungPerc, na.rm = T)
detach(Meta3)
DropVars <- names(Meta3) %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")
Meta3 <- Meta3[!DropVars]
DropVars <- Meta3 %in%
c("AgeDif","FamSize","Age","WhitePerc","GCSamePerc","MalePerc","RelBirthYoungPerc")



#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
#SET THE EFFECT SIZES
```

```
#Do this before the PCAs, it doesn't like doing it on an imputed data frame, so do it first
Meta4 <- escalc(measure = "ZCOR", ri =Correlation,
                ni = SampleSize, data = Meta3, vtype = "LS", append = TRUE)




#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#CREATE THE PCAS BASED ON VARIABLES IN Meta3 THAT HAVE NO MISSING VALUES
#re-do this step every time

MetaNOMISS <- Meta4[, which(colMeans(is.na(Meta4)) ==.00)]#creating data frame with only variables
with no missing values

names(MetaNOMISS)#I run this to get the names and then sort those into the type of variable below

## Prepare the data by first sorting out what types of variables I have - continuous variables are left off
the code but listed below
myNoms   <-
c("PeerRev","USA","UsedBeta","OverallPDT","PosPDT","NegPDT","EvalPDT","ResourcePDT","FairPDT","
ControlPDT","ChildPDTReport","ParentPDTReport","MomPDT","DadPDT","BothParsPDT","Variation","P
erception","Recollected")
myIds    <- c("ESNum", "ArticleID", "DataID")
#myOrds   <- c() #none this time
#Continuous - "SampleSize"      "Correlation"    "FamSizeC","AgeC","WhitePercC","MalePercC","yi","vi"




#Create data frame for creating PCAs - prep data step
ReadyForPCA <- prepData(rawData = MetaNOMISS,
                        nomVars = myNoms,
                        ordVars = NULL, # make sure to update this if there are ordinal variables
                        idVars = myIds,
                        moderators = NULL,
                        verbose = 2L,
                        nProcess = 4,
                        simMode = F)

## Create principal component auxiliary variables:
PCAOut <- createPcAux(pcAuxData = ReadyForPCA,
                        nComps = c(0.5,0.5),
                        maxPolyPow = 3L,
                        doImputation = FALSE,
                        interactType = 1)
```

```
## Merge the PC auxiliaries with the original data:
Meta5 <- mergePcAux(pcAuxData = PCAOut, rawData = Meta4) #use this data for imputation


#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
#CREATING INTERACTION TERMS, HIGH AND LOW OF MODERATORS AND REVERSING DUMMY CODES AS NEEDED - ALL FOR INTERACTIONS AND SIMPLE SLOPES
Meta5$FamSizeH <- Meta5$FamSizeC - sd(Meta5$FamSizeC, na.rm = T)
Meta5$FamSizeL <- Meta5$FamSizeC + sd(Meta5$FamSizeC, na.rm = T)
Meta5$AgeDifH <- Meta5$AgeDifC - sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeDifL <- Meta5$AgeDifC + sd(Meta5$AgeDifC, na.rm = T)
Meta5$AgeH <- Meta5$AgeC - sd(Meta5$AgeC, na.rm = T)
Meta5$AgeL <- Meta5$AgeC + sd(Meta5$AgeC, na.rm = T)
Meta5$WhitePercH <- Meta5$WhitePercC - sd(Meta5$WhitePercC, na.rm = T)
Meta5$WhitePercL <- Meta5$WhitePercC + sd(Meta5$WhitePercC, na.rm = T)
Meta5$GCSamePercH <- Meta5$GCSamePercC - sd(Meta5$GCSamePercC, na.rm = T)
Meta5$GCSamePercL <- Meta5$GCSamePercC + sd(Meta5$GCSamePercC, na.rm = T)
Meta5$MalePercH <- Meta5$MalePercC - sd(Meta5$MalePercC, na.rm = T)
Meta5$MalePercL <- Meta5$MalePercC + sd(Meta5$MalePercC, na.rm = T)
Meta5$RelBirthYoungPercH <- Meta5$RelBirthYoungPercC - sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$RelBirthYoungPercL <- Meta5$RelBirthYoungPercC + sd(Meta5$RelBirthYoungPercC, na.rm = T)
Meta5$NonPeerRev <- 1 #creating a variable where non-peer reviewed (mostly raw data) is coded as 1, and peer reviewed as zero
Meta5$NonPeerRev[Meta5$PeerRev == 1] <- 0
Meta5$Concurrent <- 1 # creating a variable where concurrent PDT is 1, recollected is zero
Meta5$Concurrent[Meta5$Recollected == 1] <- 0
#measurement type X reporter interactions
Meta5$DSBMXCR <- Meta5$Variation*Meta5$ChildPDTReport
Meta5$DSBMXPR <- Meta5$Variation*Meta5$ParentPDTReport
Meta5$PBMXCR <- Meta5$Perception*Meta5$ChildPDTReport
Meta5$PBMXPR <- Meta5$Perception*Meta5$ParentPDTReport


#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#DRAW SMALL RANDOM SAMPLE OF THE DATA TO TEST RUN THE ANALYSIS AND MAKE SURE IT WORKS
#Only run this the first time, and then comment it out after the test run
#Meta5 <- Meta5[sample(nrow(Meta5), 100),]


#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
```

```
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
#PREPPING THE PREDICTION MATRIX FOR THE MICE MODEL - THIS TELLS IT WHICH VARIABLES TO USE AS PREDICTORS AND WHICH TO IMPUTE
predMatrix <- make.predictorMatrix(Meta5)
#changing the matrix so that only the PCAs are used as auxiliary variables for imputing
######re-do this every time to make sure column numbers match
predMatrix[,c(1:32, 37:56)] <- 0 #A value of 1 in this matrix indicates that the corresponding column variable is used to predict the corresponding row variable
#only the PCAs should have 1s, and they should go down all the rows
predMatrix #verify - always verify this, it will mess everything up if it is wrong - only PCAs should have 1s
#checking the imputation method for each variable - it only really matters for the variables with missing data
#if a nominal variable is set to pmm, change it from numeric to factor so that it will use logistic reg
impMethod <- make.method(Meta5)
impMethod


#IMPUTE THE DATA
#IMPUTE THE DATA
#IMPUTE THE DATA
imp <- mice(Meta5, print=FALSE, m=50, predictorMatrix=predMatrix, method=impMethod, seed=1234)



#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#RUN AND EXPORT THE ANALYSIS - MODEL 1
#this runs the analysis
Model1 <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE, data=Meta5)
#this summarizes the analysis
summary(Model1, digits=3, transf=transf.ztor)
#this saves the analysis
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Model 1.txt")
RE_model <- Model1
RE_model
sink(NULL)

#this calculates the ICCs
M1ICCs<- MetaVarCompsModel1(Model1)
#this saves the ICCs
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Model 1 ICCs.txt")
RE_model <- M1ICCs
RE_model
sink(NULL)
```

```
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#CHECK FAIL SAFE NUMBER FOR MODEL 1
#no need to use the imputed data set here because there are no imputed values on the effect size
FailSafe <- fsn(Meta5$yi, Meta5$vi,type="Rosenthal", alpha=.05, digits = 4)
FailSafe
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Fail Safe Number.txt")
RE_model <- FailSafe
RE_model
sink(NULL)

#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#CHECK FOR SIGNIFICANT WITHIN SOURCE VARIATION - LEVEL 2
#Build a model without within-source variance
NoLevel2Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(0,NA, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the within-source variance
WithinSource <- anova(Model1,NoLevel2Var)
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Within Source Variance.txt")
RE_model <- WithinSource
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 3
#Build a model without between-source variance
NoLevel3Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,0, NA), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-source variance
BetweenSource<-anova(Model1,NoLevel3Var)
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Between Source
Variance.txt")
RE_model <- BetweenSource
RE_model
sink(NULL)


#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#CHECK FOR SIGNIFICANT BETWEEN SOURCE VARIATION - LEVEL 4
#Build a model without between-sample variance
NoLevel4Var <- rma.mv(yi, vi, random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),
                      sigma2=c(NA,NA, 0), tdist=TRUE, data=Meta5)
# Perform a likelihood-ratio-test to determine the significance of the between-sample variance
```

```
BetweenSample<-anova(Model1,NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Between Sample
Variance.txt")
RE_model <- BetweenSample
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#RUNNING META REGRESSION - MODEL 2
#check frequencies of all nominal variables
sapply(Meta5[,c(5,6,8:23)], table)

#There was not enough between sample variation  - no need to test USA
#Not enough on UsedBeta 134 and 3
#Not enough on ResourcePDT 114 and 23
#Not enough on ControlPDT 133 and 4
#Not enough on ChilPDTReport 124 and 13
#Not enough on BothParsPDT 135 and 2
#Not enough on Variation 127 and 10
#Not enough on Recollected 121 and 16

#OverallPDT is the reference group for domain

#this runs the analysis by imputation (50 of them)
Model2 <- withold(imp, rma.mv(yi, vi,
mods = ~
FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC + AgeC + PeerRev
+ #Level 3
DadPDT + PosPDT + NegPDT, #Level 2
random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))


#this summarizes and pools the analysis
Model2Pooled <- summary(pool(Model2))
Model2Pooled[-1] <- round(Model2Pooled[-1], digits=4)
Model2Pooled$LowerBound <- Model2Pooled$estimate - (Model2Pooled$std.error*2)
Model2Pooled$UpperBound <- Model2Pooled$estimate + (Model2Pooled$std.error*2)
Model2Pooled

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Model 2.txt")
RE_model <- Model2Pooled
RE_model
sink(NULL)
```

```
#retrieve the variance components for Model 2
DidWork <- MetaVarCompsModel2(Model2)
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Model 2 - variance components.txt")
RE_model <- DidWork
RE_model
sink(NULL)

#test the significance of the variance components in Model 2 - this is done using the function below. The function takes the model for each imputation, and constrains the variance at that level to be zero and then tests model fit against the unrestricted(full) model, it then pools the needed stats across all 50 imputations
#for the MetaVarCompsModel2LevelVarianceTest function, the needed arguments are (unrestricted model, model with no level 2 variance, model with no level 3 variance, model with no level 4 variance)
Model2NoLevel2Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + PosPDT + NegPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(0,NA, NA), tdist=TRUE))
Model2NoLevel3Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + PosPDT + NegPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,0, NA), tdist=TRUE))
Model2NoLevel4Var <- withold(imp, rma.mv(yi, vi,
                          mods = ~
                            FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC + AgeC + PeerRev + #Level 3
                            DadPDT + PosPDT + NegPDT, #Level 2
                          random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID),sigma2=c(NA,NA, 0), tdist=TRUE))
M2VarTest <- MetaVarCompsModel2LevelVarianceTest(Model2, Model2NoLevel2Var, Model2NoLevel3Var, Model2NoLevel4Var)
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Model 2 - variance components significance tests.txt")
RE_model <- M2VarTest
RE_model
sink(NULL)



#RUNNING META REGRESSION - MODEL 3
```
#RUNNING META REGRESSION - MODEL 3

```
#RUNNING META REGRESSION - MODEL 3

#No need to test model 3, there were not enough cases on ChildPDTReport and Variation



#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE
#TESTING SIMPLE SLOPES - DO NOT RE-RUN MODELS THAT THE INFO IS IN MODEL 2 - ONLY THE
INVERSE OF EACH VARIABLE

#NegPDT
#NegPDT
#NegPDT

NegPDT <- withold(imp, rma.mv(yi, vi,
mods = ~
FamSizeC + WhitePercC + MalePercC + RelBirthYoungPercC + GCSamePercC + AgeDifC + AgeC + PeerRev
+ #Level 3
DadPDT + PosPDT + OverallPDT, #Level 2
random = list(~ 1 | ESNum, ~ 1 | ArticleID, ~ 1 | DataID), tdist=TRUE))

#this summarizes and pools the analysis
NegPDT <- summary(pool(NegPDT))
NegPDT[-1] <- round(NegPDT[-1], digits=4)
NegPDT

#this exports the findings to a text file
sink(file = "Findings/Imputed Data Findings/Marital Status/Marital Status - Simple Slopes - Neg PDT.txt")
RE_model <- NegPDT
RE_model
sink(NULL)
```